

논문 2008-45CI-1-2

영역기반 이미지 검색을 위한 칼라 이미지 세그멘테이션

(Color Image Segmentation for Region-Based Image Retrieval)

황 환 규*

(Whan-Kyu Whang)

요 약

효율적인 저차원의 인덱싱을 제공하기 위해 이미지를 유사한 성질을 갖는 영역으로 나누고, 나누어진 영역에 대해 유사성을 비교하는 영역 기반 이미지 검색이 제안되었다. 그러나 영역 기반 이미지 검색은 이미지를 유사한 영역으로 나누기 위한 이미지 세그멘테이션 기술이 추가적으로 필요하다. 일반적인 칼라 자연 이미지의 경우 다양한 칼라와 질감 성분을 갖는 영역으로 나누는 것은 많은 어려움이 있다. 본 논문에서는 자동적인 칼라 이미지 세그멘테이션 알고리즘을 제안한다. 제안하는 세그멘테이션 방법은 양자화를 통해 칼라수를 줄이고 양자화 된 이미지를 Fisher의 클래스 선형 판별식을 이용하여 이미지의 전체적인 에지를 보여주는 그레이 레벨 이미지를 생성한다. 이렇게 얻은 그레이 레벨 에지 이미지를 지역적 임계치 비교를 통해 이진 에지 이미지로 변환하고 이진 에지의 끊어진 부분을 찾아내어 인접 에지에 연결하여 영역을 생성한다. 마지막으로 나누어진 영역간의 유사성을 비교하고 유사한 영역을 병합하여 최종 세그멘테이션 결과 이미지를 생성한다. 본 논문에서는 세그멘테이션 알고리즘을 이용한 영역 기반 이미지 검색 시스템을 구현하였으며, 다양한 실험에 의하면 제안한 세그멘테이션 방법이 다양한 이미지에 대하여 양질의 세그멘테이션 결과를 보이는 것으로 나타났다.

Abstract

Region-based image retrieval techniques, which divide image into similar regions having similar characteristics and examine similarities among divided regions, were proposed to support an efficient low-dimensional color indexing scheme. However, color image segmentation techniques are required additionally. The problem of segmentation is difficult because of a large variety of color and texture. It is known to be difficult to identify image regions containing the same color-texture pattern in natural scenes. In this paper we propose an automatic color image segmentation algorithm. The colors in each image are first quantized to reduce the number of colors. The gray level of image representing the outline edge of image is constructed in terms of Fisher's multi-class linear discriminant on quantized images. The gray level of image is transformed into a binary edge image. The edge showing the outline of the binary edge image links to the nearest edge if disconnected. Finally, the final segmentation image is obtained by merging similar regions. In this paper we design and implement a region-based image retrieval system using the proposed segmentation. A variety of experiments show that the proposed segmentation scheme provides good segmentation results on a variety of images.

Keywords: 내용 기반 이미지 검색, 칼라 이미지 세그멘테이션, 영역 기반 이미지 검색

I. 서 론

인터넷과 컴퓨터 성능의 급속한 발달로 텍스트 정보와 함께 이미지 정보에 대한 효과적인 저장, 검색방법이 요구되었다. 전통적인 이미지 검색 방법으로 이미지

에 대한 정보를 사람이 텍스트화 하고 이를 기존의 DBMS에 저장하는 텍스트 기반 이미지 검색 방법을 사용하였다^[1~3]. 이 같은 텍스트 기반 이미지 검색방법은 이미지에 대한 주석을 생성하는 과정에서 개인적인 주관차이로 인한 오차와 방대한 이미지 데이터에 사람이 직접 주석을 작성해야하는 별도의 노력이 필요하다는 문제점이 있다. 이와 같은 문제를 해결하기 위해 이미지 자체의 정보를 이용해 유사성을 비교하는 내용기반 이미지 검색 방법(CBIR : Content Based Image

* 정희원, 강원대학교 IT대학 컴퓨터학부 컴퓨터정보통신공학전공

(Department of Computer and Communication Eng., Kangwon National University)

접수일자: 2007년10월12일, 수정완료일: 2008년1월11일

Retrieval)이 연구되었다^[4~9].

내용기반 이미지 검색에서 이미지 정보는 다차원상의 점으로 표현되어 다차원 인덱스에 저장된다. 그러나 다차원 인덱스의 경우 20차원 이상이 되면 급속하게 성능이 저하되어 순차 검색보다 느린 검색결과를 보여준다^[10]. 따라서 효과적인 검색을 위해서 인덱스를 사용할 경우 이미지 정보의 수가 제한되므로 작은 이미지에서는 효과적이거나 이미지의 크기가 커지면 검색 능력이 떨어지는 문제점을 갖는다. 이와 같은 문제점을 해결하기 위해 이미지를 유사한 성질(칼라-질감)을 갖는 영역으로 나누어 이미지 정보의 수를 줄인 후, 나누어진 영역에 대해 유사성을 비교하는 영역기반 이미지 검색 (Region Based Image Retrieval)이 제안되었다^[8~9]. 영역기반 이미지 검색은 정보의 수를 줄일 수 있어 저차원의 인덱스를 사용하여도 효과적인 성능을 보여준다. 그러나 영역기반 이미지 검색은 이미지를 유사한 영역으로 나누기 위한 이미지 세그멘테이션 기술이 추가적으로 필요하다.

일반적으로 이미지 세그멘테이션은 특정 응용분야에 사용되는 단순한 이미지에 대해 연구가 진행되었다^[11~12]. 그러나 범용적인 일반 칼라 이미지의 경우, 이미지에 높은 수(2563)의 칼라 정보와 질감정보가 포함되어 세그멘테이션을 어렵게 하는 문제점이 있다.

본 논문에서는 범용적인 칼라 자연 이미지의 세그멘테이션 방법을 제안하고, 세그멘테이션된 영역을 이용한 영역기반 이미지 검색시스템을 구현하였다. 제안한 세그멘테이션 방법은 처음 양자화를 통해 칼라 수를 줄이고 양자화 된 이미지를 Fisher의 클래스 선형 판별식^[11]을 이용하여 이미지의 전체적인 에지를 보여주는 그레이 레벨의 이미지(gray level edge image)를 생성한다. 이렇게 얻은 그레이 레벨 에지 이미지를 윈도우 단위의 임계치 비교를 통해 이진(binary) 에지 이미지로 변환한다. 이진 에지 이미지는 잡음제거와 함께 끊어진 에지 정보를 끊어진 방향에서 가장 가까운 에지를 연결하여 끊어지지 않은 에지 이미지를 생성하고 유사한 인접영역을 합병하는 작업을 통해 최종적인 세그멘테이션 결과를 얻게 된다.

논문의 진행은 II장에서 배경 연구를 설명하고, III장에서 제안한 칼라 이미지 세그멘테이션 방법에 대해 설명한다. IV장에서는 제안한 세그멘테이션 방법을 이용한 영역기반 이미지 검색 시스템의 구현내용과 결과를 보여주고, 마지막 V장에서 결론을 맺는다.

II. 배경 연구

본 장에서는 내용기반 이미지 검색의 문제점을 설명하고 이를 해결하기 위한 방법으로 영역기반 이미지 검색방법에 대해 설명한다. 마지막으로 영역기반 이미지 검색방법의 전제가 되는 이미지 세그멘테이션 기술에 대해 설명한다.

1. 내용기반 이미지 검색

내용기반 이미지 검색이란 이미지 자체의 정보를 이용하여 이미지간의 유사성을 비교하는 방법이다. 내용기반 이미지 검색은 컴퓨터가 이미지를 객체단위로 인식하는 것이 아니라 픽셀(pixel)단위로 인식하므로 이미지의 정확한 정보를 추출하여 유사성을 비교하는 것이 어렵다.

실용적인 내용기반 이미지 검색시스템을 구현하기 위해서는 다음과 같은 두 가지 문제를 효과적으로 해결할 수 있어야 한다.

1. 이미지 정보를 어떻게 추출할 것인가?
2. 이미지 정보를 어떻게 저장, 검색할 것인가?

가장 보편적인 이미지 정보추출 방법으로 칼라 히스토그램(Color Histogram) 방법이 있다^[13]. 이 방법은 각 칼라에 대해 이미지 픽셀 빈도를 나타내는 방법이다. 칼라 히스토그램은 칼라에 대한 빈도수만으로 표현하기 때문에 이미지의 회전이나 크기 및 위치변화, 작은 시야변경에 대해 유사성을 잃지 않는 장점을 가진다. 그러나 칼라 히스토그램방식은 모든 칼라에 대한 빈도수를 가지므로 정보의 차원이 높아지는 문제점(RGB공간에서 2563개의 정보 차원 필요)이 있다. 이는 결국 순차 검색을 요구하게 되므로 방대한 양의 이미지에 대해 빠른 검색을 제공할 수가 없다.

반면 차원의 수를 줄여 인덱스를 사용할 경우 정확한 이미지의 유사성을 비교할 수 없는 문제점을 가진다. IBM의 최초 상용 이미지 검색 시스템인 QBIC의 경우, 이미지의 평균 RGB 칼라를 사용하여 이미지의 칼라 유사성을 비교하였다^[4,7]. 이 방법은 이미지를 단 3개 차원(R, G, B)의 점으로 표현할 수 있으므로 효과적인 인덱스 저장이 가능하며, 빠른 검색능력을 제공한다. 그러나 단 3개의 정보로 전체 이미지를 표현하는 것에는 많은 한계가 있다. 작은 크기의 이미지에서 평균 RGB의 사용은 어느 정도 이미지간의 유사성을 느낄 수 있으

나, 이미지 크기가 커지면 사람이 느끼는 유사성을 유지하기 어렵게 된다. 이는 이미지 크기가 커지면 이미지 내에 다양한 칼라와 오브젝트가 존재하기 때문에 이를 평균값으로 사용할 경우 많은 오차를 보일 수 있기 때문이다.

이와 같이 이미지의 크기가 커지면 표현해야 할 정보의 양이 많아지기 때문에 정확한 이미지의 표현이 어려울 뿐만 아니라 인덱싱의 어려운 문제가 발생하게 되고, 차원의 수를 줄여 인덱스를 사용할 경우 정보 요약 과정에서 잃어버리는 정보의 양이 많아져 정확한 유사성 비교능력을 보이지 못하는 상반되는 문제점을 갖게 된다.

2. 영역기반 이미지 검색

2.1절에서 설명한 것과 같이 큰 이미지에 대한 이미지 유사성 비교는 많은 어려움이 있으며 이는 내용기반 이미지 검색에서 또 다른 연구로 다루어진다^[6, 8, 13-15]. 앞에서 설명한 이미지 크기에 대한 문제와 정보의 차원 문제를 해결하기 위한 방법으로 영역기반 이미지 검색 방법이 제안되었다. 영역기반 이미지 검색은 먼저 이미지를 유사한 성질(칼라-질감)을 가진 영역으로 나눈 후, 나누어진 각 영역에 대해서 정보를 추출하여 저장하는 방법을 사용한다. 나누어진 영역은 전체 이미지와는 다르게 유사한 성질을 가지고 있으므로 저차원의 정보로도 정확하게 표현할 수 있다. 그림 1은 3개의 영역으로 세그멘테이션한 결과를 보여준다.

나누어진 영역은 영역 정보와 함께 해당 영역을 포함하고 있는 이미지 이름을 포함하여 하나의 엔트리(entry)로 인덱스에 저장된다. 사용자가 질의 이미지에 대해 검색하고자하는 영역을 선택하여 질의를 하면, 이미지 데이터베이스는 유사한 영역을 찾은 후 해당 영역을 포함하는 이미지를 사용자에게 출력하여 보여주게 된다.

이처럼 영역기반 이미지 검색은 이미지 크기에 대한 유사성 문제와 인덱스 문제를 동시에 해결해 줄 수 있

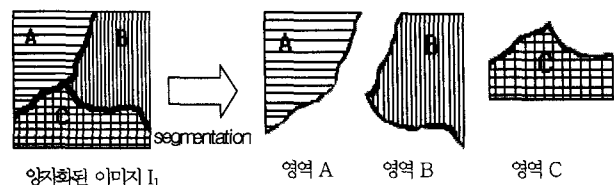


그림 1. 이미지 세그멘테이션
Fig. 1. Image Segmentation.

다. 그러나 영역기반 이미지 검색방법은 유사영역으로 나누는 추가적인 작업을 필요하게 된다. 이처럼 전체 이미지를 유사한 성질을 갖는 영역으로 나누는 작업은 사람이 직접 처리할 경우 정확한 결과의 분할이 가능하나, 방대한 이미지 처리를 위해서는 많은 시간이 소요되므로 이 작업 역시 자동적으로 이루어져야 한다. 이처럼 이미지 특성에 따라 이미지를 나누는 방법을 이미지 세그멘테이션이라 하며 효과적인 성능의 세그멘테이션은 영역기반 이미지 검색의 성능을 좌우하는 중요한 역할을 한다.

3. 이미지 세그멘테이션

영역기반 이미지 검색을 위한 영역은 동일한 질감-칼라 성분을 유지하여야 하며 모든 픽셀은 어느 한 영역에 속해야 한다. 또한 영역간은 중첩되지 않으므로 영역간의 교집합은 공집합이 된다. 그림 2는 세그멘테이션 과정을 통해 생성되는 영역의 정의를 보여준다. 이미지 세그멘테이션은 영역 생성방법에 따라 픽셀 기반, 영역기반, 에지 기반 이미지 세그멘테이션 방법이 있다.

픽셀 기반 세그멘테이션은 픽셀 단위로 영역을 구분하는 방법으로, 일반적으로 칼라 히스토그램의 1개 이상의 임계치를 통해 각각의 픽셀들이 어느 영역의 원소인지를 구분한다^[12-13]. 히스토그램을 이용한 칼라 세그멘테이션의 경우 임계치를 선택하는 방법이 성능의 주요 핵심이 된다. 예를 들어 칼라 이미지의 밝기 히스토그램의 최대치를 임계치로 잡을 경우 그림 3과 같은 이미지 세그멘테이션 결과가 된다. 임계치 1개를 사용하였을 경우 영역은 2개(그림 3의 검은 부분과 흰 부분)로 나누어지게 된다. 히스토그램 방법은 좀 더 세분화된 영역으로 나누기 위해 여러 개의 임계치를 두거나, 여러 칼라 공간(RGB, HSI, YIQ)의 각 채널의 히스토그램의 임계치를 이용한다^[13]. 그러나 일반적으로 픽셀 기반 이미지 검색은 인접 픽셀과의 관계를 고려하지 않기 때문에 서로 떨어져 있는 픽셀이 동일한 영역이 될 수 있다. 그러므로 픽셀 기반 이미지 세그멘테이션은 내용기반 이미지 검색에는 효과적이지 못하다.

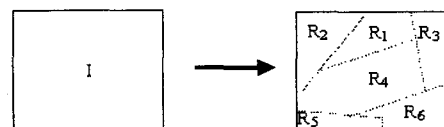


그림 2. 영역정의
Fig. 2. Region Definition.

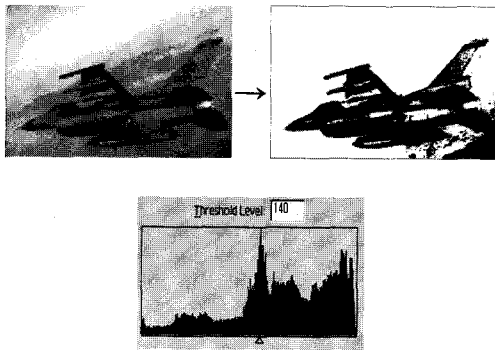


그림 3. 히스토그램을 이용한 이미지 세그멘테이션
Fig. 3. Image Segmentation using Histogram.

영역기반 이미지 세그멘테이션은 씨앗(seed) 영역을 찾아 낸 후, 그 씨앗 영역을 인접 픽셀 혹은 인접영역과 비교하여 영역을 확장하는 방법^[15]과 영역을 분할하거나 병합하는 과정을 통해 이미지를 나누는 방법을 사용한다^[16]. 영역기반 이미지 세그멘테이션은 씨앗 영역을 효과적으로 찾는 어려움이 있다.

반면 에지 기반 이미지 세그멘테이션 방법은 영역 기반 이미지 세그멘테이션이 씨앗 영역을 찾아내는 것과는 달리 에지를 먼저 잡아내는 방법을 사용한다. 추출한 에지는 에지 연결과정을 통해 영역을 생성하게 된다. 영역 기반과 에지 기반은 둘 다 지역적으로 인접한 픽셀들을 영역으로 판단하므로 영역 기반 이미지 검색을 위한 영역 생성이 가능하다. 본 논문에서는 에지 기반 방법으로 이미지 세그멘테이션 알고리즘을 제안하고 구현하였다.

범용적인 목적의 칼라 이미지 세그멘테이션은 많은 어려움이 존재한다. 이는 이미지 내에 많은 칼라 정보와 함께 질감 정보가 포함되어 있어 명확한 에지를 판단하기 어렵기 때문이다. 이런 문제로 인해 이미지 세그멘테이션은 특정 분야의 이미지(캐릭터, 적외선, x-레이 사진 등)에 대해서만 연구가 진행되어왔다. 특정분야의 이미지는 오브젝트와 배경간의 구별이 쉽고, 단순 칼라를 가진다는 장점이 있어 비교적 쉽게 세그멘테이션이 가능하다. 그러나 범용적인 분야에서의 이미지 세그멘테이션은 영역구분이 쉽지 않기 때문에 많은 선처리 과정이 필요하다.

4. 이미지 에지 추출

에지 기반 이미지 세그멘테이션에서 에지 성분을 정확히 잡아내는 것은 이미지 세그멘테이션 결과에 큰 영향을 미친다. 일반적으로 많이 사용되는 이미지 에지

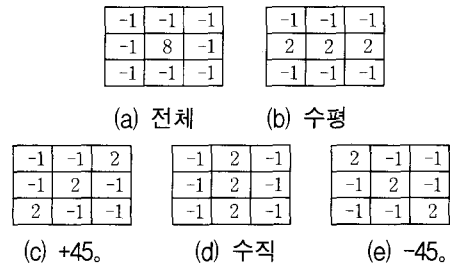


그림 4. 고주파 공간필터링 마스크
Fig. 4. High-Frequency Spatial Filtering Mask.

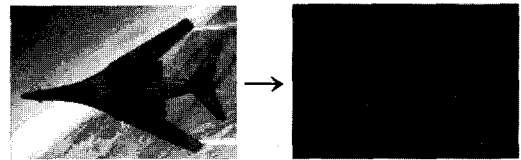


그림 5. 고주파 공간 필터링을 사용한 이미지 에지 추출
Fig. 5. Image Edge Extraction using High-Frequency Spatial Filtering.

추출 방법은 고주파 공간 필터(high-frequency spatial filter)방법을 사용한다^[18]. 이 방법은 3×3 크기의 마스크를 사용하여 에지 성분을 찾아내는 방법으로 8개 방향의 인접 픽셀의 밝기 정보 차를 이용하여 에지 여부를 판단하게 된다. 3×3 필터는 추출하고자 하는 에지의 특징에 따라 그림 5와 같이 다른 필터를 사용하는데 일반적으로 (a)의 형태를 사용한다. 그림 4의 (a)값을 이용할 경우 에지 여부를 판단하고자 하는 픽셀을 필터의 중심에 맞춘 다음 8개의 인접 픽셀 값에 -1 값을 곱하고 중심 픽셀에는 8을 곱한 값들을 합하여 총 마스크의 크기(9)로 나눈 값 중에서 큰 픽셀을 에지로 판단하게 된다. 그림 4의 마스크 (a)는 전체에 대한 에지를 추출하기 위해 사용하며, (b)는 수평 방향, (d)는 수직 방향의 에지 성분을, (c)는 45도 방향, (e)는 -45도 방향의 에지를 추출하기 위해 사용된다. 그림 5는 그림 4의 (a) 마스크를 사용하여 에지를 추출한 모습이다.

고주파 공간필터를 사용한 이미지 경계추출은 인접 픽셀과의 밝기 정보만을 비교하므로 이미지 내에 존재하는 질감성분을 구별하지 못하는 문제점을 가진다. 그러므로 질감성분이 존재하지 않는 단순한 이미지(오브젝트와 영역이 분명히 구분되는)의 경우 효과적인 경계추출이 가능하나, 질감성분을 가지는 복잡한 이미지에 대해서는 명확한 영역구분이 어렵다는 문제점을 갖는다.

이처럼 이미지 내의 질감 성분은 이미지 경계 추출에

많은 어려움을 준다. 질감성분이 포함된 이미지에서 질감성분에 둔감한 에지를 추출하기 위해서는 좀 더 큰 윈도우를 사용하는 방법과, 질감 성분에 둔감한 에지 추출 방법이 필요하다. 본 논문에서는 질감성분에 대해서 효과적인 경계를 찾기 위한 방법으로 Fisher의 선형 판별식(Fisher's Linear discriminant)^[11]을 이용하여 그레이 레벨(gray level)의 경계 이미지를 추출한다.

III. 제안한 칼라 이미지 세그멘테이션 알고리즘

본 논문에서 제안하는 칼라 이미지 세그멘테이션 방법은 먼저 에지를 추출한 후 에지를 연결하여 영역을 생성하는 에지 기반 세그멘테이션 방법을 따른다. 제안된 세그멘테이션 알고리즘의 순서는 그림 6과 같다.

이미지 검색과 마찬가지로 많은 수의 칼라 정보는 이미지 세그멘테이션 추출에 어려움을 제공한다. 그러므로 세그멘테이션은 이미지 양자화(image quantization)를 통해 이미지 정보가 적은 칼라로 줄인 후 세그멘테이션 과정을 수행하게 된다. 이미지 양자화는 적은 칼

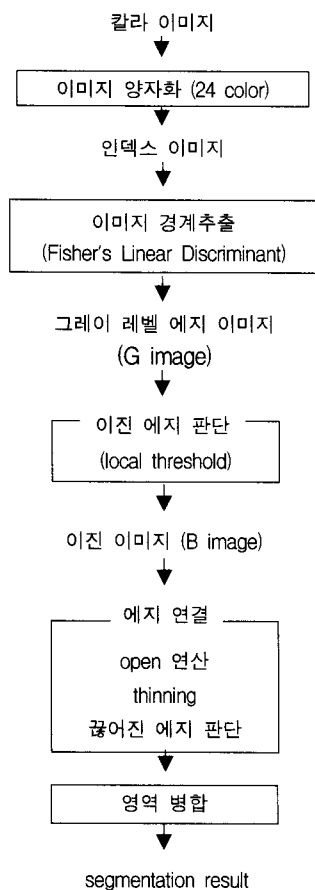


그림 6. 세그멘테이션 알고리즘

Fig. 6. Segmentation Algorithm.

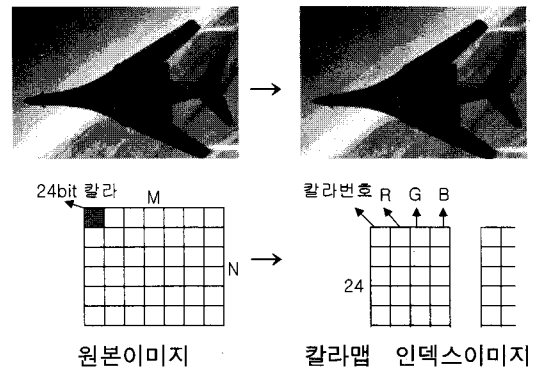


그림 7. 이미지 양자화

Fig. 7. Image Quantization.

라 수로 원본 이미지와의 유사성을 높이는 것을 목적으로 이미지 압축 및 전송에 연구되고 있다. 본 논문에서는 MATLAB의 RGB2IND함수를 사용하여 24 칼라의 이미지를 생성하였다.

양자화 된 이미지는 그림 7과 같이 칼라 정보를 저장하는 칼라 맵(color map)과, 칼라 번호로 이루어진 인덱스 이미지(index image)로 이루어진다. 이미지 양자화는 이미지 내에 주된 24개 칼라를 추출하여 칼라 맵에 저장하고, 원본 이미지를 칼라 맵에 저장된 24개의 칼라를 통해 표현하는 방법으로, 인덱스 이미지 각 픽셀의 값은 24비트의 칼라 정보 대신 칼라 맵의 칼라번호가 들어가게 된다. 이같이 생성된 인덱스 이미지는 에지를 추출하기 위한 정보로서 사용된다.

1. 이미지 경계 추출(Edge Extraction)

에지 기반 이미지 세그멘테이션에서 에지의 정확한 추출은 상당히 중요한 요소이다. 2장에서 설명한 것처럼 칼라 자연 이미지는 다양한 질감 성분의 존재로 인해 정확한 에지 판단의 어려운 문제가 있다. 이를 해결하는 방법은 인접 픽셀간의 정보를 비교하는 것이 아니라 좀 더 넓은 범위의 윈도우에서 그 칼라 정보의 분포를 통해 에지 여부를 판단해야 한다.

본 논문에서는 질감성분이 존재하는 영역의 에지를 효과적으로 찾기 위해 Fisher의 선형 판별식(Fisher's Linear discriminant)^[11]을 이용하였다. Fisher의 선형 판별식을 통해 생성된 값은 0에서 255사이의 값으로 확장하여 그레이 레벨의 에지 이미지로 표현할 수 있다.

가. Fisher의 선형 판별식

(Fisher's Linear Discriminant)

Fisher의 선형 판별식은 두 개의 클래스로 이루어진

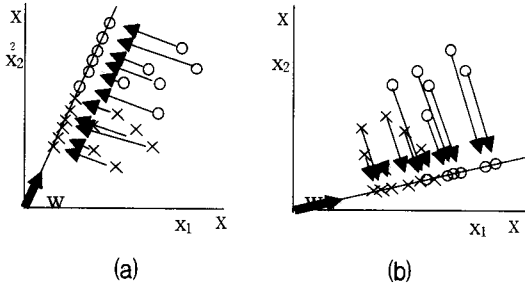


그림 8. 2 차원 클래스의 1차원 직선에 대한 맵핑
Fig. 8. Mapping 2-D Class into 1-D Line.

d 차원 점들의 집합이 1차원의 직선에 사상(mapping) 될 때, 두 클래스를 가장 잘 나누는 직선의 기울기를 찾기 위해 제안되었다. 그림 8은 두 클래스(O, X)를 가지는 2차원 점들이 각각 다른 1차원 직선에 사상되는 것으로, (b)의 직선보다 (a)의 직선이 두 개의 클래스를 잘 나누었음을 알 수 있다.

두 클래스를 잘 나누는 직선을 찾기 위해서는 1차원 직선에 사상된 두 클래스의 분할 정도를 판별할 수 있는 식이 필요하다. 이 식은 본 논문에서 사용한 예지 추출방법의 핵심이 된다. d 차원의 점이 기울기 벡터가 w 인 1차원 직선에 사상된 클래스의 분할 정도를 판단하는 함수(J(w))는 다음과 같은 과정으로 계산된다.

d 차원의 점 n개를 x_1, x_2, \dots, x_n 라 하자. n개의 점은 두 개의 클래스로 나누어지므로 각 클래스에 따라 두 개의 부분 집합 X_1, X_2 에 속하게 된다. (X_1 은 O 클래스, X_2 는 X 클래스) d 차원의 점은 방향 벡터 (vector) w에 의해 1차원의 스칼라(scalar) y 값으로 사상된다.

$$y = w^t x \quad \|w\| = 1 \quad (1)$$

이렇게 생성된 n개의 yi 값 역시 클래스에 따라 부분 집합 Y_1, Y_2 로 나뉘게 된다. 분할 정도를 판단하기 위해 d 차원의 클래스별 평균(m_i)과 1차원 직선상의 클래스별 평균(\tilde{m}_i) 그리고 표준편차의 성질을 가지는 \tilde{s}_i^2 를 구한다.

$$m_i = \frac{1}{n_i} \sum_{x \in X_i} x \quad (2)$$

$$\tilde{m}_i = \frac{1}{n_i} \sum_{y \in Y_i} y \quad (3)$$

$$= \frac{1}{n_i} \sum_{x \in X_i} w^t x = w^t m_i \quad (4)$$

$$\tilde{s}_i^2 = \sum_{y \in Y_i} (y - \tilde{m}_i)^2 \quad (5)$$

위 식을 이용해 최종적인 Fisher의 선형 판별 함수 J(w)를 구한다. 이는 1 차원 직선에 사상된 y 성분의 클래스별 평균의 차이와 y성분의 표준 편차와 유사한 \tilde{s}_i^2 의 합으로 표현된다.

$$J(w) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2} \quad (6)$$

(6)의 J(w)는 다음과 같은 계산에 의해 d 차원의 x값을 이용한 S_B 와 S_w 로 표현 할 수 있다.

$$S_i = \sum_{x \in X_i} (x - m_i)(x - m_i)^t \quad (7)$$

$$S_w = S_1 + S_2 \quad (8)$$

$$\begin{aligned} \tilde{s}_i^2 &= \sum_{x \in X_i} (w^t x - w^t m_i)^2 \\ &= \sum_{x \in X_i} (w^t x - m_i)(x - w^t m_i)^t \\ &= w^t S_i w \end{aligned} \quad (9)$$

$$\tilde{s}_1^2 + \tilde{s}_2^2 = w^t S_w w \quad (10)$$

$$\begin{aligned} (\tilde{m}_1 - \tilde{m}_2)^2 &= (w^t m_1 - w^t m_2)^2 \\ &= w^t (m_1 - m_2)(m_1 - m_2)^t w \\ &= w^t S_B w \end{aligned} \quad (11)$$

(11)에서

$$S_B = (m_1 - m_2)(m_1 - m_2)^t \quad (12)$$

$$J(w) = \frac{w^t S_B w}{w^t S_w w} \quad (13)$$

[11]에서는 \tilde{s}_i^2 을 scatter, S_w 는 클래스내의 분산 행렬(within-class scatter matrix), 그리고 S_B 를 클래스 간 분산 행렬(between-class scatter matrix)이라고 정의하였다. 결과적으로 J 값이 클 수록 두 개의 클래스가 잘 나누어졌음을 의미하고, 작은 J 값은 반대의 의미를 가진다. 이는 통계적으로 두 클래스가 서로 잘 나누어졌을 때는 두 클래스의 평균값의 차이가 크고, scatter의 합(S_w)이 작다는 점을 이용한 것이다.

위 식은 두 개의 클래스에 대한 정보를 보여준다.

클래스가 2개 이상일 경우에는 각 클래스간의 평균 차이를 구하기 위해서는 클래스 구분이 없는 전체의 평균을 먼저 구하고, 전체 평균과 각 클래스 평균간의 차이를 누적하여 SB 성분을 구한다. 2개 이상의 클래스에 대한 판별 함수 J(w)는 다음과 같이 계산된다.

$$S_w = \sum_{i=1}^c S_i \tag{14}$$

$$S_T = \sum_x (x - m)(x - m)^t \tag{15}$$

$$\begin{aligned} S_T &= \sum_{i=1}^c \sum_{x \in X_i} (x - m_i + m_i - m)(x - m_i + m_i - m)^t \\ &= \sum_{i=1}^c \sum_{x \in X_i} (x - m_i)(x - m_i)^t \\ &\quad + \sum_{i=1}^c \sum_{x \in X_i} (m_i - m)(m_i - m)^t \\ &= S_w + \sum_{i=1}^c n_i (m_i - m)(m_i - m)^t \\ S_B &= \sum_{i=1}^c n_i (m_i - m)(m_i - m)^t \end{aligned} \tag{16}$$

$$S_T = S_w + S_B \tag{17}$$

$$J(w) = \frac{w^t S_B w}{w^t S_w w} \tag{18}$$

나. Fisher의 선형 판별식

(Fisher's Linear Discriminant)의 응용

인덱스 이미지는 양자화된 칼라의 개수를 클래스로 가지는 2차원 점으로 나타난다. 본 논문에서는 24개의 칼라를 추출하였으므로 총 24개의 클래스를 가진 2차원 점에 대해 Fisher의 선형 판별식을 사용하게 된다. 이미지의 각 픽셀은 해당 크기의 윈도우에 포함되는 인접 픽셀과의 선형 판별식을 통해 에지 정도를 판단하게 된다. 사람이 느끼는 클래스의 분할 여부는 x축과 y축에 대해 클래스 분할 정도를 느끼게 되므로 윈도우의 x축에 사상된 클래스의 분할 여부와 y축에 사상된 클래스의 분할 여부를 이용해 전체 이미지 클래스의 분할 여부를 판단하게 된다. x축의 방향 벡터는 (1,0)이 되고 y축의 방향 벡터는 (0,1)이므로 J((1,0)) 과 J((0,1))의 값의 평균을 이용하여 에지 여부를 판단할 수 있다. 그런데 여기서 (0,1)과 (1,0)은 기저(basis)를 이루게 되므로 사

상 과정 없이 2차원 벡터(x,y)에 대해 바로 분할 정도를 판단할 수 있다. 그러므로 J(w) 함수는 2차원 클래스 이미지에 대해 기울기 벡터를 제거하여 다음과 같이 표현할 수 있다.

$$J = \frac{S_B}{S_w} = \frac{S_T - S_w}{S_w} \tag{19}$$

위 J 값의 범위는 정규화 되어 있지 않으므로 정확한 에지 정도를 판단하기 위해 정규화 과정이 필요하다. 0에서 1사이의 값으로 정규화 하기 위해 Sw 값을 ST 값으로 변경한다. ST 값은 SB와 Sw 값의 합과 같으므로 ST 값으로 변경하면 0에서 1사이의 값을 얻게 된다. 본 논문에서는 이 같이 정규화된 J 값을 Jnom이라고 한다.

$$J_{nom} = \frac{S_B}{S_T} = \frac{S_T - S_w}{S_T} = 1 - \frac{S_w}{S_T} \tag{20}$$

양자화된 이미지에서 각각의 픽셀에 대한 Jnom의 값은 픽셀이 영역간의 경계인지를 판단하는 기준이 된다. 그림 9는 5x5 크기 윈도우에 대한 Jnom 값 추출 과정을 보여준다.

		1	2	3	4	5
1	o	o	o	x	x	
2	o	o	o	x	x	
3	o	o	o	x	x	
4	*	*	*	*	x	
5	*	*	*	*	x	

$m_o = (2.0, 2.0)$ $m_x = (2.63, 4.63)$ $m_* = (4.5, 2.5)$
 $S_o = 12.0$ $S_x = 15.75$ $S_* = 12.0$
 $S_w = 39.75$ $S_T = 100.0$
 $J_{nom} = 0.6025$

o	x	o	x	o
*	o	*	o	*
o	x	o	x	o
*	o	*	o	*
o	x	o	x	o

(a) Jnom=0.0

o	o	o	x	x
o	o	o	x	x
o	o	o	x	x
*	*	*	x	x
*	*	*	x	x

(b) Jnom=0.6

o	o	x	x	x
o	o	x	x	x
o	o	x	+	+
*	*	*	+	+
*	*	*	+	+

(c) Jnom=0.752

o	o	x	+	x
o	o	+	x	+
o	o	x	+	x
o	o	o	x	+
o	o	o	+	x

(d) Jnom=0.375

그림 9. 2차원 양자화 이미지에 대한 Jnom 값
 Fig. 9. Jnom Value for 2-D Quantization Image.

그림 9의 o, x, *, +는 독립된 칼라 번호를 의미한다. Jnom 값은 (a)와 같이 3가지 칼라가 혼재하는 상황에도, 분포 정도를 비교하여 동일한 질감의 영역으로 판단하여 0인 값으로 나타나게 된다. (b)의 경우 중앙(3,3)의 픽셀이 에지일 가능성이 높음을 Jnom 값 0.6을 통해 보여주게 된다.

이처럼 양자화 된 인덱스 칼라 이미지의 각 픽셀은 윈도우에 대한 Jnom값을 계산하고 이를 그레이 스케일 이미지로 표현하기 위해 각 Jnom값에 255값을 곱한다. 본 논문에서는 이렇게 생성된 이미지를 G 이미지라 한다.

$$G(x,y) = Jnom(x,y) \times 255$$

이때 Jnom 값을 계산하기 위해 사용되는 윈도우의 크기는 에지 추출 정보에 큰 영향을 미친다. 윈도우 크기가 작을 경우에는 이미지의 세부적인 에지를 잡는 역할을 하고, 윈도우 크기가 클 경우에는 이미지의 전체적인 에지를 잡는 역할을 한다. 또한 윈도우 크기가 작으면 이미지 질감에 민감하여 불필요한 잡음이 발생하게 되고, 윈도우 크기가 크면 Jnom 값 계산 비용이 높아지고 추출된 에지가 지나치게 두껍게 되는 문제가 발생한다. 그림 10은 그림 7의 인덱스 이미지를 각 윈도우 크기 5×5, 7×7, 11×11, 25×25에 대해 G 값을 구한 후 그레이 이미지로 나타낸 것을 보여준다. 윈도우의 크기는 이미지 내의 오브젝트의 크기와 오브젝트의 명확성 정도에 의해 가변적이어야 한다. 그러나 이 같은 정보의 판단 역시 또 다른 문제점으로 작용하므로 본 논문에서는 데이터 이미지의 크기와 계산 비용을 고려한 윈도우 크기로서 11×11의 윈도우를 하였으며 윈도

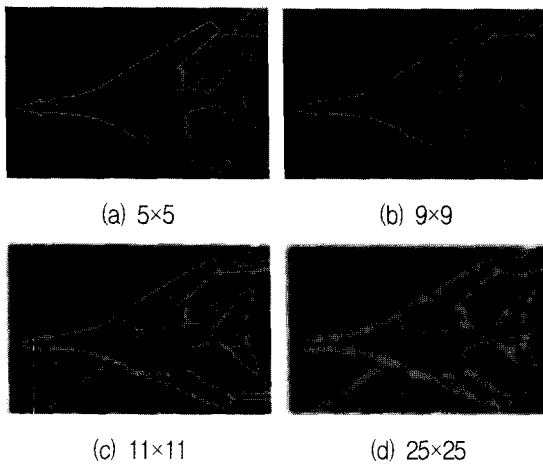


그림 10. 각 윈도우 크기에 대한 G 이미지
Fig. 10. G Image for Each Window Size.

우의 중심과 동일한 거리 내의 픽셀만을 이용하기 위해 원형 윈도우를 사용하였다.

2. 이진 에지 이미지(binary edge image) 생성

G 이미지는 그레이 레벨 이미지이므로 영역과 경계로 명확히 구분하기 위하여 이진 정보를 가지는 이미지로 변환하여야 한다. Fisher의 선형판별식으로 생성된 그레이 레벨 이미지는 지역적으로 에지의 밝기가 다르게 나타나기 때문에 윈도우 단위로 지역적인 임계치를 구해 에지 여부를 판단하는 것이 효과적이다. 본 논문에서는 먼저 최저 밝기의 임계치 LowT값을 두고 이보다 밝은 픽셀에 대해 밝기 우선 순위를 계산하여 밝기가 ε 내에 해당하는 픽셀을 에지로 잡고 나머지 픽셀들을 영역으로 판단한다.

$$B(x,y) = \begin{cases} 1, & \text{if priority}(G(x,y)) < \epsilon \times \text{window}^2 \\ & \text{and } G(x,y) > \text{LowT} \\ 0, & \text{else} \end{cases}$$

priority(G(x,y)) : 윈도우내에서 G(x,y) 밝기 순위
(0 ≤ priority(G(x,y)) ≤ window²)

본 논문의 구현에서는 이진 에지 이미지의 에지를 판단하기 위한 윈도우로 11×11 크기의 윈도우를 사용하였으며 최저 임계치 값인 LowT값으로 30, ε의 값으로 0.3을 사용하였다. 위의 윈도우 단위에 대한 임계치를 통해 생성된 모습은 그림 11과 같다. 그림 11에서와 같이 지역적인 임계치를 이용해 생성된 이진 이미지는 뚜렷한 에지의 형태와 함께 잡음성분이 많이 존재하게 된다. 이렇게 잡음이 포함된 이진 이미지에서 명확한 에지만을 남기고 불필요한 잡음을 제거하기 위해 형태 연산(morphological operation)인 열림(Opening) 연산을 사용한다.^[17~19]

열림(Opening)연산은 침식(Erosion)작업 후 팽창

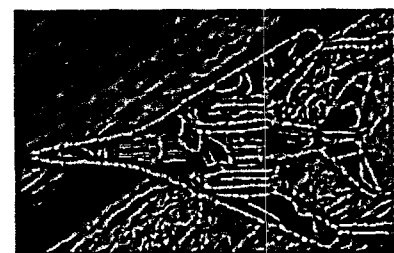
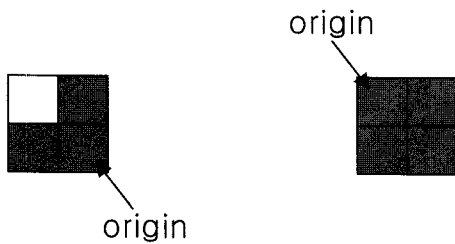


그림 11. 이진 이미지 생성 결과
Fig. 11. Result of Binary Image Generation.

(Dilation)작업을 연속적으로 수행하는 것을 말한다^[19]. 침식과 팽창의 수학적 형태론을 이용하여 영상을 변형한다. 이진 영상에 있어서, 문체의 집합들은 2차원 정수 공간 Z^2 의 구성요소로 되어 있다. 침식은 잡음성분을 제거하고 오브젝트를 줄여주는 역할을 하며, 팽창은 작은 틈을 채워주고 오브젝트를 확장해 주는 역할을 한다. 침식과 팽창 그리고 열림 연산의 수학적 정의는 다음과 같다.

$$\begin{aligned} \text{침식} : E &= B_E \otimes S = \{x, y | S_{x,y} \subseteq B_E\} \\ \text{팽창} : D &= B_D \oplus S = \{x, y | S_{x,y} \cap B_D \neq \emptyset\} \\ \text{열림} : O &= (S \otimes B_E) \oplus B_D \end{aligned}$$



침식 연산의 경우 침식 연산에 사용하는 블록(BE)의 원점을 이진이미지의 해당 픽셀에 맞춘 후 해당 픽셀과 위쪽 그리고 오른쪽 픽셀이 모두 1인 픽셀은 남기고 그렇지 못한 픽셀은 0으로 바꾸는 방식으로 이루어진다. 이와 같은 침식 연산은 그림 12에서 보는 것과 같이 떨어져 있는 작은 픽셀들을 제거하고 선의 형태를 줄여주는 역할을 한다. 이진 이미지에서 침식 연산은 잡음 제거를 하게 된다. 그러나 침식 연산은 연산 과정에서 필요한 정보까지 제거하는 경우가 발생하게 된다. 이 같은 부분을 복구해 주기 위해 팽창 연산을 수행하게 된다. 팽창 연산은 이미지의 해당 픽셀과 팽창 연산에서 사용할 블록(BD)의 원점을 맞추고 원점과 픽셀이 모두 1값을 가지면 왼쪽, 위쪽 그리고 왼쪽 위의 픽셀을 모두 1로 바꾸어 준다. 팽창 연산은 이진 에지를 두껍게

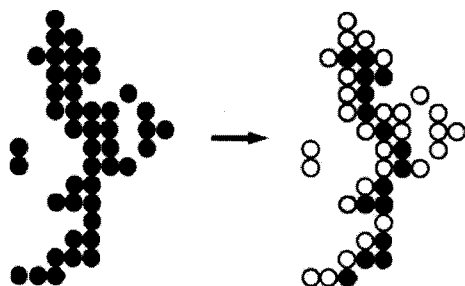


그림 12. BE 블록을 사용한 침식 연산 결과
Fig. 12. Result of Erosion Operation using BE Block.

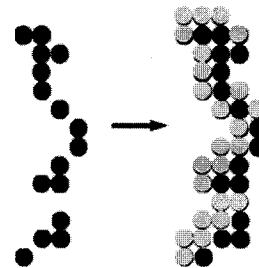


그림 13. BD 블록을 사용한 팽창 연산 결과
Fig. 13. Result of Dilation Operation using BD Block.

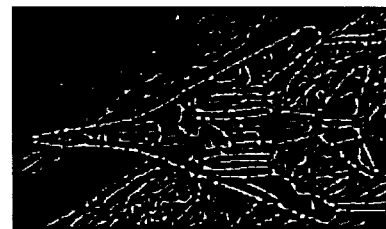


그림 14. 열림 연산 수행 결과
Fig. 14. Result of Opening Operation.

해 주면서 작은 픽셀 사이를 채워주는 역할을 하게 된다. 그림 12와 그림 13은 침식 연산과 팽창 연산을 통해 주요 에지 형태만을 잡고 불필요한 잡음을 제거해 주는 모습을 보여준다.

침식과 팽창의 연산을 연속적으로 수행하는 열림 연산은 이미지를 두 번 스캔하여 결과 이미지를 얻을 수 있다. 그림 14는 그림 11을 열림 연산을 통해 변환된 결과이다.

3. 에지 연결 (Edge Linking)

그림 14에서 보는 것과 같이 생성된 이진 에지 이미지는 연산과정에서 끊어진 부분을 가지게 된다. 이미지가 끊어져 있는 경우 정확한 영역 구별이 어려우므로 끊어진 에지를 찾아 인접 에지에 연결 시켜주어야 한다. 끊어진 에지를 찾기 위해서는 에지가 끊어졌는지를 판단할 수 있어야 한다. 에지의 굵기가 가변적인 경우 그 에지의 굵기로 인해 정확히 에지가 끊어졌는지를 판단하기가 어려운 문제점이 있다. 그러므로 에지의 굵기를 모두 1로 줄인 후 3x3 크기의 윈도우를 통해 에지의 끊어짐 여부를 판단한다.

가. Thinning

경계의 굵기가 가변적일 경우에는 경계가 끊어졌는지를 판단하기가 어렵다. 그러므로 끊어진 경계 판단 이전에 각 경계의 굵기를 1로 줄이기 위한 thinning 단계가 필요하게 된다. 본 논문에서 thinning 작업은 이진

이미지의 x축과 y축을 한번씩 체크하는 2 단계 알고리즘을 사용하였다.

먼저 x축으로 연속적인 1 값을 가지는 픽셀이 존재할 경우, 연속적인 1의 개수를 구한 다음, 연속적인 1의 중간 값만을 1로 남기고 나머지를 0으로 변경한다. 동일한 방법으로 y축에 대해서도 연속적인 1값의 중간 값만을 1로 남기는 방법으로 변경한다. 이렇게 각 축에 대해 생성된 두 개의 결과이미지를 합집합 연산을 통해 합하여 thinning 이미지를 생성한다. 다음은 본 논문에서 사용한 thinning 알고리즘을 보여준다.

```

=====
// 이미지의 크기가 N×M = n인 이진 이미지에 대해
for (int y = 0; y < N; y++)
    for (int x = 0; x < M-1; x++)
    {
        if (B[y][x] == 1 && B[y][x+1] == -1)
        {
            int n_Of_1 = 2;
            while(B[y][x+n_Of_1] != 0) { n_Of_1++; }
            for (int t = 0; t < n_Of_1; t++)
                { T_x[y][x+t] = 0; }
            T_x[y][x+n_Of_1/2] = 1;
        }
    } // x축에 대한 thinning
for (int y = 0; y < N-1; y++)
    for (int x = 0; x < M; x++)
    {
        if (B[y+1][x] == 1 && B[y][x] == -1)
        {
            int n_Of_1 = 2;
            while(B[y+n_Of_1][x] != 0) { n_Of_1++; }
            for (int t = 0; t < n_Of_1; t++)
                { T_y[y+t][x] = 0; }
            T_y[y+n_Of_1/2][x] = 1;
        }
    } // y축에 대한 thinning
for (int y = 0; y < N; y++)
    for (int x = 0; x < M; x++)
        { B[y][x] = T_x[y][x] | T_y[y][x] }
// x축과 y축에 대한 두 thinning 결과 이미지 합성
=====

```

그림 15는 위 thinning 알고리즘을 통해 경계의 굵기를 1로 줄인 결과 이미지이다.

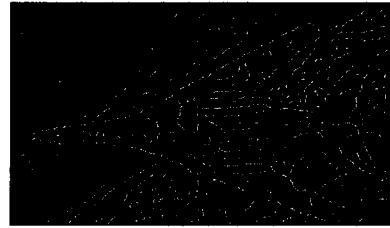


그림 15. Thinning 작업 수행후의 이진 이미지
Fig. 15. Binary Image after Thinning Processing.

나. 끊어진 에지 판단

굵기가 1로 변환된 이진 에지 이미지에서 끊어진 에지 판단은 3×3 윈도우를 사용해 이루어진다. 그림 16의 16가지의 형태를 가진 에지는 끊어진 점으로 판단하고 인접 에지에 연결한다.

에지 판단을 통해 끊어진 점으로 판정된 픽셀은 가장 가까운 에지를 찾아 연결하게 되는데 이때 가까운 픽셀을 찾는 방향은 인접 픽셀의 위치에 의해 결정된다. 예를 들어 그림 16의 (1), (9), (10)의 경우 동쪽(E)방향으로 가까운 에지에 연결하고 (2), (11), (12)는 북쪽(N), (3), (13), (14)은 서쪽(W), (4), (15) (16)은 남쪽(S), (5), (6), (7), (8)은 각각 SE, NE, NW, SW 방향으로 스캔 작업을 통해 찾아낸 에지와 직선으로 연결한다. 그림 17은 그림 16의 끊어진 에지의 형태에 따라 가까운 에지를 찾은 방향을 보여준다.

그림 18은 위의 에지 판단을 통해 끊어진 에지를 인접 에지로 연결한 이미지를 보여준다.

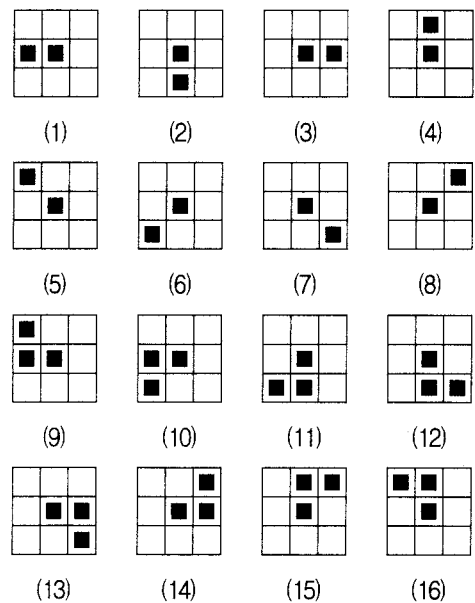


그림 16. 끊어진 에지
Fig. 16. Disconnected Edge.

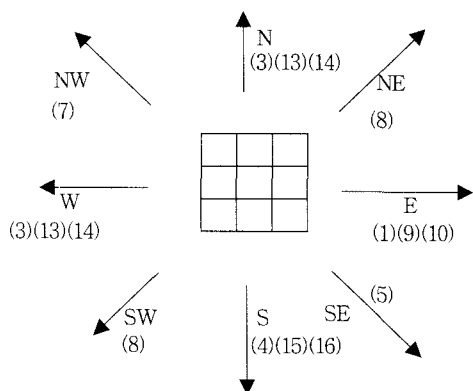


그림 17. 에지 연결 방향
Fig. 17. Direction of Edge Connection.

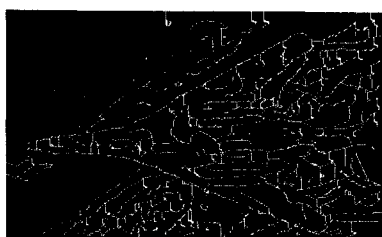


그림 18. 에지 연결 결과 이미지
Fig. 18. Image after Edge Connection.

이처럼 생성된 이미지는 에지 연결과정에서 불필요한 영역을 생성하므로 각 영역의 정보를 비교하여 유사한 영역을 합치는 영역 병합 과정을 거쳐 최종 세그멘테이션 이미지를 얻게 된다.

4. 영역 병합 (Region Merge)

에지로 둘러싸인 영역은 각 영역을 구분하기 위해 영역에 포함되는 픽셀 값에 영역 번호를 부여한다. 구분된 영역은 원본 이미지로부터 영역에 대한 정보를 추출하고 영역간의 병합을 위한 유사성 비교를 제공하게 된다.

픽셀에 영역 번호를 붙여주기 위해서는 먼저 이미지와 동일 크기의 배열(R[M][N])을 생성하여야 한다. 이 배열은 이미지의 칼라 정보가 아닌 영역 번호를 저장하게 된다. 최초 영역 번호는 1로 초기화하고 좌측 상단의 픽셀부터 차례대로 영역번호를 입력한다. 이때 해당 픽셀의 위와 왼쪽에 대한 이전 픽셀 값이 1이면 분리된 다른 영역으로 이동했다고 판단하고 영역번호를 증가시켜준다. 이와 같은 방법으로 이미지의 모든 픽셀에 대해 영역번호를 부여하게 된다. 자세한 알고리즘은 다음과 같다.

```

=====
//이미지의 크기가 N×M=n인 이진 이미지에 대해
int Label_Num=0;

for(y=1; y<N; y++)
  for(x=1; x<M; x++)
  {
    // 그림 19 (a)의 경우
    if (B[y][x]==0&&B[y][x-1]==1&&B[y-1][x]==1)
      {
        Label_Num++;
        R[y][x]=Label_Num;
      }
    // 그림 19 (b)의 경우
    if (B[y][x]==0&&B[y][x-1]==1&&B[y-1][x]==0)
      R[y][x]=R[y-1][x];
    // 그림 19 (c)의 경우
    if (B[y][x]==0&&B[y][x-1]==0&&B[y-1][x]==1)
      R[y][x]=R[y][x-1];
    // 그림 19 (d)의 경우
    if (B[y][x]==0&&B[y][x-1]==0&&B[y-1][x]==0)
      R[y][x]=R[y-1][x];
  }
=====

```

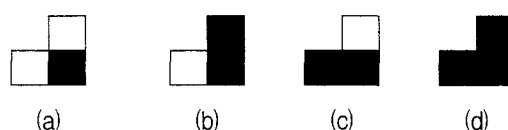


그림 19. 영역 구분을 위한 인접 픽셀 관계 ■=0 □=1
Fig. 19. Adjacent Pixel Relation for Region Boundary ■=0 □=1.

영역 라벨링 알고리즘은 영역이 볼록(convex) 하지 않을 경우 동일 영역임에도 다른 영역 번호가 주어질 수 있다. 이는 그림 19의 (d) 형태에서 일어나게 되는데 이런 경우 각각의 번호를 동일 영역으로 판단하고 저장한다.

이와 같이 생성된 영역 번호를 통해 각 영역의 크기 (영역을 이루는 픽셀의 수)와 원본 이미지의 평균 RGB 칼라를 구할 수가 있으며, 이렇게 구한 정보는 영역 합병을 위한 유사성 비교를 위해 리스트에 저장한다. 저장된 영역 정보는 칼라 정보의 유사성을 비교하여 유사한 영역간의 칼라를 동일하게 한 후 동일 칼라 사이의 에지를 제거하는 방법으로 영역을 병합해 간다. 영역간

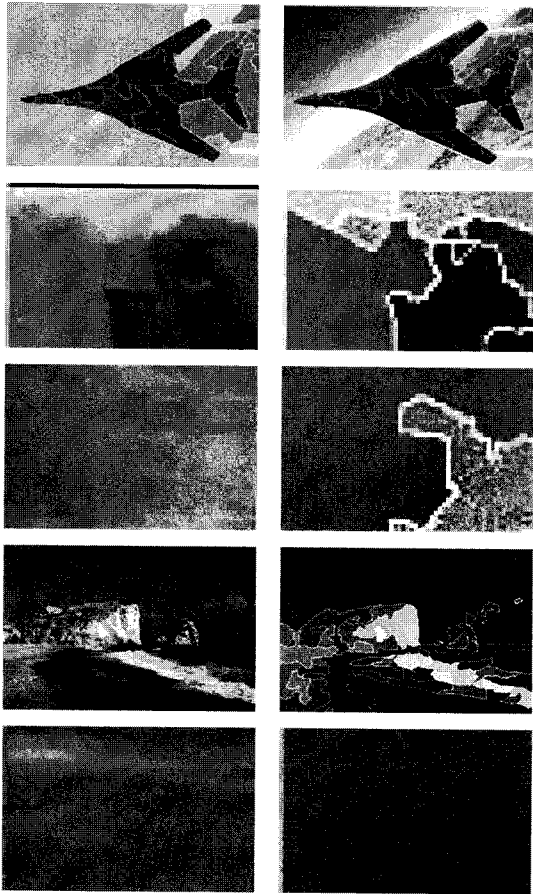


그림 20. 최종 세그멘테이션 결과
Fig. 20. Final Segmentation Result.

의 유사성 계산은 평균 RGB의 유클리드 거리계산으로 구한다.

$$dis(R1,R2) = \sqrt{(r_2 - r_1)^2 + (g_2 - g_1)^2 + (b_2 - b_1)^2}$$

본 논문의 구현에서는 영역의 평균 칼라의 유사 거리가 30이하인 영역을 동일하게 판단하였다. 동일한 칼라로 이루어진 영역은 3x3 크기의 윈도우를 통해 동일한 칼라 사이에 있는 에지를 제거하여 두 영역을 병합하게 된다. 이 같은 영역 병합 작업은 더 이상 병합되는 영역이 없을 때까지 반복하여 최종 세그멘테이션 이미지를 얻게 된다. 그림 20은 본 세그멘테이션 알고리즘을 통해 생성한 최종 세그멘테이션 이미지를 보여준다. 세그멘테이션 결과 이미지는 영역의 평균 칼라를 구해 표현하였다.

IV. 영역기반 이미지 검색 시스템 구현

본 논문에서 구현한 영역기반 이미지 검색시스템은

windows상에서 JAVA언어를 사용하였다. 입력데이터로 사용한 이미지는 corel photo이미지로 300x400에서 400x500 사이의 이미지 크기를 가지며 총 300 여개의 이미지 데이터가 입력되었다.

1. 이미지 데이터 입력

이미지 데이터 입력과정은 그림 21과 같다.

이미지가 입력되면 먼저 원본이미지를 별도의 이미지 데이터베이스에 저장하고, 세그멘테이션 과정을 통해 영역 분할 정보를 저장한다. 그리고 나누어진 각 영역에 대해 정보 추출 과정을 통해 영역 정보 데이터베이스에 저장한다. 원본 이미지는 질의 이미지나 결과이미지로서 출력되기 위해 사용되며, 영역 분할 정보는 질의 이미지에서 사용자가 원하는 영역 선택을 도와주는 역할을 한다. 영역 정보는 영역간의 유사성을 비교하기 위한 목적으로 사용되며 각 영역정보는 빠른 검색을 위해 영역의 평균 R, G, B를 키로 하는 3차원 인덱스(3-d tree[20])를 사용하여 저장하게 된다. 각 영역 정보는 평균 RGB와 함께, 영역을 포함하고 있는 이미지 이름과 영역의 위치정보, 영역의 크기정보로 이루어진다. 그림 22는 영역 정보를 가지고 있는 노드의 구조를 보여준다.

이미지 파일 이름은 해당 영역을 포함하는 이미지 이름을 의미하며, 영역 크기는 전체 이미지 크기에 대한 영역의 크기의 비율로 표현되고 결과이미지에서 영역

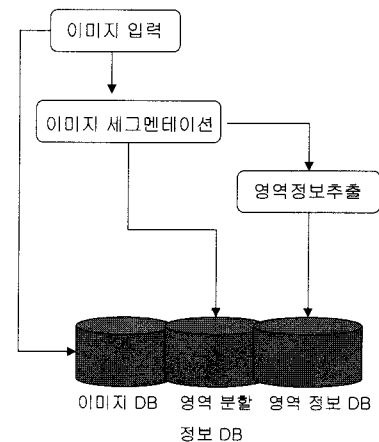


그림 21. 이미지 데이터 입력과정
Fig. 21. Image Data Input Processing.

R	G	B
filename	영역크기	영역위치

그림 22. 노드 구조
Fig. 22. Node Structure.

크기순으로 출력을 위해 사용된다. 영역 위치정보는 전체 이미지를 9개 방향(북서, 북, 북동, 서, 중앙, 동, 남서, 남, 남동)으로 나누어 영역의 무게중심이 속한 이미지의 위치를 1에서 9의 값으로 표현하여 저장한다. 이 정보는 위치정보를 포함한 영역의 유사성 비교를 목적으로 사용될 수 있다.

2. 사용자 질의

본 영역기반 이미지 검색은 최초 사용자가 데이터베이스 상의 질의 이미지를 불러오고 질의 이미지에서 사용자가 찾기 원하는 영역을 선택하여 질의를 하게 된다. 사용자가 이미지를 불러오면 동시에 영역 정보를 가지고 있는 파일을 불러오게 되므로 사용자가 이미지의 해당 영역을 마우스로 누를 때 자동적인 영역 선택이 이루어지게 된다. 그림 23은 사용자 질의에 대한 처리과정을 보여준다.

사용자는 자신이 원하는 검색의 정확성을 조정하기 위해 영역간의 거리를 입력할 수 있다. 이때의 유사 거

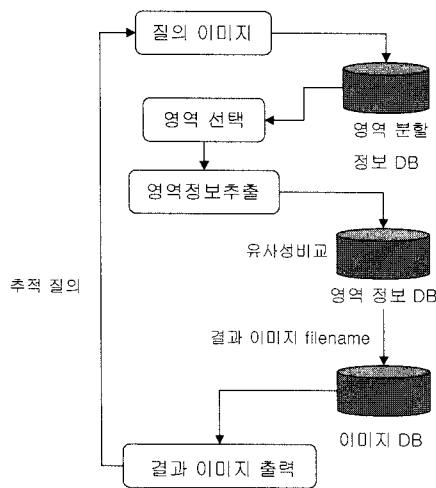


그림 23. 사용자 질의 처리과정
Fig. 23. User Query Processing.

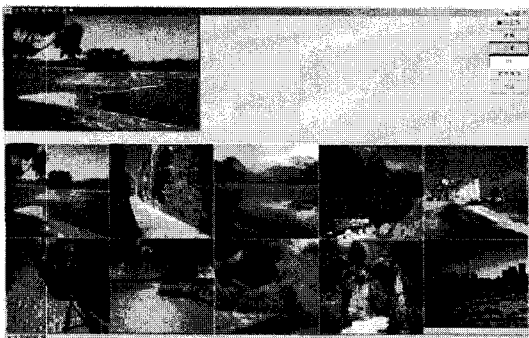


그림 24. 유사거리 35내의 유사영역 검색 결과
Fig. 24. Similar Region Retrieval within Similar Distance less than 35.

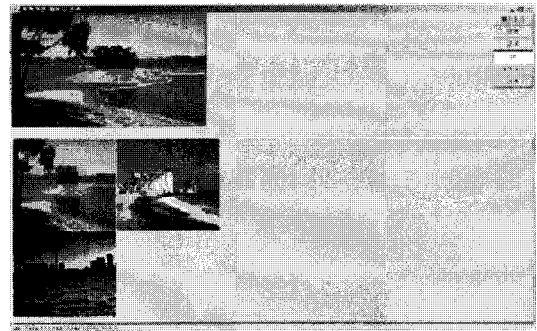


그림 25. 위치정보를 포함한 유사 영역 검색 결과 (유사 거리 35)
Fig. 25. Similar Region Retrieval considering Location Information (Similar Distance less than 35.)

리는 영역간의 평균 RGB의 최대 거리가 된다. 이 거리 내에 있는 영역에 대해서 유사한 순서대로 10개의 이미지를 출력하게 된다 (그림 24). 결과 이미지는 결과창에 맞는 크기로 출력되며 사용자는 결과로 출력된 이미지를 선택하여 반복적인 질의를 던질 수 있게 구현하였다. 또한 결과 이미지에서 영역 크기 순서의 출력과 위치 정보를 포함한 검색이 가능하게 구현하였다 (그림 25).

질의 응답 시간은 질의 이미지에 따라 다르지만 대략 수초에서 수십 초 정도 걸렸다. 질의 응답에 영향을 주는 주요 요소는 유사 거리이다. 유사 거리를 크게 할 때 유사한 이미지를 많이 출력하게 되며 이에 따라 응답하는데 시간이 더 걸렸다. 유사 이미지의 출력 개수를 가변적으로 하여 응답 시간을 조절하였다.

V. 결론 및 향후과제

전체 이미지 정보를 효과적으로 추출하고 인덱싱 하는 것은 많은 제약이 따른다. 이 같은 문제를 영역 기반 검색을 통한 영역 단위의 검색으로 해결하려 하였다. 영역 기반 검색은 동일한 성질의 영역을 나누는 이미지 세그멘테이션의 성능에 많은 영향을 받는다. 일반적인 칼라 자연 이미지는 질감 정보와 많은 칼라 정보로 인해 정확한 영역 분할의 어려운 문제점을 보인다.

본 논문에서는 효과적인 영역기반 이미지 검색을 위해 이미지의 질감 성분에 둔감한 에지 추출 방법을 이용해 이미지를 나누는 세그멘테이션 방법을 제안하고 이를 이용한 영역기반 이미지 검색 시스템을 구현하였다. 본 논문에서 사용한 에지의 추출방법은 윈도우상에 존재하는 적은 수의 칼라에 대해서 그 평균과 표준편차

의 특징을 갖는 분포를 이용해 질감에 둔감한 에지를 찾아내는 방식을 사용하였다. 에지 추출에 사용되는 윈도우의 크기는 세그멘테이션 결과에 큰 영향을 미치는 요소이다. 본 논문의 구현에서는 11×11의 고정된 크기의 윈도우를 사용하였지만 이는 이미지 내에 존재하는 오브젝트의 크기와 선명도에 달라질 필요가 있다. 또한 영역 병합 등에 사용된 다양한 인자 값 역시 모든 이미지에 동일하게 적용하는 것은 효과적이지 못하다. 따라서 이미지의 기본적인 특징을 찾아내 윈도우와 다른 임계치 값을 가변적으로 판단할 수 있는 방법이 요구된다. 또한 여러 가지 단계를 통해 이미지 세그멘테이션 과정을 거치므로 데이터 입력과정에 많은 비용이 들게 되는 문제점이 존재하는데, 위와 같은 문제점을 고려하여 좀더 경제적이며 정확성 있는 세그멘테이션 방법이 필요하다.

참 고 문 헌

- [1] Chang N. S. and Fu K. S., "A Relational Database System for Images", Technical Report RT-EE, Purdue University, pp. 79-28, 1979.
- [2] Chang N. S. and Fu N. S. "Query-By Pictorial Example", Proc. of IEEE Trans. On Software Engineering, 1980.
- [3] Chang S. K., Yan C. W., Dimitroff, C. and Timothy Arndt., "An Intelligent Image Database System", Proc. of IEEE Trans. on Software Engineering, 1988.
- [4] Niblack W., Barber R. and et al. "The QBIC Project Querying Images by Content Using Color, Texture and Shape". Proc. of SPIE Storage and Retrieval for Image and Video Databases, 1994.
- [5] Androustos D., et al, "Image Retrieval Using Directional Detail Histograms", Proc. of SPIE Storage and Retrieval for image and Video Databases VI, pp.129-137, 1998.
- [6] Natsev A., Rastogi R. and Shim K. S., "WALRUS: A Similarity Retrieval Algorithm for Image Databases," Proc. of SIGMOD Conference, pp. 395-406, 1999.
- [7] Flickner M. Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Huang Q., Byron Dom, Monika Gorkani, Jim Hafine, Denis Lee, Dragutin Petkovic, David Steele, and Peter Yanker, "Query by Image and Video Content: The QBIC System", Proc. of IEEE Computer, 1995.
- [8] Deng, Y. and Manjunath, B. S., "An Efficient Low-Dimensional Color Indexing Scheme for Region-Based Image Retrieval", Proc. of IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), pp. 3017-20, 1999.
- [9] Chua, T. S., Lim, S. K. and Pung, H. K., "Content-Based Retrieval of Segmented Images", Proc. of ACM Multimedia, pp 211-218, San Francisco, 1994.
- [10] R. Weber, H.-J. Schek, and S. Blott, "A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces", Proc. of VLDB, pp. 194-205, 1998.
- [11] Duda, R. O. and Hart, P. E., Pattern Classification and Scene Analysis, John Wiley & Sons, 1970.
- [12] Bonsiepen, L. and Coy W., "Stable Segmentation Using Color Information", Proc. of Computer Analysis of Images and Patterns, ed. R. Klette, Proc. of CAIP'91, Dresden, pp.77-84, September 1991.
- [13] Ohlander, R., Price K., and Reddy D.R., "Picture Segmentation Using a Recursive Region Splitting Method", Proc Computer Graphics and Image Processing, vol.8, pp. 313-333, 1978.
- [14] Markus S., and Markus O., "Similarity of Color Images", Proc. of SPIE Storage and Retrieval for Image and Video Databases, 1995.
- [15] Chang, S. F., and Smith, J. R., "Finding Images/Video in Large Archives", D-Lib Magazine, 1997.
- [16] Deng, Y. Manjunath, B. S. and Shin, H. "Color Image Segmentation", Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, 1999.
- [17] Skarbek, W., Methods for Digital Image Representation, Akademicka Oficyna Wydawnicza PLJ, Warszawa (in Polish) 1993.
- [18] Rafael C. G., and Richard E. W., Digital Image Processing, Addison Wesley Longman 1992.
- [19] Lybanon, M., Lea, S. and Himes, S., "Segmentation of Diverse Image Types Using Opening and Closing", Proc. of IMMC, pp. 135-138, 1999.
- [20] Beckley, D. A., Evens, M. W. and Raman, V. K., "Multikey Retrieval from K-d Trees and Quad-Trees", Proc. of SIGMOD Conference pp. 291-301, 1985.

— 저 자 소 개 —

황 환 규(정회원)

대한전자공학회 논문지

제41권 CI편 제2호 참조