2000년 한국정보처리학회 추계 학술발표논문집 제 7 권 제 2 호

컬러 히스토그램의 공간 정보를 이용한 영상 검색에 관한 연구

윤호섭, 배영래, 양영규
한국전자통신연구원 영상처리부
e-mail : yoonhs@etri.re.kr

A Study on Image Retrieval Using Space Information of Color Histogram

Ho-Sub Yoon, Younglae J. Bae, Young-Kyu Yang
Division of Image Processing, ETRI

요 약

영상에서 나타난 컬러 정보를 이용해 검색을 수행하고자 하는 연구 방법에서 일반적으로 적용되어 사용되어진 방법 중 하나로 컬러 히스토그램의 유사성을 이용한 매핑을 들 수 있다. 이 방법의 장점은 쉽게 영상에서 나타난 컬러 정보를 그룹화하여 매핑 할 수 있게 해주나, 전체 영상에서 나타난 컬러의 양 정보만을 제공할 뿐, 각각의 컬러에 대한 공간적 정보를 제공하지는 못한다. 즉, 동일한 컬러가 한 곳에 집중해서 나타난 영상과 분산되어 나타난 영상이 동일한 컬러 정보로 매핑되므로 원치 않는 검색 결과를 얻게 된다. 이러한 문제점을 해결하기 위해 기존의 방법은 컬러 히스토그램에서 얻어진 컬러 정보와 별개의 공간 정보 추출 알고리즘을 통해 얻어진 정보를 결합(fusion) 하여 문제를 해결하였다. 본 연구에서는 이러한 단점을 해결하기 위해 각각의 히스토그램의 빈(bin)들에 컬러에 대한 양 정보만이 아니라 컬러의 중심점과 분산 값 정보를 구하여 이를 영상 검색을 위한 매핑 정보로 사용한다.

1. 서론

영상 데이터베이스의 검색 방법은 문자 기반(text based), 내용 기반(content based), 의미론적 기반(semantic based)의 3 부류로 나눌 수 있다[1]. 내용 기반 검색에서 사용하는 특징들로는 모양(shape), 질감(texture), 컬러(color) 등이 주로 사용된다. 이때 컬러는 모양과 질감보다 특징을 추출하기가 용이하고, 사용자가 시각적인 예를 제시하는 방식(query by example)에 적용하기 적합하다[2].

기존의 컬러 색인 시스템은 주로 컬러 히스토그램의 빈(bin) 정보만을 이용하고 있다. 이때 두 컬러 히스토그램간의 차이를 이용한 유사도 측정 방법은 실제 사람의 인지적 유사도(perceptual similarity)의 측정결과와 동일한 결과를 갖지 못하는 경우가 종종 발생한다. 이는 히스토그램의 컬러 특징값 즉, 각 빈들의 값이 전역(global) 특징 정보를 나타내기 때문으로 전역적 특징 정보만으로는 정확히 원하는 내용을 가진 영상을 검색하기 어렵다. 즉, 전역 특징은 영상의 회전이나 약간의 위치변화에는 영향을 받지 않는 장점이 있으나, 공간 정보를 전혀 포함하지 못한다는 것이 큰 단점이다. 이러한 공간정보를 포함하지 못하는 전역 특징의 특성 때문에 컬러 정보만을 이용하여 검색할 경우 검색결과에서 원치 않은 영상 에러(false positive error)를 유발 할 수 있다.

영상에서 공간정보를 추출하는 것은 매우 어려운 작업이기 때문에 기존의 방법은 주로 영상을 부 영역(sub-region)으로 나누어 각 부 영역에 대해 전역 특징을 구하는 방법들이 제안되고 있다. 이러한 방식을 컬러 레이아웃(color layout)색인 방법이라고 한다. 가장 간단한 컬러 레이아웃 색인방식으로는 영상을 균일크기의 블록으로 나눠 각 블록에서 컬러 특징을 추출하는 방법이 있으며, 이러한 방식은 컬러

변화가 심한 이미지에서는 적합하지 않다. 다른 방식으로는 영상을 분할해서 각 영역에 대한 특징을 구하는 방식이 있다. 실제로 영상의 객체들에 대해 분할 하는 것은 무척 어려운 작업이므로 완전 자동화 시스템을 구현하기 어렵다. 실제 시스템에서는 반 자동 또는 수동적(manual)으로 사용자가 객체를 지정하는 분할 작업이 사용되고 있다. 요즘 새로운 컬러 레이아웃 색인 방법으로 웨이블렛 변환(wavelet transform)을 사용한 방법들이 제안되고있다[3].

이와 같이 기존의 방법은 컬러 히스토그램에서 전역적인 컬러 정보와 영역을 분할하여 각 부 영역에서 공간정보를 추출한 후, 이 두 결과를 합성하여 주로 검색이 이루어 졌다. 이러한 방법의 문제점은 두 구해진 정보간의 관계가 전혀 없으므로 어떻게 이를 종합할 것인가에 대한 문제와 전혀 별개의 두 특징을 두 번 추출하는 과정이 요구되어 비 효율적이다. 본 연구에서는 이러한 단점을 해결하기 위해 컬러 히스토그램을 생성하면서 각각의 빈들이 그 컬러에 대한 중심점과 분산 값을 갖도록 함으로서 컬러정보와 공간 정보의 결합을 가능하도록 하였다.

본 연구의 구성은 다음과 같다. 먼저 1장에서는 서론에 대해 기술하였고 2장에서는 공간정보와 결합된 컬러 특징값 검출 방법을 기술한다. 3장에서는 구해진 특징값의 유사성 검출 방법에 대해 기술하고, 마지막으로 4장에서 결론에 대해 논한다.

2. 특징값 검출

본 장에서는 영상 데이터베이스 내에 존재하는 수많은 영상 정보를 검색하기 위해 입력 영상에서 컬러 히스토그램을 구하고, 또한 각 빈들의 중심 위치 및 분산 값을 추출하는 방법을 기술한다. 이를 위해 그림 1 과 같은 단계를 거치게 된다.

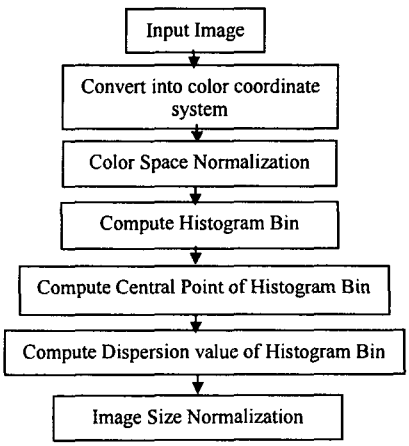

그림 1. 특징값 검출을 위한 단계

그림 1 에서 첫번째 단계는 입력 영상을 하나씩 주사하면서 각각의 좌표에 대한 컬러 값을 대응되는 빈에 누적하여 특징을 측정한다.

이때, RGB 의 3 개의 축을 가진 컬러 히스토그램은 처리해야 할 정보의 양이 많으므로 이를 YIQ, YUV, HSV 등의 컬러 좌표계로 변환한 후, 이중 IQ 나 UV 등과 같은 밝기를 제외한 2 차원 컬러 히스토그램으로 변환하여 특징의 개수를 줄인다. 컬러 특징을 줄이는 또 하나의 방법으로는 스케일링 변수를 사용한다. 즉 256 단계의 컬러 정보를 128 단계로 축소시키기 위해선 단지 RGB 로부터 얻어진 컬러 정보를 스케일 팩터(SCALE_RGB) 2 로 나누어 주면 된다.

이러한 처리는 컬러 공간에서 특징의 개수를 감소시키게 되나, 이는 미세한 컬러 정보를 그룹화하여 평활화 된 하나의 대표 컬러로 변환 시키는 결과를 갖는다. 이러한 대표 컬러를 사용하는 것은 유사성 척도에 사용될 특징들의 양을 줄여주며, 잡음등에 영향을 받지 않는 장점이 있으나, 영상의 전체 컬러 해상도를 떨어 뜨려 복잡한 영상에서 정밀한 처리를 어렵게 할 수도 있으므로 응용에 따라 스케일링 정도를 결정한다.

아래의 알고리즘은 히스토그램에서 컬러 빈의 누적 및 각 x, y, x^2, y^2 값의 누적 과정을 보여주며, 이때 I, Q 히스토그램은 전체 컬러 해상도 만큼 생성되는 것이 아니라 단지 그 컬러가 한번이라도 발생되었을 경우에만 생성되어 불필요한 메모리의 사용을 줄이고 유사도 매핑 시의 처리 속도를 빠르게 한다.

```
for (y=0; y<Height; y++){
  for (x=0; x<Width; x++){
    // (x,y) 위치의 RGB 컬러값을 가져온다.
    rgb= pDib->GetColor(x,y);
    II = (int)((0.596*rgb.rgbRed -0.275*rgb.rgbGreen -
      0.321*rgb.rgbBlue));
    // Scaling
    II=  (int)(II/SCALE_RGB + 0.5);
    QQ = (int)((0.212*rgb.rgbRed -0.523*rgb.rgbGreen
      +0.311*rgb.rgbBlue));
    // Scaling
    QQ = (int)(QQ/SCALE_RGB + 0.5);
    found = FALSE;
    for (k=0; k< Model_bin; k++){
      if (IQ_Histogram[k].i == II &&
        IQ_Histogram[k].q == QQ) {
            found = TRUE;
            IQ_Histogram[k].cnt++;
          // 히스토그램의 x 좌표 누적
            IQ_Histogram[k].Sumx +=x;
          // 히스토그램의 x^2 좌표 누적
            IQ_Histogram[k].Sum2x +=(x*x);
          // 히스토그램의 y 좌표 누적
            IQ_Histogram[k].Sumy +=y;
          // 히스토그램의 y^2 좌표 누적
            IQ_Histogram[k].Sum2y +=(y*y);
            break;
```

```
        }
    }
    if (!found) {
        Allocate_New_IQ_Histogram();
        Model_bin++;
    }
  }// end of for x
}
```

위의 단계를 거치면 전체 영상으로부터 각 I, Q 히스토그램에 대한 컬러 컬러 빈의 누적 및 각 x, y, x^2, y^2 값이 구해진다. 각 빈들의 평균과 분산은 아래와 같은 공식으로 간단히 계산된다. 이를 들면 x 좌표의 평균 m_x이라 하고 분산을 $\delta^2{}_x$이라하면

$$m_x = \frac{1}{n}\sum_{i=1}^{n} x_i \ , \ \delta^2{}_x = \frac{1}{n}\sum_{i=1}^{n} x^2 - m^2$$

이 된다.

위의 공식을 사용하여 생성된 컬러 히스토그램에 대해 적용하면 컬러 히스토그램은 각 컬러 값의 누적 값과 그 컬러의 중심점(m_x m_y) 및 분산값 ($\delta^2{}_x$, $\delta^2{}_y$)을 구할 수 있다. 이때 구해진 누적 값과 중심점 및 분산 값은 전체 영상의 크기로 나누어 정규화 시켜야 한다.

3. 유사성 척도

2 장에서의 특징 추출 방법을 이용해 영상데이터베이스에 수 많은 영상을 저장한 후, 사용자는 입력 영상을 제공하고 가장 유사한 영상을 검색하라는 질의를 입력하게 된다. 이에 대해 본 연구에서는 유사성 측정 척도 *Similarity*를 아래와 같이 정의하였다.

$$Similarity = W_1 S_{color} + W_2 S_{xm} + W_3 S_{ym} + W_4 S_{xd} + W_5 S_{yd} + W_6 S_{count}$$

위의 유사성 측정치 값은 기본적으로 정의된 여섯개의 특징값과 여섯개의 가중치값으로 계산된다. 이때 가중치 값은 유사성 매칭시 어떠한 특징이 영상을 검색하는지에 가장 적절한지에 따라 결정되어 지는데 이는 응용 시스템의 성격에 따라서 달라 질 수 있다. 또한 각각의 특징값들을 구하는 공식은 부록(Appendix)에 정의되어 있으며 전체 값들이 0.0 에서 1.0 사이의 값을 갖도록 정규화 되어 처리한다. 위의 공식에서 유사성 척도 S_{color} 는 기존의 컬러 히스토그램의 빈들의 값을 의미한다. 즉, 각 컬러 히스토그램에 얼마나 많은 컬러 정보가 포함되어 있나를 결정한다. 또한 S_{xm}는 해당 컬러 히스토그램의 x 좌표의 중심점을 S_{ym} 은 y 좌표의 중심점을 의미한다. 마찬가지로 S_{xd} 는 컬러 히스토그램의 x 좌표들이 얼마나 분산되어 나타났는가를 의미하며 S_{yd} 는 y 좌표들의 분산값을 의미하게 된다. 마지막으로 S_{count} 는 S_{color} 가 영상에서 동일한 컬러를 갖는 빈들의 정보량을 비교한다고 할 때, S_{count} 는 단지 동일한 컬러 정보를 갖는 빈들이 얼마나 존재하는지을 비교한다.

이와 같이 유사성 측정치가 구해지면 전체 메타데이터 베이스에서 유사도가 작은 순서로 검색하여 검색 영상을 사용자에게 출력한다. 그림 2 에서 유사성 척도를 이용한 검색 단계를 보여준다.

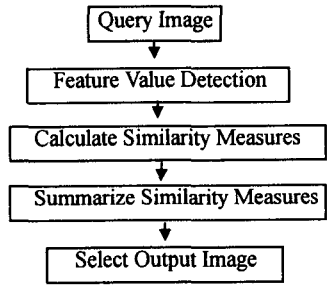

그림 2. 유사성 척도를 이용한 검색 단계

4. 결론

본 연구는 기존의 컬러 히스토그램을 이용한 검색 방법이 공간정보를 이용하지 않아 별개의 공간정보 추출 알고리즘을 사용해야 하는 단점을 해결하였다. 즉, 각각의 컬러 히스토그램의 컬러 분포 정보와 그 분포에 해당하는 빈의 영상에서의 위치 정보 및 분산 정보를 동시에 추출하고, 이를 검색 유사성 척도로서 종합적으로 이용함으로써 좋은 검색 결과를 얻을 수 있었다. 이때 위치 정보 및 분산 정보를 히스토그램 생성시 한번에 수행 함으로써 별도의 처리 시간이 필요치 않고, 컬러 정보와 위치 정보 및 분산 정보를 가중치를 이용하여 연계함으로써 기존의 검색 방법에 비해 효율적인 결과를 갖게 된다.

또한 컬러 히스토그램의 공간 정보는 영상 검색 뿐만 아니라 영역 합병 방법의 영상 분할(segmentation)시 Seed 정보로 사용될 수 있으며, 컬러 영상 이해 시스템의 특징으로도 사용 가능하며 이 밖의 여러 영상 처리 분야에서 기본적인 특징 벡터로 사용할 수 있다.

참고문헌

[1] John R.Smith and Shih-Fu hang, "Tools and Techniques for Color Image Retrieval," SPIE Storage and Retrieval for Still Image and Video Databases IV, Vol 2670, pp.426-437, 1996, Feb.

[2] W. Niblack and etc, "The QBIC project: Querying Images By Content Using Color, Texture, and shape," SPIE Storage and Retrieval for Still Image and Video Databases I, Vol. 1908, pp.173-187, 1993, Feb.

[3] James Wang and etc, "Wavelet-based image indexing techniques with partial sketch retrieval capability," Proceeding of 4th Forum on research and technology advances in digital libraries, pp. 13-24, 1997, May.

부록(Appendix)

$$S_{color} = \sum_{k=1}^{input\_bin} \sum_{l=1}^{model\_bin} f_{color}(k,l)$$

$$f_{color}(k,l) = \begin{cases} if(H^{Input} i_k == H^{Model} i_l \ and \ H^{Input} q_k == H^{Model} q_l) \\ \qquad : abs(H^{Input}(i,q)_k.count - H^{Model}(i,q)_l.count) \\ otherwise \quad : 0 \end{cases}$$

$$S_{xm} = \sum_{k=1}^{input\_bin} \sum_{l=1}^{model\_bin} f_{xmean}(k,l)$$

$$f_{xmean}(k,l) = \begin{cases} if(H^{Input} i_k == H^{Model} i_l \ and \ H^{Input} q_k == H^{Model} q_l) : abs(H^{Input}(i,q)_k.xm - H^{Model}(i,q)_l.xm) \\ otherwise \quad : 0 \end{cases}$$

$$S_{ym} = \sum_{k=1}^{input\_bin} \sum_{l=1}^{model\_bin} f_{ymean}(k,l)$$

$$f_{ymean}(k,l) = \begin{cases} if(H^{Input} i_k == H^{Model} i_l \ and \ H^{Input} q_k == H^{Model} q_l) : abs(H^{Input}(i,q)_k.ym - H^{Model}(i,q)_l.ym) \\ otherwise \quad : 0 \end{cases}$$

$$S_{xd} = \sum_{k=1}^{input\_bin} \sum_{l=1}^{model\_bin} f_{xdeviation}(k,l)$$

$$f_{xdeviationr}(k,l) = \begin{cases} if(H^{Input} i_k == H^{Model} i_l \ and \ H^{Input} q_k == H^{Model} q_l) : abs(H^{Input}(i,q)_k.xd - H^{Model}(i,q)_l.xd) \\ otherwise \quad : 0 \end{cases}$$

$$S_{yd} = \sum_{k=1}^{input\_bin} \sum_{l=1}^{model\_bin} f_{ydeviation}(k,l)$$

$$f_{ydeviationr}(k,l) = \begin{cases} if(H^{Input} i_k == H^{Model} i_l \ and \ H^{Input} q_k == H^{Model} q_l) : abs(H^{Input}(i,q)_k.yd - H^{Model}(i,q)_l.yd) \\ otherwise \quad : 0 \end{cases}$$

$$S_{count} = \sum_{k=1}^{input\_bin} \sum_{l=1}^{model\_bin} f_{count}(k,l)$$

$$f_{count}(k,l) = \begin{cases} if(H^{Input} i_k == H^{Model} i_l \ and \ H^{Input} q_k == H^{Model} q_l) : 1 \\ otherwise \quad : 0 \end{cases}$$

위의 수식에서 $H^{input}i_k$ 는 메터데이터베이스부터 입력된 컬러히스토그램에서 k 번째 I 컬러특징 벡터값을 의미하고, $H^{input}q_k$ 은 k 번째 Q 컬러 특징 벡터값을 의미한다. $H^{input}(i,q)_k.count$ 는 k 번째 I 및 Q 값을 갖는 2 차원 컬러 히스토그램의 누적 빈도 특징 값을 의미하고, $H^{input}(i,q)_k.xm$ 은 k 번째 I 및 Q 값을 갖는 2 차원 컬러 히스토그램의 x 좌표의 평균값을 갖는 특징 벡터를 의미하며, ym 은 y 좌표의 평균값을 갖는 특징 벡터, xd 는 x 좌표의 분산값을 갖는 특징벡터, yd 는 y 좌표의 분산값을 갖는 특징벡터값을 의미한다. 마찬가지로 $H^{Model}i_k$ 는 처리 영상이 입력 영상인 것 이외에 나머지 표기는 동일하다.