

히스토그램 블록 기반 유사 영상 맵 생성 및 영상 합성 알고리즘

유재성, 이은별, 김하린, 이재만, 이의상, 김규현
경희대학교

jsyu624@khu.ac.kr, dmsquf3212@naver.com, khukhr6239@hanmail.net,
Hellokangman@hanmail.net, euisang91@khu.ac.kr, kyuheon.kim@khu.ac.kr

Histogram Block-based Similarity Image Map and Image Stitching Algorithm.

Jaeseong Yu, Eunbyeol Lee, Harin Kim, Jeman Lee, Euisang Lee, Kyuheon Kim
Kyunghee University

요 약

본 논문에서는 다수의 영상을 빠르고 오류 없이 정합하기 위하여 정합과정의 전 처리로써 유사도 맵 생성 알고리즘을 제안한다. 본 논문에서는 블록화한 히스토그램을 통하여 영상간의 관계를 판별하게 된다. 두 영상의 블록 히스토그램을 비교하여 영상 간의 유사성과 위치관계를 8 방향으로 판별하고 이를 이용하여 유사도 맵에 영상들을 정렬하게 된다. 유사도 맵의 생성으로 정합 알고리즘을 적용해야 하는 경우의 수가 줄어들어 복잡도는 낮아지게 되어 이후 정합과정에서 속도의 이득을 얻을 수 있다. 또한 정합 방법으로 변형이 적은 영상을 정합하는데 탁월한 성능과 속도를 보이는 히스토그램을 이용한 방법을 제안한다. 제안 알고리즘을 이용하여 실험한 결과 기존의 다중 영상 스티칭 알고리즘에 비하여 매우 빠른 속도를 확인 할 수 있고 결과 영상 또한 오류가 적은 것을 확인 할 수 있다.

1. 서론

최근 디스플레이 기술이 발전하면서 점차 커다란 화면을 사용하게 되었다. 그에 따라 디스플레이 하고자 하는 영상 또한 커지는 추세에 있다. 일반적인 카메라의 시야 각은 60 도에 불과하여 무대 혹은 경기장과 같이 광활한 영역을 실감 나게 전달하기 힘들다. 이를 해결하기 위해 다수의 카메라가 다 각도, 다 시점에서 찍은 영상을 합성하여 하나의 커다란 영상으로 만들려는 여러 가지 방법들이 제안되고 있다.

가장 일반적으로 많이 사용되는 방법이 Scale-invariant feature transform(SIFT)과 같이 불변 특징점을 검출하거나 형태를 바탕으로 영상들을 합성 하는 방법이다. 특징점을 영상합성에 이용하면 영상간의 합성을 하는 데는 매우 유리하다. 매우 정확한 합성 위치를 연산 할 수 있으며 영상의 변형에도 강하기 때문이다. 하지만 정확도에 비례하여 비교할 특징점과 처리할 정보 또한 많아지기 때문에 속도가 매우 느리고 많은 연산이 필요하다는 단점이 있다.

본 논문의 구성은 다음과 같다, 2 장에서는 히스토그램을 이용한 블록 기반 유사도 검사방법을 설명한다. 3 장에서는 영상 관계 맵 생성 및 히스토그램 영상 정합 알고리즘을 제안한다. 4 장에서는 실험 방법 및 결과를 보여주고 마지막으로 5 장에서는 본 논문에 대한 결론을 맺는다.

2. 히스토그램 매칭

2.1 YUV 영상으로의 변환

YUV 는 색상 표현 방식의 하나로써 휘도 신호인 Y 와 색차 신호인 U, V 로 색을 표현한다. RGB 값을 YUV 로 변환하는 식은 8bit 영상을 기준으로 다음과 같다.

$$\begin{aligned} Y &= 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \\ U &= (R - Y) \cdot 0.713 + 128 \\ V &= (B - Y) \cdot 0.564 + 128 \end{aligned}$$

RGB 방식에 비하여 YUV 를 사용하여 히스토그램 비교를 수행하게 되면 색상과 형태를 분리하여 문턱 값을 설정하고 비교할 수 있다는 장점이 있다.

2.2 히스토그램 블록을 이용한 유사도 검사방법

히스토그램은 데이터의 분포를 막대 그래프 형태로 표현한 것으로서, 영상을 히스토그램 분석하면 영상 내의 데이터 분포가 어떻게 이루어지는지 알 수 있다. 이처럼 히스토그램은 영상 분석을 위한 중요한 척도이다. 따라서 히스토그램이 유사한 두 영상은 유사한 영상으로 판단 할 수 있다.

히스토그램 비교 계산 시 아래의 상관계수 식을 사용한다.

$$d(H_1, H_2) = \sum_i \frac{\sum_i (H_1(i) - \bar{H}_1)(H_2(i) - \bar{H}_2)}{\sqrt{\sum_i (H_1(i) - \bar{H}_1)^2 \times \sum_i (H_2(i) - \bar{H}_2)^2}}$$

이때 $\bar{H}_k = \frac{1}{N} \sum_j H_k(j)$, 표본평균

히스토그램 비교 이후 비교한 결과값이 문턱 값을 넘는지 판단하여 유사성을 검사하게 되는데 그 정도를 형태의 정보를

가지는 Y 영역은 0.8, 색상의 정보를 가지는 U, V 는 0.5 로 하였다.

검사 순서는 다음과 같다. 먼저 두 영상을 입력 받는다. 입력 받은 영상을 전체를 우선 히스토그램 비교한다. 이 때 계산한 결과가 정해진 임계 값을 넘으면 두 영상은 유사도가 매우 높아 거의 같은 위치에 존재한다고 판단한다.

만약 그렇지 않다면 영상을 가로로 이분할, 세로로 이분할 하여 분할된 영상끼리 히스토그램 분석하고 비교한다. 이를 통해 두 영상의 위치관계에 관해 판단 가능하다.

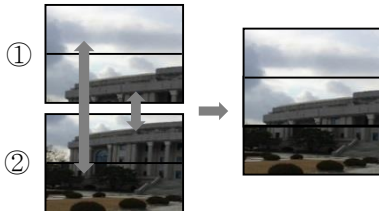


사진 2. 히스토그램 블록을 이용한 유사도 검사 예시 1

사진 2 과 같이 1 번 영상의 아랫부분과 2 번 영상의 윗부분을 히스토그램 비교하여 유사하다고 판정되면 2 번영상은 1 번영상의 1/2 만큼 겹치면서 아래쪽에 위치한다고 판단한다. 만약 반대로 1 번의 윗부분과 2 번의 아랫부분이 유사하다고 판정되면 1 번의 위쪽에 2 번영상이 위치한다고 판단하는 것이다.

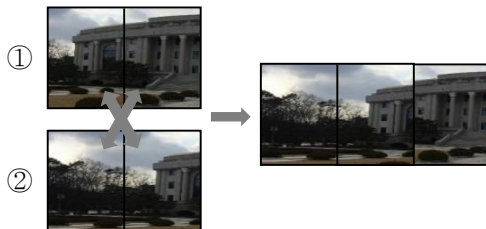


사진 3 히스토그램 블록을 이용한 유사도 검사 예시 2

또한 사진 3 같이 1 번 영상의 왼쪽 부분과 2 번 영상의 오른쪽 부분을 히스토그램 비교하여 유사도가 높다고 결과가 나오면 2 번 영상은 1 번 영상의 1/2 만큼 겹치면서 왼쪽에 위치한다고 판단할 수 있다.

만약 반대로 1 번의 오른쪽 부분과 2 번의 왼쪽 부분이 유사하다고 판정되면 1 번의 오른쪽에 2 번 영상이 위치한다고 판단한다.

이 과정을 통해서도 영상 간에 유사도가 있다고 판정되지 않는다면 영상을 4 분할 하여 비교한다. 나눈 영상간의 히스토그램 비교를 통해 두 원본 영상이 대각선 방향의 어느 위치에 관계되는지 알 수 있다.

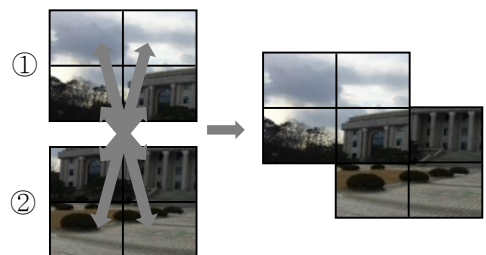


사진 4. 히스토그램 블록을 이용한 유사도 검사 예시 3

사진 4 를 보면 1 번 영상의 오른쪽 아랫부분과 2 번 영상의 왼쪽 윗부분이 히스토그램 비교 시 유사도가 있다고 판정이 됐다. 따라서 2 번 영상은 1 번 영상의 오른쪽 아래 대각선에 1/4 만큼 겹치면서 위치하고 있다고 있다고 판단한다.

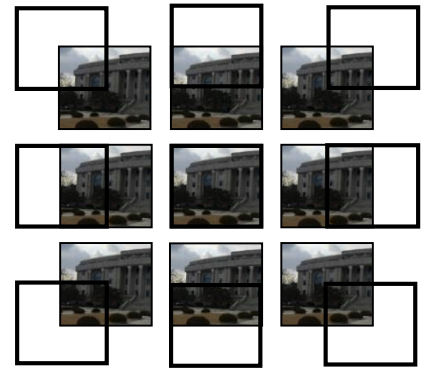
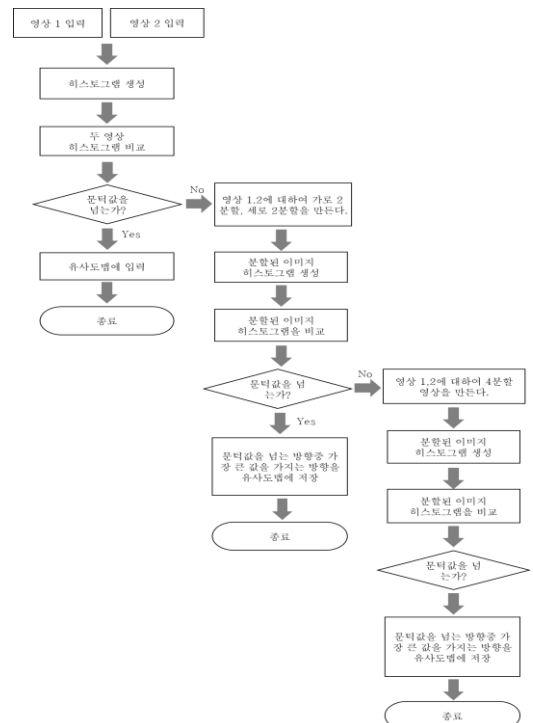


사진 5. 유사도 비교를 통해 확인 가능한 위치 관계

이렇게 두 개의 영상을 비교하면서 알 수 있는 위치관계는 사진 5 와 같이 한 영상을 기준으로 총 8 개로 할 수 있다. 전체 알고리즘 진행을 플로우 차트로 나타내면 다음 순서도 1 과 같다.



순서도 1. Histogram Block Similarity Algorithm Flow chart

3. 유사도 관계 맵 생성 및 영상 합성

본 논문에서 제안한 시스템은 다음과 같은 순서로 동작한다. 먼저 비교하고자 하는 영상을 입력 받는다. 이때 입력 받는 영상은 512x512 사이즈의 컬러 영상으로 한다. 이 영상들의 데이터 표현 방식을 YUV 로 바꾸고 히스토그램 분석하여 영상간의 유사도를 검사한다. 검사한 유사도를 바탕으로 유사도 맵을 생성한다. 생성된 유사도 맵에서 주변의 영상들과 히스토그램 블록을 단위로 하는 매칭을 수행한다.

매칭된 결과를 가지고 영상을 합성하여 결과 영상을 출력한다.

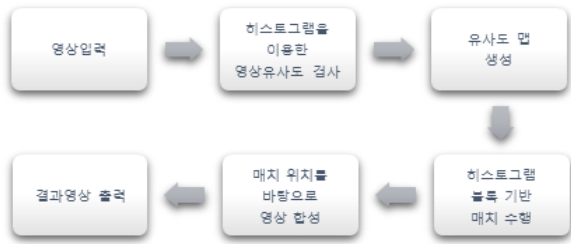


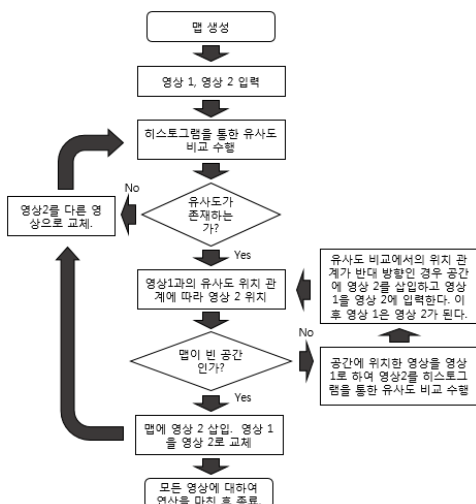
사진 1. System Block Diagram

3.1 이미지 유사도 관계 맵 생성

히스토그램을 이용한 유사도 검사 방법을 이용하게 되면 두 영상이 어느 특정 위치에 존재하는지 알 수 있게 된다. 이 결과값을 바탕으로 이미지 간의 유사도 관계를 나타내 주는 이미지 맵을 생성하게 된다. 이미지 맵은 유사도가 존재하는 영상들의 집합으로써 A 영상이 B 영상의 오른쪽을 촬영한 영상이라면 맵 상에서 A 영상을 B 영상의 오른쪽에 위치하도록 만들어 진다.

정합 과정의 전처리 과정으로써 이미지 맵을 생성하는 것에는 연산량을 줄일 수 있다는 장점이 있다. n 개의 영상을 입력하였을 때 이 영상을 합성하기 위한 검사 량은 $O(n^2)$ 으로 표현 할 수 있다. 하지만 먼저 유사도를 바탕으로 한 이미지 맵을 생성하게 되면 영상 합성을 위한 과정에 $O(n)$ 의 연산량만이 필요하게 된다.

이미지 관계 맵을 생성하기 위하여 먼저 영상들을 입력 받는다. 영상을 입력 받은 후 첫 번째 영상을 가지고 입력된 모든 영상들과 히스토그램 블록 기반 유사도 검사를 수행하게 된다. 수행 중 두 영상이 유사성을 가지고 있다면 그 상대적 위치를 받아 첫 번째 영상을 맵에 위치하고 이에 대한 상대적 위치에 유사성이 존재하는 영상을 위치 시키게 된다. 한번 유사한 영상을 찾은 영상에 대한 다른 영상과 검사를 하지 않는다. 첫 번째 영상과 모든 영상을 비교한 이후에는 첫 번째 영상과 유사성이 존재한다고 판단한 영상을 기준으로 유사성 영상을 찾지 못한 나머지 영상들과 유사성 검사를 반복해서 진행하게 된다.



순서도 2. 이미지 맵 생성 알고리즘 플로우 차트

이미지 맵 을 생성하면서 가장 중요한 것은 이미지 맵 위 하나의 구역에는 하나의 영상만을 두어야 한다는 것이다. 완전히 동일한 영상이 아니라면 그 영상은 상대적 위치가 생길 수 밖에 없기 때문이다. 만일 두 영상이 매우 비슷하다면 다른 영상 하나는 제거된다. 이를 위해서 정렬을 하게 되는데 과정은 순서도 2 와 같다.

만약 이미 이미지 맵에 영상이 있는 상태에서 다른 영상을 삽입하게 되면 입력된 영상과 입력을 할 영상을 비교하여 맵에 영상을 입력할 위치를 결정하게 된다.

3.2 히스토그램 영상 정합

유사도 맵을 생성한 이후 이를 바탕으로 영상 정합을 수행하게 된다. 정합을 할 때에는 유사도 맵에서 주변에 위치한 영상들을 끼리 수행한다. 히스토그램을 이용한 정합은 다음과 같은 과정으로 수행된다.

사진 6 은 1 번 영상과 2 번 영상을 정합하려는 예시이다. 2 번 영상을 기준으로 두고 1 번 영상을 붙이게 되는데 이 때 1 번 영상의 중앙 좌표를 2 번 영상에서 빨간 점으로 표시한 위치에 놓는다. 위치는 2 번 영상의 중앙점을 (x,y)라고 하고 영상 가로 길이의 1/2 를 size 라 할 때, (x-size, y-size), (x, y-size), (x+size, y-size), (x-size, y), (x, y), (x+size, y), (x-size, y+size), (x, y+size), (x+size, y+size)로 한다. 영상을 각 위치에 놓은 후 겹치는 부분에 대하여 히스토그램 비교를 수행한다. 가장 히스토그램 비교 값이 높은 위치를 매칭이 잘된 위치라 가정하고 그 위치를 (x,y)에 대입한다. 이후 size 를 2 로 나누어 위 과정을 size 가 1 이 될 때까지 반복한다. 최종적으로 결정된 위치가 영상을 정합하게 될 위치이다.

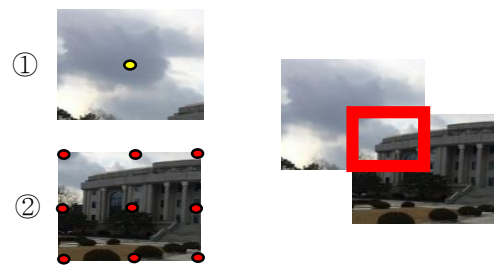


사진 6. 히스토그램 영상 정합 예시

4. 실험 방법 및 결과

알고리즘을 검증하는 실험을 수행하기 위하여 본 논문에서는 512x512 해상도의 영상들을 실험 영상으로 사용한다. 각 영상은 풍경 일부를 담고 있는 영상들이다. 실험 환경은 Microsoft 사의 Microsoft Visual Studio 2013 과 OpenCV2.4.10 라이브러리를 이용하여 구현하고, 3.4GHz 의 인텔 i7 프로세서를 사용하였다. 사용한 영상들은 다음 사진 7 과 같다.



사진 7. 실험 영상 세트

위 영상들을 입력으로 하여 이미지 유사도 관계 맵을 생성하였다. 그 결과는 사진 8 과 같다.



사진 8. 생성된 이미지 맵

위 영상 생성에 걸린 시간은 943ms 이다. 맵을 살펴보면 되면 각 영상이 유사한 관계성에 따라 정렬된 것을 확인 할 수 있다. 이를 바탕으로 히스토그램 영상 정합을 수행하였다. 그 결과는 다음 사진 9 와 같다.



사진 9. 정합 완료된 영상

결과 영상을 생성하는데 걸린 수행 시간은 3,437ms 이다. 결과 영상의 크기는 1721x974 의 크기이다. OpenCV 에서 자체적으로 제공하는 Sticher 함수가 영상 2 개만을 정합하는데 8700ms 의 시간이 필요한 것에 비하여 제안 알고리즘은 여러 영상에 대하여 정합하는데 매우 짧은 시간이 소요되었다.

5. 결론 및 향후 과제

본 논문에서 제안하는 알고리즘은 형태가 변하지 않는 다수의 영상간에 영상 정합을 빠르게 수행하기 위하여 히스토그램을 이용한 유사도 관계 맵을 이용하였다. 이 방법을 사용하게 되면 서로 다른 영상이 정합되는 경우도 막을 수 있으며 연산 속도 또한 증가시킬 수 있다는 장점이 있다. 또한, 만들어진 맵을 바탕으로 히스토그램을 기반으로 하는 영상 정합도 수행하였다. 이 방법은 왜곡이 적은 영상을 빠르게 정합하는 데 매우 효과적이었으며 전체 수행 시간은 기존의 영상 정합 알고리즘보다 매우 빨랐다.

본 논문에서 제안하는 알고리즘은 회전 및 변형에

약하다는 단점도 존재한다. 이러한 단점을 극복할 수 있는 방법을 개발하는 것을 향후 과제로 한다.

* 본 논문은 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [B0126-15-1013, 퍼즐형 Ultra-wide viewing 공간 미디어 생성 및 소비 기술 개발]

REFERENCES

- [1] Brown, Matthew, "Automatic Panoramic Image Stitching using Invariant Features", International Journal of Computer Vision, vol. 74, no. 1, Aug. 2007
- [2] Lowe, David G, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, Vol. 60, no. 2, Nov. 2004
- [3] OpenCV Reference (<http://docs.opencv.org>)