

## 정렬 오류 누적에 강인한 실시간 파노라마 합성 방법

김범수 조남익

서울대학교

bskim@ispl.snu.ac.kr nicho@snu.ac.kr

## Real-time panoramic stitching algorithm robust to alignment error accumulation

Kim, Beom Su Cho, Nam Ik

INMC, School of Electrical Engineering and Computer Science, Seoul National University

## 요약

모바일 기기에서 연속적으로 입력되는 영상을 파노라마 합성을 하여 사용자에게 실시간으로 결과를 보여주는 기존의 실시간 파노라마 기법은 트래킹을 기반으로 하고 이미 채워진 영역은 재투영 하지 않기 때문에, 정렬 오류가 누적되고 누적된 정렬 오류가 결과 영상에 그대로 반영되는 문제가 있다. 이를 해결하기 위하여 본 논문에서는 실시간으로 합성된 파노라마 결과에서 정렬 오류가 존재하는 부분과 장면에서 움직이는 물체가 투영된 부분을 판별하고 이 부분만을 다시 투영하는 방법을 제안한다. 정렬 오류가 발생한 부분을 판별하기 위하여, 시간차가 존재하는 여러 장의 영상을 정렬한 후 같은 위치의 픽셀에 속하는 컬러 값을 큐에 저장한다. 정렬 오류가 발생하거나, 움직이는 물체가 존재하는 경우 큐에 저장된 컬러 값의 차이가 커지게 되고 이러한 부분은 다시 투영하여 파노라마 결과 영상에서 오류를 보정하게 된다. 또한 정렬 오류를 최대한 보정하기 위하여 두 단계로 이루어진 블렌딩 방법을 제안한다. 제안하는 방법은 실시간으로 동작하면서 정렬 오류가 발생한 부분을 효과적으로 판별하여 기존의 방법에 비하여 정렬 오류가 줄어들음을 확인하였다.

## 1. 서론

파노라마 영상은 사진 기술이 발명된 순간부터 시작되었을 정도로 오랜 역사를 가지고 있다.[1] 파노라마 영상의 시초는 기존의 필름보다 가로 비율이 더 넓은 필름을 사용하는 방식이었으며, 이후 디지털 영상의 획득 및 처리 시스템이 발전하면서 여러 장의 영상을 이용하여 더 넓은 Field Of View(FOV)를 가지는 한 장의 큰 영상으로 합성하는 단계로 발전하였다.[1, 10] 초기의 파노라마 합성은 카메라 움직임의 방향을 수평 또는 수직의 한 방향으로 고정시켜 변환 행렬(transformation matrix)의 계산이 높은 정확도를 가지면서도 간단한 방식으로 이루어져 있다.[2] 하지만 카메라의 움직임이 제한되어 있기 때문에 사용자가 손쉽게 파노라마 영상을 얻을 수 없는 단점이 존재한다. 이후 affine 변환과 조명 변화에 강인한 descriptor 기반의 SIFT나 SURF 등의 특징점 추출 기술이 개발되었고 이를 이용하여 다양한 카메라 움직임을 처리할 수 있는 파노라마 합성 기술이 개발되었다.[6, 7, 11, 12] 이 같은 파노라마 합성 방법은 일반적으로 모든 영상에서 특징점 추출, 영상 간에 추출된 특징점의 매칭, outlier 제거, 변환 행렬 계산, 투영의 다섯 단계로 이루어져 있다.[10] 하지만 이 같은 방식의 파노라마 합성은 복잡한 계산을 요구하기 때문에 휴대기기에 바로 적용하기 어려운 단점이 있다. 현재는 스마트폰 같은 휴대기기의 발전으로 휴대기기에서 손쉽게 여러 장의 영상을 바로 합성하여 파노라마 영상을 얻을 수 있으며, 여러 장의 영상 뿐 만이 아니라 입력되는 동영상의 매 프레임을 실시간으로 합성하여 그 결과를 보여주는 시스템도 개발되었다.[3, 4, 5, 8, 9] 계산 성능이 일반적인 PC에 비하여 현저하게 떨

어지는 모바일 기기에서 초당 30프레임 이상의 동영상을 실시간으로 파노라마 합성을 하기 위해서는 기존의 파노라마 합성과는 다른 방식의 방법을 이용하게 된다. 일반적인 파노라마 영상 합성에서 가장 계산 시간이 오래 걸리는 단계는 특징점 추출과 매칭 단계이다. 이에 소요되는 계산 시간을 줄이기 위하여 복잡한 특징점 추출 알고리즘 대신 영상의 엣지 정보를 이용하여 인접한 프레임 사이에 평행이동 성분을 계산하고, 그레디언트 정보를 활용한 특징점을 추출하여 변환 행렬을 계산하는 방법이 제안되었다.[4] 이 방법은 모바일 기기에서 실시간 파노라마 합성을 가능하게 하는 최초의 연구이지만, 카메라의 움직임이 커지게 되면 높은 확률로 오류가 발생하게 되는 단점이 존재한다. 이후 모바일 기기에서 동작하는 실시간 파노라마 합성의 품질을 개선하기 위하여 트래킹을 사용한 새로운 방법이 제안되었다.[9] 이 방법은 특징점 추출에 소요되는 계산 시간을 줄이기 위하여 동영상의 매 프레임에서 특징점을 추출하는 대신 다른 영상과 겹치는 부분이 적은 소수의 영상에서만 특징점을 추출하고 나머지 영상에서는 기존에 추출된 특징점을 트래킹하게 된다. 또한 계층적인 3단계의 해상도에서 트래킹을 수행하며, 낮은 해상도에서 높은 해상도로 순차적으로 변환 행렬을 갱신해가는 전략을 사용한다. 이러한 계층적인 방식은 트래킹을 수행할 때 검색 범위를 줄일 수 있게 되어 계산 시간을 단축하는 효과를 가진다. 이때 갱신하고자 하는 변환 행렬은 원기둥 투영(cylindrical projection)의 파라미터로 사용되며, 원기둥 투영은 비선형 모델이기 때문에 비선형 최적화 기법인 iterative Gauss-newton 방법을 이용하여 변환 행렬을 계산하게 된다. 그리고 모든 프레임을 투영하게 되면 상당한 시간을 소모하기 때문에 모바일 기기에서 실시간 합성이 불가

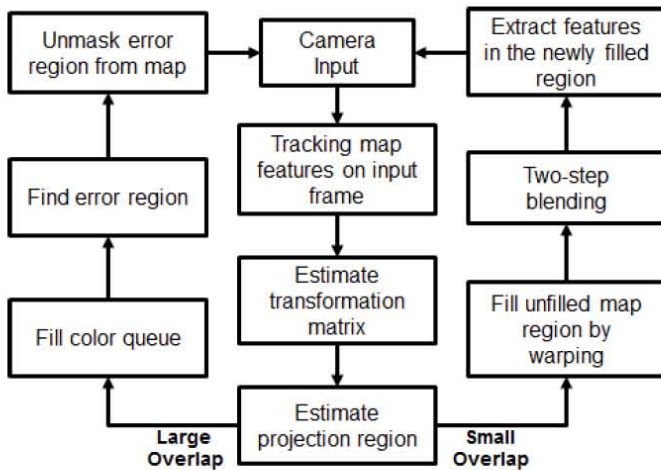


그림 1. 제안하는 시스템의 구성도.

능하다. 따라서 [9]의 방법은 기존의 채워진 부분은 제외하고 빈 부분만을 투영하여 채우게 된다.

하지만 이 방법은 잘못 정렬된 영상이 투영될 경우 후처리를 통한 수정이 불가능하기 때문에 제대로 된 파노라마 합성을 위해서는 여러 번의 시도가 필요하게 된다. 즉, 정렬 오류가 없는 파노라마 합성 결과를 얻기 위하여 여러 번 같은 과정을 반복하여야 한다. 또한 영상 취득 과정에서 장면 속에 움직이는 물체가 존재하게 되면 파노라마 합성 과정이 실패하는 단점이 존재한다. 이를 해결하는 가장 간단한 방법은 잘못 정렬된 영상이 추후 수정될 수 있도록 입력 영상의 모든 픽셀을 계속해서 투영하는 것이다. 하지만 모든 픽셀을 투영하게 되면, 모바일 기기에서 실시간으로 동작하기 힘들기 때문에 이 방법은 현실적으로 불가능하다. 또한 이러한 접근법은 움직이는 물체를 구분하기 힘들기 때문에 배경에 움직이는 물체가 존재하는 경우 이를 해결할 수 없다.

본 논문에서는 이를 해결하기 위하여 실시간으로 합성된 파노라마 결과에서 정렬 오류가 존재하는 부분과 장면에서 움직이는 물체가 투영된 부분을 판별하고 이 부분만을 다시 투영하는 방법을 제안한다. 이를 위하여, 정렬 오류가 발생하지 않거나 움직이는 물체가 없는 배경 영역에서는 여러 영상에서 같은 위치에 속한 픽셀의 컬러 값이 변하지 않고 일정하다는 가정을 이용한다. 정렬 오류가 발생한 부분을 판별하기 위하여, 시간차가 존재하는 여러 장의 영상을 정렬한 후 같은 위치의 픽셀에 속하는 컬러 값을 큐에 저장한다. 정렬 오류가 발생하거나, 움직이는 물체가 존재하는 경우 큐에 저장된 컬러 값의 차이가 커지게 되고 이러한 부분은 다시 투영하여 파노라마 결과 영상에서 오류를 보정하게 된다. 또한 정렬 오류를 최대한 보정하기 위하여 두 단계로 이루어진 블렌딩 방법을 제안한다. 제안하는 방법은 실시간으로 동작하면서 정렬 오류가 발생한 부분을 효과적으로 판별하여 기존의 방법에 비하여 정렬 오류가 줄어들음을 확인하였다.

## 2. 제안하는 방법

### 가. 제안하는 시스템의 개요

제안하는 시스템은 연속적으로 입력되는 카메라의 영상을 맵(map)이라 불리는 파노라마 영역으로 매 순간 투영하여, 실시간으로

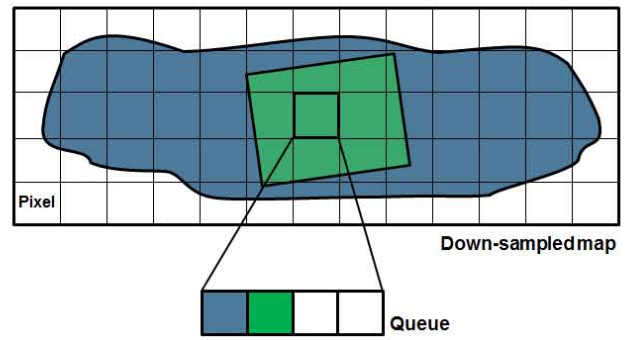


그림 2. 정렬 오류를 검출하기 위한 컬러 큐.

맵이 갱신되며 파노라마 영상 합성이 이루어지는 모습을 사용자에게 연속적으로 보여주는 방식으로 동작한다.

제안하는 방법은 크게 특징점 추출, 추출된 특징점의 트래킹, 변환 행렬 추정, 투영 영역 추정, 빈 영역으로 입력 영상을 투영하는 단계로 구성되어 있으며, 기본적인 시스템의 구성은 기존의 방법 및 선행 연구의 방식을 채용하였다. [9, 13] 여기에 정렬 오류를 판별하고 이를 보정하기 위하여 기존에 계산된 파노라마 영상과 현재 영상과의 오버랩 영역을 계산하고 오버랩 영역의 크기에 따라 투영하거나 또는 정렬 오류가 발생하는 부분을 찾는 단계가 새롭게 추가된다. 오버랩 영역이 클 경우, 맵으로 투영을 할 부분이 작기 때문에 맵을 채우는 대신 정렬 오류가 발생하는 부분을 찾게 된다. 반대로 오버랩 영역이 작은 경우 입력된 영상을 투영하여 맵을 채우게 된다. 그림 1 은 제안하는 시스템의 구성도를 보여준다.

### 나. 맵에서 발생하는 정렬 오류의 판별

이미 채워진 맵 영상에서 정렬이 잘못된 곳이나 움직이는 물체가 투영된 곳을 찾아내기 위하여 그림 2와 같이 맵의 각 픽셀마다 큐(queue)를 설정한다. 이 큐에는 맵으로 투영되는 입력 영상의 컬러 정보를 순차적으로 저장하게 된다. 큐가 가득 차게 되면 컬러 값의 최대 거리를 계산하게 된다. 컬러는 R, G, B 삼차원 벡터로 정의되며, 컬러 벡터 사이의 거리는 다음과 같은 Mahalanobis distance로 정의된다.

$$d_{ij}^2(I_i(x,y), I_j(x,y)) = (I_i(x,y) - I_j(x,y))^T C^{-1} (I_i(x,y) - I_j(x,y))$$

여기서, C는 R, G, B 컬러 사이의 공분산 행렬이며 입력 영상을 토대로 계산된다. (x,y)는 맵의 픽셀 좌표이며, I<sub>i</sub>는 큐에서 i번째 컬러 벡터를 나타낸다. 이때 픽셀 (x,y)의 최대 거리는 다음과 같이 구하게 된다.

$$d_m^2(x,y) = \max(d_{12}^2, d_{13}^2, d_{14}^2, \dots)$$

이렇게 구한 컬러의 최대 거리 값을 이용하여 정렬 오류 또는 움직이는 물체가 맵으로 투영되었는지를 다음과 같이 판단하게 된다.

$$p(x,y) = \begin{cases} correct & d_m^2(x,y) \leq \tau \\ error & else \end{cases}$$

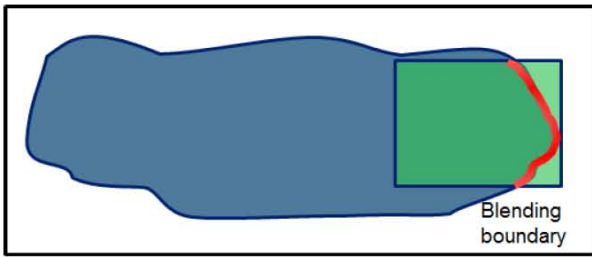


그림 3. 선형 블렌딩.

맵의 큐를 채우기 위해서는 각 입력 영상의 모든 픽셀을 맵으로 투영하여야 한다. 하지만 이 과정은 모바일 기기에서 실시간으로 처리하기에는 한계가 있다. 따라서 맵의 원본 해상도에서 큐를 채우는 과정을 수행하는 대신에, 4배로 down-sampling 된 해상도에서 큐를 채우게 된다. 이렇게 down-sampling 된 해상도에서 정렬 오류를 검출하게 되고 정렬 오류가 검출되는 경우 원본 해상도로 up-sampling을 수행하여 맵에 반영하게 된다. 맵에서 정렬 오류가 검출된 픽셀은 추후 입력되는 영상을 이용하여 다시 투영을 수행하여 채우게 된다.

**다. 정렬 오류를 보정하기 위한 블렌딩**

정렬 오류를 판별하기 위하여 입력된 영상을 투영하여 큐에 컬러 정보를 채우는 과정은 입력 영상을 투영하여 맵을 채우는 과정과 비슷한 시간이 걸리기 때문에, 두 과정을 동시에 실시간으로 수행하는 것은 모바일 기기에서 한계가 있다. 따라서 제안하는 시스템은 그림 1과 같이 입력되는 모든 프레임을 맵으로 투영하지 않고 맵과 입력영상과의 오버랩 영역이 작은 경우에만 투영을 수행하고 오버랩 영역이 큰 경우 투영을 수행하지 않고 큐에 컬러 정보를 채우는 과정을 수행한다. 이 경우, 맵을 채우는 입력 영상 사이에 약간의 시간차가 생기게 되고 맵과 입력 영상 사이에 정렬이 완벽하지 않은 경우 그림 5 (위) 와 같이 서로 다른 입력 영상 사이로 경계선이 생기게 된다. 이를 보완하기 위하여 맵으로 투영을 수행할 시 그림 3과 같이 투영의 경계선을 주위로 선형 블렌딩을 수행한다. 또한 정렬이 잘못된 것으로 판별된 픽셀만을 추후 다시 채우는 경우 채울 픽셀과 인접 픽셀의 소스 영상 간에 시간차가 상당히 크게 된다. 현재 시스템에서는 트래킹에 기반하여 입력 영상을 정렬하기 때문에 시간이 지날수록 오차가 누적된다. 따라서 인접한 픽셀 사이에 소스 영상 간의 시간차가 크게 되면 누적된 오차로 인하여 경계선이 생길 가능성이 높아진다. 이를 보정하기 위하여 맵을 셀 (cell)이라고 불리는 작은 영역으로 나눈 후, 정렬 오류가 발생한 픽셀을 포함하는 셀을 투영의 기본 단위로 한다. 예를 들어, 하나의 셀을 모두 채울 수 있는 소스 영상이 입력되면, 셀의 정렬 오류가 발생한 영역 뿐만 아니라 기존에 이미 다른 영상으로 채워진 부분도 함께 하나의 소스 영상을 투영하여 채우게 된다. 그리고 인접한 셀과의 정렬 오류를 보정하기 위하여, 그림 4와 같이 셀의 네 변을 주위로 선형 블렌딩을 수행하게 된다. 이렇게 두 단계로 이루어진 블렌딩은 정렬 오류가 누적되는 현상을 효과적으로 해결할 수 있다.

**4. 실험 결과 및 분석**

실험은 320x240 크기의 입력 영상을 이용하여 수행하였고, 맵의

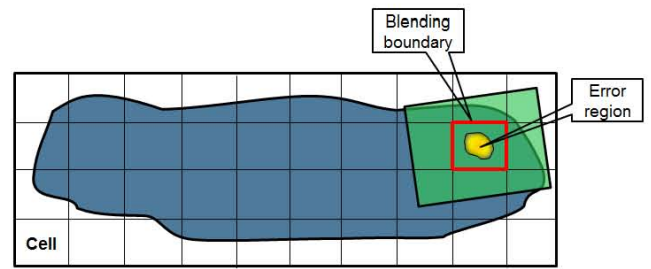


그림 4. 정렬 오류 영역 블렌딩.

크기는 2048x512 이며 셀의 크기는 64x64 이다. 그림 5 (위)는 기존의 실시간 파노라마 합성 결과의 결과 그림이다. 빨간 원 안 쪽에서 보듯이 인접 픽셀에서 소스 영상이 다를 경우 경계선이 나타나는 문제가 나타난다. 또한 형광등이 있는 부분에 정렬 오류가 발생한 모습을 볼 수 있다. 그림 5 (아래)는 같은 입력에 대하여 제안하는 방법을 적용한 실험 결과이다. 기존에 방법에서 발생하던 정렬 오류가 효과적으로 제거되었음을 확인할 수 있다. 또한 인접 픽셀에서 소스 영상이 다를 경우 발생하는 경계선도 제안하는 방법에서는 나타나지 않음을 볼 수 있다. 2.67Ghz 인텔 i5 CPU를 사용한 PC에서 한 프레임을 처리하는데 걸리는 평균적으로 2~3ms 로 측정되었다. 기존의 방법에 비하여 정렬 오류가 검출하는 부분이 추가되었지만 입력 영상과 맵 사이에 오버랩 영역의 크기에 따라 수행 여부를 결정하기 때문에 수행시간에서 기존의 방법과 크게 차이가 없음을 확인할 수 있다. [9] 따라서 제안하는 방법은 실시간 파노라마 합성에서 실시간 조건을 만족시키면서 합성 결과를 효과적으로 개선시킨다고 볼 수 있다.

**5. 결론 및 향후 과제**

본 논문에서는 기존의 실시간 파노라마 합성 기법이 제대로 처리하지 못하던 맵에서 발생하는 정렬 오류 누적 현상과 장면에서 움직이는 물체가 존재하는 경우 맵에서 발생하는 여러 오류를 해결하고자, 이미 발생한 정렬 오류를 판별하고 이 영역을 블렌딩을 통하여 채투영하여 정렬 오류를 줄이는 방법을 제안하였다. 실시간으로 정렬 오류를 판별하기 위하여, 입력 영상과 맵 사이에 오버랩 영역이 큰 경우에는 투영을 실시하고, 오버랩 영역이 작은 경우 정렬 오류 영역을 추출하게 된다. 정렬 오류를 판별하기 위하여, 시간차가 존재하는 여러 장의 영상을 정렬한 후 같은 픽셀에 속하는 컬러 값을 큐에 저장한다. 정렬 오류가 발생하거나, 움직이는 물체가 존재하는 경우 큐에 저장된 컬러 값의 차이가 커지게 되고 이러한 부분은 다시 투영하여 파노라마 결과 영상에서 오류를 보정하게 된다. 또한 정렬 오류를 최대한 보정하기 위하여 두 단계로 이루어진 블렌딩 방법을 제안하였다. 실험 결과, 제안하는 방법은 기존의 실시간 파노라마에서 발생하는 경계선 현상이나 정렬 오류가 발생하는 영역을 효과적으로 줄이는 것을 확인하였다. 또한 수행시간 측면에서도, 기존의 방법에 비해 큰 차이 없이 실시간으로 동작함을 확인하였다. 현재 제안하는 방법은 정렬 오류가 없는 배경 영상만을 실시간으로 파노라마 합성하고 있다. 추후, 이 방법을 확대 적용하여 실시간으로 장면의 배경을 파노라마 합성하는 동시에 움직이는 물체를 추출하고 이를 트래킹하여 진경 분리를 수행할 수 있는 알고리즘을 개발하고자 한다.



그림 5. 실험 결과 (위) 기존의 방법[9] (아래) 제안하는 방법.

### 감사의 글

이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. 2012-0000913).

### 참 고 문 헌

- [1] R. Szeliski, "Image alignment and stitching: a tutorial," *Found. Trends. Comput. Graph. Vis.*, vol. 2, no. 1, pp. 1-104, 2006. [Online]. Available: <http://dx.doi.org/10.1561/0600000009>
- [2] S. J. Ha, H. I. Koo, S. H. Lee, N. I. Cho, and S. K. Kim, "Panorama mosaic optimization for mobile camera system," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 4, pp. 1217-1225, 2007.
- [3] S. J. Ha, S. H. Lee, N. I. Cho, S. K. Kim, and B. J. Son, "Embedded panoramic mosaic system using auto-shot interface," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 1, pp. 16-24, 2008.
- [4] A. Adams, N. Gelfand and K. Pulli, "Viewfinder alignment," *Computer Graphics Forum (Proceedings of Eurographics)*, pp. 597-606, 2008
- [5] P. Baudisch, D. Tan, D. Steedly, E. Rudolph, M. Uyttendaele, C. Pal, and R. Szeliski. "Panoramic viewfinder: providing a real-time preview to help users avoid flaws in panoramic pictures," *In Proceedings of the 17th Australia conference on Computer-Human Interaction (OZCHI '05)*, Australia, 2005.
- [6] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "SURF: Speeded-up robust features", *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346-359, 2008.
- [7] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *International Journal of Computer Vision*, vol. 74, no. 1, 2007.
- [8] D. Steedly, C. Pal, and R. Szeliski, "Efficiently Registering Video into Panoramic Mosaics," *In Proceedings of the 10th International Conference on Computer Vision (ICCV '05)*, 2005.
- [9] D. Wagner, A. Mulloni, T. Langlotz, and D. Schmalstieg, "Real-time panoramic mapping and tracking on mobile phones," *Virtual Reality Conference (VR '05)*, 2010.
- [10] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.
- [11] M. Brown and D. Lowe, "Recognising panoramas," *in Proceedings of Ninth IEEE International Conference on Computer Vision (ICCV '03)*, 2003.
- [12] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *International Journal of Computer Vision (IJCV '07)*, vol. 74, no. 1, pp. 59-73, 2007.
- [13] B. S. Kim, S. H. Lee and N. I. Cho, "Real-time Panorama Canvas of Natural Image," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 4, pp. 1961-1968, 2011.