

다수의 이미지 정합을 위한 동적 프로그래밍 스티칭 적용

*이연경 *심규동 *박종일

*한양대학교

yklee@mr.hanyang.ac.kr, kdsim@mr.hanyang.ac.kr, jipark@hanyang.ac.kr

Multiple Seamless Image stitching using Adaptive Dynamic Programming Method

Younkyoung Lee, Kyudong Sim, Jong-Il Park
Hanyang University

요 약

본 논문에서는 동적 프로그래밍 스티칭을 이용하여 다수의 이미지를 경계가 보이지 않게 정합하여 고해상도의 이미지를 얻는 방법을 소개한다. 제안하는 방법에서는 수직, 수평방향으로 일정한 간격으로 촬영한 다수의 지역 이미지와 전체를 촬영한 전역 이미지를 사용해서 각각의 지역 이미지와 전역 이미지의 특징점을 추출하고 이를 매칭하여 호모그래피를 계산한다. 이를 이용하여 정합할 두 지역 이미지간의 호모그래피를 구하고 좌표를 변환한 후 겹치는 영역에 동적 프로그래밍 스티칭 방법을 적용하여 두 이미지를 정합한다. 동적 프로그래밍 스티칭 방법이란 두 이미지를 정합할 때 겹치는 영역의 차이를 계산하고 차이가 가장 적은 픽셀을 경계로 하는 방법이다. 다수의 이미지를 수직방향으로 정합하고 정합된 이미지들을 수평방향으로 정합하여 하나의 고해상도 이미지를 만들 수 있다. 제안하는 스티칭 기법을 적용함으로써 이미지간의 경계가 드러나지 않을 뿐만 아니라 각 픽셀의 세밀한 정보도 유지한 고해상도의 이미지를 획득할 수 있음을 보였다.

1. 서론

영상처리에서의 이미지 스티칭은 두 개 이상의 이미지를 정합하여 하나의 이미지를 만드는 것을 의미한다. 이미지 스티칭에서는 정합할 두 이미지 사이의 관계를 나타내는 호모그래피를 계산하여 이미지의 좌표를 변환시키는 과정과 겹치는 영역에서의 경계선을 없애기 위해 이미지를 블렌딩 하는 과정이 포함된다.

다수의 이미지를 정합하기 위해서는 겹치는 영역에서의 특징점을 매칭하여 하나의 2D 호모그래피를 구하여 이미지를 변환시키는 방법이나, 지역 호모그래피와 전역 호모그래피를 구하고 이를 조정하여 이미지를 자연스럽게 변환시키는 방법 등이 제안되었다[2][3]. 중첩된 이미지의 경계를 보이지 않게 하면서 자연스럽게 처리하는 블렌딩 과정에는 알파 블렌딩, 멀티 밴드 블렌딩[5] 등이 제안되었으나 이미지의 선명도가 낮아지는 문제점이 있다. 이를 해결하기 위하여 경계가 보이지 않으면서 두 이미지 사이의 경계를 정하는 방법이 제안되고 있다. [1]에서는 이미지의 구조와 픽셀 사이의 오차를 동시에 고려하여 두 이미지의 차이가 가장 최소가 되는 픽셀을 경계로 하는 방법을 제안했다.

본 논문의 구성은 다음과 같다. 2 장에서는 격자 형태의 이미지를 동적 프로그래밍을 이용해 스티칭 하는 방법에 대해 소개한다. 3 장에서는 제안된 방법을 통해 다수의 이미지를 스티칭 한 결과를 확인하고 4 장에서는 본 논문에 대한 결론을 맺는다.

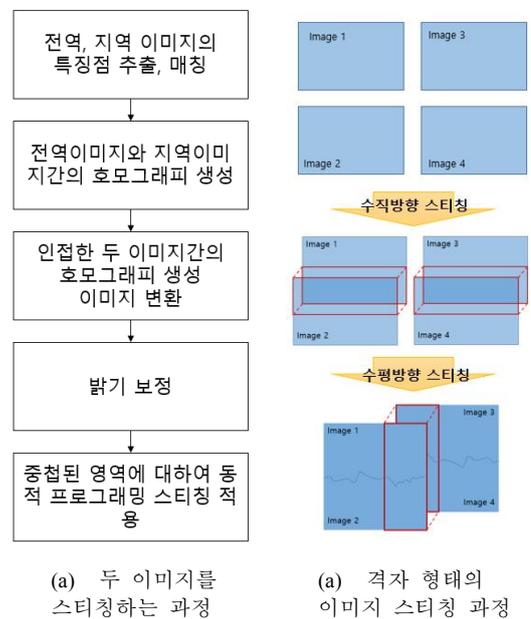


그림 1. 다수의 이미지 정합 과정

2. 다수의 이미지 정합

본 논문에서 제안하는 스티칭 방법은 동적 프로그래밍을 사용한 스티칭과 다수의 이미지를 스티칭하는 두 단계로 나뉜다.

1 교신 저자

수 있고 각각 그림 1 의 (a)와 (b)를 통해 확인할 수 있다. 그림 1(a)는 두 이미지 사이의 기하관계를 나타내는 호모그래피를 구하고 이미지를 정합하여 동적 프로그래밍 스티칭을 적용하는 과정이다. 그림 1(b)의 다수의 이미지를 스티칭하기 위해 스티칭 된 이미지에 스티칭 알고리즘을 다시 적용하는 과정이다. 또한 하나의 고해상도 이미지를 얻기 위해 수직, 수평방향으로 일정한 간격으로 촬영한 지역 이미지와 전체를 촬영한 전역 이미지를 사용한다.

2.1 두 장의 이미지 스티칭

두 이미지를 스티칭하기 위해서는 두 이미지의 기하관계를 나타내는 호모그래피 행렬을 구하여야 한다. 나란히 있는 두 이미지 A, B 사이의 호모그래피를 구하기 위해 전역 이미지 I를 사용함으로써 다수의 이미지를 스티칭한 결과 이미지가 전체적으로 왜곡되는 현상을 방지한다.

인접한 두 이미지 A, B 에서 I 로의 관계를 나타내는 호모그래피 H_A, H_B 를 구하기 위해서는 먼저 전체 이미지와 A, B 이미지 각각에서 특징점을 검출해야 한다. 검출된 특징점들을 매칭하고 RANSAC(Random Sample Consensus)기법을 이용하면 H_A, H_B 를 구할 수 있다. B 에서 A 로의 관계를 나타내는 호모그래피 $H_{B \rightarrow A}$ 는 H_A, H_B 를 이용해 아래의 식과 같이 구할 수 있다.

$$H_{B \rightarrow A} = H_A^{-1} * H_B \quad (1)$$

또한 두 이미지 사이의 조명에 의한 밝기 차이를 줄이기 위한 색 보정과정을 거친다. 스티칭 된 이미지의 밝기를 균일하게 하기 위해 전역 이미지 I 를 사용한다. 이미지 A 와 전체 이미지 I 에서 A 와 매칭되는 영역 I(A)의 밝기의 평균 $Y_A, Y_{I(A)}$ 를 구하고 (2)를 통해 A 이미지의 각 픽셀의 밝기를 전체 이미지의 밝기와 가깝게 만들어준다. 이를 통해 두 이미지 사이의 밝기 차이를 줄인다.

$$Y'_A(p, q) = Y_A(p, q) * \frac{Y_{I(A)}}{Y_A} \quad (2)$$

A 와 B 사이의 호모그래피를 구하였다면 두 이미지 사이의 중첩되는 영역에 대하여 동적 프로그래밍 스티칭을 기법을 적용한다. 중첩되는 영역의 각 픽셀 오차를 $D(p, q)$ 에 저장하고, 가장 적은 오차를 가지고 있는 픽셀을 스티칭 경계로 정한다. 이 때 픽셀 오차를 저장하는 $D(p, q)$ 는 (3)으로 정의한다.

$$D(p, q) = |I^A(p, q) - I^B(p, q) + offset| \quad (3)$$

픽셀 오차를 최소로 하는 픽셀을 찾기 위하여 이미지 구조를 반영하는 가중치 함수를 사용한 에너지 함수를 사용한다. 가중치 함수와 에너지 함수는 (4)와 같이 정의한다.

$$E_{total} = \alpha w(p, q)D(p, q) + (1 - \alpha)\alpha w(p, q)$$

$$w(p, q) = \exp\left\{-\frac{I_{min}(p, q)}{\sigma}\right\} \quad (4)$$

$$I_{min}(p, q) = \min\{I^A(p, q), (p, q)\}$$

위의 제안된 가중치 함수를 통해 에너지 함수는 픽셀 오차가 가장 적으면서 밝기가 가장 큰 값일 때 최소가 되고, 최소가 되는 픽셀을 두 이미지 사이의 경계로 정한다.

2.2 다수의 이미지 스티칭

다수의 이미지를 스티칭하기 위해 2.1 장에서 소개된 스티칭 방법을 반복적으로 수행한다. 각각의 지역 이미지들은 수직, 수평 방향으로 일정한 간격으로 촬영하였으므로 각 이미지를 좌측 상단부터 (0, 0), (0, 1), (0, 2), ..., (1, 0), (1, 1), (1, 2), ...와 같이 나타낼 수 있다. 위의 이미지들을 먼저 수직방향으로 스티칭하여 상하로 긴 이미지를 취득한다. 이 이미지는 변환하는 과정에서 반듯한 직사각형 형태가 아니므로 다음 스티칭을 위해 직사각형 형태로 잘라준다. 이 직사각형 형태의 상하로 긴 이미지를 수평방향으로 스티칭하여 다수의 이미지가 정합된 하나의 고해상도 이미지를 획득할 수 있다.

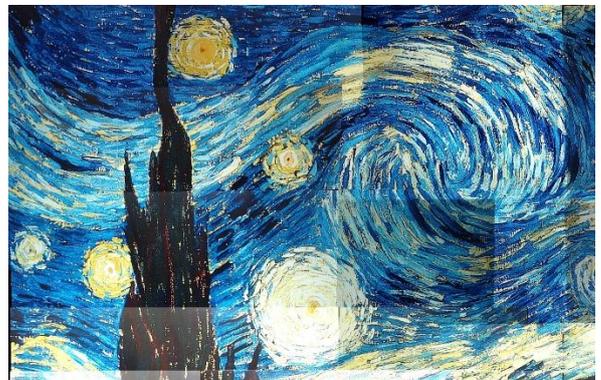
3. 실험 결과

본 논문에서 제안된 방법을 사용하여 그림 2(b)의 9 장의 이미지를 하나의 고해상도 이미지로 정합하였다. 실험에 사용된 이미지는 로지텍 HD Pro Webcam C920 을 촬영하였고 인접한 이미지간의 중첩되는 비율이 0.5가 되도록 하였다. 각 이미지의 크기는 4800*2700 이다. 이미지 사이의 호모그래피를 구하기 위해서 BRISK(Binary Robust Invariant Scalable Keypoint) [6] 기법을 사용하여 특징점을 추출, 기술하였다.



(a) 전역이미지 (b) 9 장의 지역 이미지
그림 2. 실험에 사용된 전역 이미지와 지역 이미지

이미지를 수직, 수평방향으로 스티칭 하면서 8238 * 4860 크기의 한장의 이미지를 만들었을 뿐만 아니라 블렌딩을 하지 않았기 때문에 각 픽셀에 들어있는 자세한 정보들을 그대로 사용해서 이미지를 합칠 수 있었다. 또한 그림 3 을 통해 동적 프로그래밍 스티칭과 색 보정으로 더 자연스러운 결과물을 얻을 수 있음을 알 수 있었다.



(a) 동적 프로그래밍 스티칭을 적용하지 않은 결과



(b) 동적 프로그래밍 스티칭을 적용했지만 색 보정을 하지 않은 결과



(c) 제안된 스티칭 방법에 따른 결과

그림 3. 각 방법에 따른 스티칭 결과 이미지

[3] C. Lin, S. U. Pankanti, “Adaptive As-Natural-As-Possible Image Stitching,” IEEE, 2015

[4] D. Gebreyohannes, J. Choi, “Composition of Foreground and Background Images using Optical Flow and Weighted Border Blending,” 한국컴퓨터그래픽스학회, 2014

[5] Cedric Allene, Jean-Philippe Pons, Renaud Keriven, “Seamless image-based texture atlases using multi-band blending,” ICPR, 2008

[6] S. Leutenegger, M. Chli, R. Y. Siegwart, “BRISK: Binary Robust invariant scalable keypoints,” ICCV, 2011

4. 결론

본 논문에서는 동적 이미지 스티칭 기법을 사용하여 다수의 이미지를 정합하는 방법을 소개하였다. 두 장의 이미지를 정합하는데 동적 프로그래밍 스티칭 기법을 사용하여 오차가 적은 최적의 경계선을 찾을 수 있었다. 또한, 전역 이미지를 사용해 왜곡이 없고 밝기가 균일한 고해상도의 이미지를 획득하였다. 3 장의 실험결과로 이미지 사이의 경계가 드러나지 않으면서 자세한 정보들이 유지됨을 확인할 수 있었으며 이를 통해 왜곡이 없는 고해상도 이미지를 획득하는데 도움이 될 것으로 기대된다.

감사의 글

본 연구는 문화체육관광부 및 한국콘텐츠진흥원의 2017년도 문화기술 연구개발 지원사업으로 수행되었음

참고문헌

[1] J. Park, K. Sim, S. Lee, J. Park, “Adaptive Dynamic Programming Method for Seamless Image Stitching,” ICEIC, 2017

[2] R. Szeliski, “Image alignment and stitching,” Computer Graphics and Vision, 2006