

# 다양한 촬영기기의 특성에 독립적인 영상 스티칭 기법

유신, 이종원, 김한수

서원대학교

[ryoushin@naver.com](mailto:ryoushin@naver.com), [jwlee56760@naver.com](mailto:jwlee56760@naver.com), [kutestar@seowon.ac.kr](mailto:kutestar@seowon.ac.kr)

## Image stitching technique independent of the characteristics of various filming devices

Shin Ryou, Jongwon Lee, Hansoo Kim  
Seowon University

### 요 약

스티칭 영상을 구성하는 과정에서 합성되는 영상들의 특징(해상도, framerate 등)을 빠르고 효율적으로 조정하는 것은 스티칭 알고리즘을 적용하는 중요한 요소이다. 현재 주로 사용되는 영상들은 서로 형식이 달라 기존의 스티칭 알고리즘을 바로 적용하기 어렵다. 본 논문은 첫째, 서로 다른 framerate에 대해서 각 영상의 평균값을 이용하여 새 이미지를 생성하고, 응용하여 framerate를 조정하는 방법을 제안하였다. 둘째, 이미지의 해상도 평균으로 최적의 해상도를 도출하여 해상도를 조정하는 방법을 제안하였다. 본 논문은 실험 결과를 통해, 제안하는 기술을 사용하면 서로 다른 특징을 갖는 영상에서도 스티칭 알고리즘이 원활하게 동작하는 것을 확인하였다.

### I. 서 론

영상 스티칭은 다수의 영상을 합성하여 카메라의 좁은 시야각(Field of View) 문제를 해결하는 기술이다. 이에 따른 최근 동영상 기반 Panorama, Super Resolution, 360 VR(Virtual Reality) 등의 콘텐츠 사용이 증가함에 따라, 보다 빠르고 정확한 영상 스티칭 기술의 필요성이 커지고 있다. 기술이 크게 발전하고 있는 만큼 수많은 종류의 영상(사진 및 동영상 등)이 다양한 기기(카메라, 스마트폰, 블랙박스, CCTV, 드론 등)에서 여러 가지의 촬영주체(기관/기업, 개인 등)에 의해 생산되고 있고 카메라들도 다양하게 많이 제공되고 있어, 수작업으로 분석해야 할 분량이 기하급수적으로 증가하고 있다. 영상분석을 위한 시간과 노력의 증가로 분석에 어려움을 겪거나 골든타임을 놓치는 경우가 발생할 가능성 높고 공공안전 분야에 구축된 대부분의 CCTV 시스템은 모니터링 요원의 감시활동에 의해 이루어지므로, 모니터링 요원의 피로도 및 집중도 등에 지나치게 의존하고 있다. 기존의 스티칭 프로그램은 원거리와 근거리 물체들 간의 시차가 다르기 때문에 다른 각도에서 같은 카메라로 같은 영상을 찍을지라도 상대적인 위치 차이가 존재하게 되고 카메라에 담겨지는 피사체가 동적이거나, 피사체의 형태가 수시로 바뀌는 경우, 물체의 일부가 잘리는 스티칭 영상을 얻기 쉽고 동적이지 않아도 카메라 그림자, 카메라 삼각대, 카메라 플래시 등 카메라와 함께 움직이는 경우에도 실제와 큰 차이가 보이는 파노라마 영상이 발생한다. 그리고 두 영상 밝기 차이가 클 경우 스티칭 영상에서 밝기가 급격히 변하는 결과가 나타난다.[1]

본 논문은 다양한 촬영기기에서 얻어지는 여러 종류의 영상에 대해서 framerate, 해상도를 조정하는 기능을 추가하여 스티칭

알고리즘을 원활하게 수행할 수 있는 알고리즘을 제안하였고, 실험을 통해 그 효율성을 확인하였다.

### II. 본론

#### 1. 기존 기술

##### 1.1 다양한 영상 촬영기기

현재 사용되고 있는 다양한 촬영기기에서 획득할 수 있는 영상의 특징은 표1과 같다.

기기	종류	codec	framerate	resolution
cctv	민간기업	H.264, H.265, MJPEG	15fps	2048 x 1536
	국가기관 (도로, 방범)	MP2G-2, H.264	30fps	1920 x 1080
블랙박스	아이나비 QUANTUM	H.265	30fps	2560 x 1440
	파인드라이브 Octagon2	H.264	30fps	2560 x 1440
스마트폰	삼성 갤럭시 S10	10-bit HEVC(H.265), H.264, VP9	30,60fps	3840 x 2160
	아이폰 11 PRO	HEVC, H.264, MPEG-4 Part 2, Motion JPEG, HDR10	24,30,60 fps	3840 x 2160
	LG V 50s ThinQ	H.263, H.264, H.265(HEVC), MPEG-4, VP8/9, xvid, MJPEG	60fps	2340 x 1080
드론	DJI	H.264, H.265	60fps	3840 x 2160

표 1. 다양한 촬영기기의 영상 특징[2,3]

##### 1.2 이미지 스티칭 과정

이미지 스티칭(Image Stitching)은 중첩되는 영역을 가진 여러 장의 사진을 오버랩 영역(Overlap Roi)에서 이어 붙여 하

나의 영상을 만드는 작업이다. 크게 그림 1과 같이 특징점 검출 (Features extraction), 특징점 매칭(Matching Features), 번들 조정(Bundle Adjustment), 영상 왜핑(Warping Images), 경계선 추출(Seam Finder), 경계선 블렌딩(Blending)의 순서대로 진행된다.



그림 1. Image Stitching Algorithm

먼저, SIFT(Scale Invariant Feature Transform), SURF(Speeded Up Robust Features), ORB(Oriented FAST and Rotated BRIEF), FAST(Features from Accelerated Segment Test) 등으로 특징점을 검출 (Features extraction)한다. 그리고 현재 영상의 검출된 특징점과 가장 유사한 중첩하는 영상의 특징점을 찾아 일대일로 매칭 (Matching Features)시킨다. 일반적으로 RANSAC (Random Sample Consensus)을 이용해 가장 많은 내외층 (In-lier)을 가진 매칭 모델을 선택하고, 이 모델을 벗어나는 이상치(Outlier)를 제거한다. 그리고 여러 장의 영상의 카메라 이동을 동시에 추정하는 번들 조정(Bundle Adjustment) 입력된 지상기준점(GCP)를 사용하여 카메라의 촬영위치를 추가로 조정하는 과정을 거친 후 여러 장의 영상이 하나의 사진으로 이어지도록 여러 장의 영상을 현재 이미지 좌표계상에서 구형, 평면형 등의 좌표계 상으로 픽셀을 매칭시키고, 다시 평면 이미지로 투영시켜 저장한다(Warping Images). 마지막으로, 이러한 영상 간의 오차가 적은 경계선을 추출(Seam Finder)하고 경계선을 중심으로 영상을 자연스럽게 이어주는 블렌딩을 한다. [1]

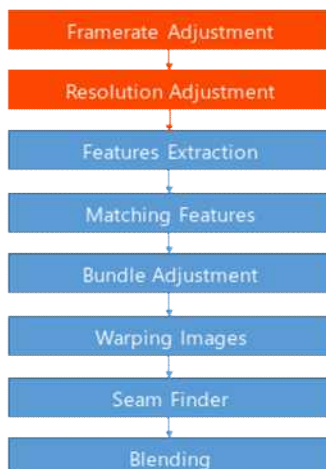


그림 2. 제안하는 Image Stitching Algorithm

## 2. 제안하는 아이디어

제안하는 스티칭 알고리즘은 기존의 OpenCV Stitcher 함수에 'framerate 조정' 과 '해상도 조정' 과정을 추가한 것으로 다양한 촬영기기 보급에 따른 동영상에 대해 문제없이 스티칭을 원활하게 수행하게 한다. 그림 2는 제안하는 아이디어의 순서도이다.

### 2.1 framerate 조정

framerate 조정은 framerate가 다른 영상에 대해서 framerate를 맞춰 주는 과정이다. 이 과정을 통해 framerate이 서로 다른 영상에 대해서 원활한 스티칭 영상을 만들어 줄 수 있다. 예를 들어 24fps 영상과 30fps 영상이 있다고 가정 그러면 24fps 영상의 이미지를 추가함으로써 30fps의 영상에서 만들어진 이미지 수를 같게 만들어 준다. 즉, 서로 다른 framerate 영상에 대해 framerate가 높은쪽으로 맞춘다. 부족한 frame 수를 전체 frame 수로 나눈 후 식(1)을 사용하여 각 지점마다 새 이미지를 생성해 부족한 framerate를 전 이미지와 후 이미지 평균값을 통해 맞춰준다.

$$dst(x,y) = saturate(src1(x,y)*\alpha + src2(x,y)*(x,y)*\beta + \gamma) \quad (1)$$

24fps 와 30fps에서 부족한 frame은 6이므로 24를 6으로 나누고 1/6, 2/6, 3/6, 4/6, 5/6, 6/6 지점에 식 (1)을 이용해 평균값으로 이미지를 생성한다.

### 2.2 해상도 조정

영상의 해상도는 각 촬영기기의 특성에 맞게 다양한 해상도로 구성된다. 기존의 스티칭 알고리즘들은 서로 다른 해상도에 대해 원활한 스티칭 영상을 제공하지 않는다. 예를 들어 그림 3과 같이, 원본 영상의 크기가  $w \times h$ 이고 확대한 영상의 크기가  $w' \times h'$ 이 된다. 원본 영상의 가로 픽셀의 크기가  $w$ 이고 확대한 영상의 가로 크기가  $w'$ 이기 때문에 가로 방향의 크기 변환 비율은  $s_x = w'/w$  수식으로 계산할 수 있다. 마찬가지로 세로 방향의 크기 변환 비율은  $s_y = h'/h$  수식으로 계산된다.

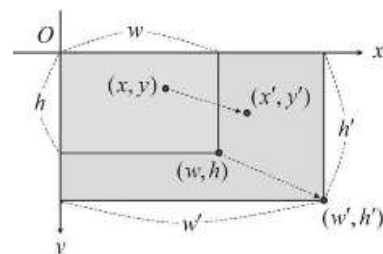


그림 3. 이미지의 크기 변환

입력 영상의 좌표  $(x, y)$ 로부터 크기 변환 결과 영상의 좌표  $(x', y')$ 를 계산하는 수식은 식(2)와 같다.

$$\begin{cases} x' = s_x x \\ y' = s_y y \end{cases} \quad \text{또는} \quad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (2)$$

영상의 크기 변환을 나타내는 어파인 변환 행렬  $M$ 은 식(3)과 같다.

$$M = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \end{pmatrix} \quad (3)$$

보간법은 결과 영상의 픽셀 값을 결정하기 위해 입력 영상에서 주변 픽셀 값을 이용하는 방식을 의미한다. 이러한 보간법 알고리즘을 사용하여 영상을 확대하거나 축소한다.[4]

### 2.2.1 공통 resolution

이미지 1의 해상도가 1920x1080 이고, 이미지 2의 해상도가 640x480이라고 가정할 때, 해상도는 이 둘의 평균인 1280x780으로 해상도를 조정한다. 그 이유는 한쪽 이미지를 너무 키우거나 줄이게 되면 스티칭 결과 영상이 원활하게 나오지 않을 가능성이 있기 때문이다. 그래서 본 논문에서는 두 이미지의 평균값으로 조정한다.

## III. 결론

### 1. 실험결과

서로 다른 frame에 대해서 그림4와 같은 일을 수행한다.

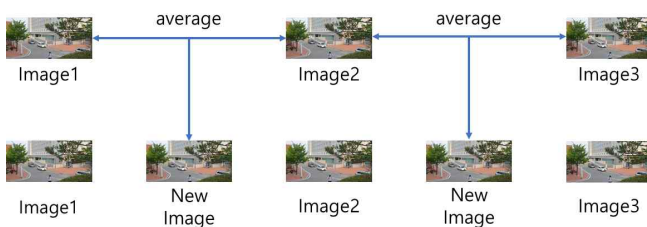


그림 4. framrate 조정

기존의 스티칭 알고리즘은 해상도가 다른 이미지에 대해서 스티칭 결과가 나오지 않는다. 본 논문에서 제한한 해상도 조정을 적용한 알고리즘 결과, 그림5 및 그림6과 같이 자연스러운 스티칭 결과 영상을 확인 하였다.

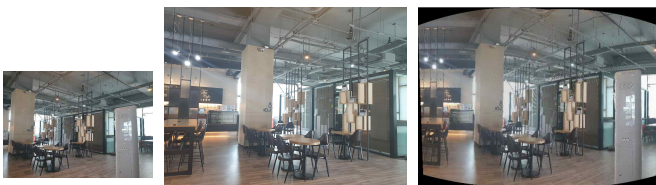


그림 5. 실험에 사용된 영상의 예시1

(좌: 해상도 640x480, 중: 해상도 4032x3024, 우: 스티칭 결과)

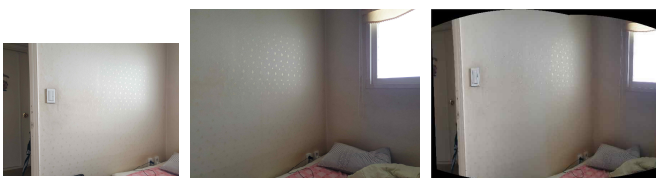


그림 6. 실험에 사용된 영상의 예시2

(좌: 해상도 640x480, 중: 해상도 1920x1080, 우: 스티칭 결과)

## 2. 고찰 및 추후과제

본 논문에서는 다양한 영상에서의 효과적인 이미지 스티칭 알고리즘을 제안하였다. 서로 다른 framerate 영상에 대해서 영상의 평균값을 이용하여 새로운 이미지를 만들고 두 영상에서 framerate가 큰쪽을 기준으로 framerate를 맞추므로써 스티칭 이미지를 스티칭 동영상으로 합칠 때의 문제점을 해결하였다. 그리고 기존 스티칭 알고리즘에서는 해상도 크기가 다르면 스티칭 영상이 원활하게 나오지 않게 된다. 따라서 모든 영상의 이미지 해상도를 조정하여 스티칭 하여 스티칭 영상이 원활하게 나올 수 있게 하였다. 이에 따라 실용적이고 효율적인 기술로 실무에 원활하게 보급이 가능하고 지속적인 영상특성(해상도 및 framerate 등)의 변화에 제한 없이 스티칭 알고리즘을 적용할 수 있다.

하지만 밝기 차이가 명확한 영상, 다양한 각도에서 찍힌 영상, 피사체 크기 차이에 대한 영상에 대해서 자연스러운 스티칭 영상을 도출하는 것에는 어려움이 있다. 이를 보완하기 위한 다음 연구에서는, 밝기 정보를 조절하는 과정을 추가하고 다양한 각도, 피사체 크기 차이에 대해 해결책을 제시하는 것이 추후 과제이다.

## ACKNOWLEDGMENT

본 연구는 2018년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2018RIC1B5043326).

## 참 고 문 헌

- [1] .방송공학회논문지, 「360 VR을 구성하는 영상들 간 밝기 차이를 이용한 seam finding 알고리즘」 p897-p898 2018.11
- [2] 갤럭시 S10 자급제 (프리즘 화이트, 128G) | Samsung 대한민국 <https://www.samsung.com/sec/smartphones/galaxy-s10-g973/SM-G973NZWAKOO/>
- [3] iPhone 11 Pro - 제품 사양 - Apple (KR) <https://www.apple.com/kr/iphone-11-pro/specs/>
- [4] .길벗, 「OpenCV 4로 배우는 컴퓨터 비전과 머신 러닝」 p233, p288 2019.04.12.