

# 효율적인 영상 포렌식을 위한 Image Stitching 기법의 구현

유 신\*, 김 한 수<sup>o</sup>

## An Implementation of Image Stitching Method for Efficient Image Forensics

Shin Ryou\*, Hansoo Kim<sup>o</sup>

요 약

범죄 수사, 화각 확보, 위치 인식, 관광 및 스포츠 관람 등에서 여러 사진 및 영상들을 이어붙이는 영상 스티칭(image stitching) 기술이 많이 활용되고 있다. 스티칭 영상을 구성하는 과정에서 합성되는 서로 다른 영상들의 다양한 특징, 즉 해상도(resolution), 프레임속도(framerate) 등을 빠르고 효율적으로 조정하는 것은 스티칭 알고리즘을 성공적으로 적용하는데 중요한 요소이다. 현재 많이 사용되는 영상들은 서로 형식이 달라 기존의 스티칭 알고리즘을 바로 적용하기 어렵다. 본 논문은 첫째, 서로 다른 프레임속도에 대해서 각 영상의 평균값을 이용하여 새 이미지를 생성하고, 이를 응용하여 프레임속도를 조정하는 방법을 제안하였다. 둘째, 이미지의 해상도 평균으로 최적의 해상도를 도출하여 해상도를 조정하는 방법을 제안하였다. 이를 검증하기 위해, 알려져 있는 물체 추적(object tracking) 기법을 스티칭된 영상에 적용하여 각 이미지에서 식별한 물체가 동일 물체인지 여부를 확인하였다. 본 논문은 실험 결과를 통해, 제안하는 기술을 사용하면 서로 다른 특징을 갖는 영상에서도 스티칭 알고리즘이 원활하게 동작하는 것을 검증하였고 스티칭된 영상에서의 물체 추적을 통해 동일 물체 식별 여부를 확인하였다.

키워드 : 영상 포렌식, 해상도, 프레임속도, 영상 스티칭

Key Words : Image Forensics, Resolution, Framerate, Image Stitching

### ABSTRACT

Image stitching technology, that connects multiple photographs and videos into one image or video, has been widely used in criminal investigation, securing a field of view, location recognition, and even in tourism and sports. Adjusting the features such as resolution and framerate from different images is an important factor in enhancing the completeness of the stitched image. It is difficult to apply the existing stitching algorithm directly to the various kinds of images that have different features. In this paper, three main schemes are devised to create the stitched image properly and to secure the identity and continuity of the object inside the images. First, new frames are created according to the average pixel value of adjacent frames, to adjust the different framerates. And, new resolution is derived regarding the average value of different resolutions. To verify this, the well-known object tracking framework has been applied to the stitched image so that the

\* 이 성과는 2018년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2018R1C1B5043326).

• First Author : IT Division((Information Security), Seowon University, ryoushin@naver.com, 학생회원

o Corresponding Author : IT Division((Information Security), Seowon University, kutestar@seowon.ac.kr, 정회원

논문번호 : 202012-340-C-RE, Received December 30, 2020; Revised February 21, 2021; Accepted March 2, 2021

identity and continuity of the object from different images is secured. Experimental Results show that the stitching method with the proposed schemes works properly, in various images with different features. Also, objects in different images are clearly secured as identical.

## I. 서론

영상 스티칭(image stitching)은 다수의 영상을 합성하여 카메라의 좁은 시야각(Field of View) 문제를 해결하는 기술이다<sup>[1]</sup>. 최근 동영상 기반의 파노라마(Panorama), 초고화질(Super Resolution), 360 VR(Virtual Reality) 등의 콘텐츠 사용이 증가함에 따라, 보다 빠르고 정확한 영상 스티칭 기술의 필요성이 커지고 있다. 특히 범죄 수사를 위한 영상 포렌식(image/video forensics), 피사체의 위치 인식 및 추적, 주차 등을 위한 시야 확보, 관광 및 스포츠영화 관람에 이르기까지 활용 분야가 넓어지며, 영상 스티칭의 중요성은 날로 커져 가고 있다.

기술이 빠르게 발전함에 따라, 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 있어, 범죄 수사 및 위치 추적 등에서 수작업으로 분석해야 할 영상의 분량이 기하급수적으로 증가하고 있다. 그러나, 공공안전 분야에 구축된 대부분의 CCTV 시스템은 모니터링 요원의 감시활동에 의해 이루어지므로, 모니터링 요원의 피로도 및 집중도 등에 지나치게 의존하고 있다. 즉, 기술의 발달로 인해 영상 증거의 양은 늘어났으나, 그만큼 복잡하고 정형화되어 있지 않은 영상 정보들로 인해 분석과 활용에 어려움을 겪고 영상분석을 위한 시간과 노력의 증가로 분석에 어려움을 겪거나 골든타임을 놓치는 경우가 발생할 가능성이 높아지게 된다.

영상 분석 분야는 오래전부터 연구되어 온 주제이나, 극도의 저화질 영상의 분석 또는 수사에 활용하기 위한 포렌식 관점에서의 접근은 매우 제한적이다. 기존의 스티칭 프로그램은 다른 각도에서 촬영된 영상인 경우 피사체들간의 상대적인 위치 및 거리 차이에 따른 문제가 발생하게 되고, 피사체가 움직이거나 그 형태가 수시로 바뀌는 경우 피사체의 일부가 잘리는 스티칭 영상을 얻기 쉬우며, 피사체가 정지해 있는 경우에도 카메라의 그림자, 삼각대 및 플래시 등의 영향에 따라, 또는 두 영상의 밝기 차이에 따라 원활한 스티칭 영상을 얻기 어려운 경우가 발생한다<sup>[1]</sup>.

본 논문은 다양한 촬영기기에서 얻어지는 여러 종

류의 영상에 대해서 해상도(resolution), 프레임속도( framerate)를 조정하는 기능을 추가하여 스티칭 알고리즘을 원활하게 수행할 수 있는 알고리즘을 제안하였고, 실험을 통해 그 효율성을 확인하였다. 스티칭된 영상에 알려져 있는 물체 추적(object tracking)을 적용하여, 서로 다른 이미지에서 식별한 물체가 동일 물체인지를 확인함으로써 본 알고리즘을 검증하였다.

본 논문의 순서는 다음과 같다. 2장에서는 다양한 촬영기기, 영상 처리의 필요성, 영상 스티칭 및 물체 추적과 같은 기존 기술에 대해 자세히 서술하고, 3장에서는 본 논문에서 제안하는 아이디어 적용 과정을 세부적으로 설명한다. 4장에서는 실험 결과를 바탕으로 성능을 분석하고, 마지막 5장에서는 고찰 및 추후 과제로 결론을 맺는다.

## II. 기존 기술

### 2.1 다양한 영상 촬영기기

현재 사용되고 있는 다양한 촬영기기에서 획득할 수 있는 영상의 특징은 표 1과 같다.

표 1. 다양한 촬영기기의 영상 특징[2,3]  
Table 1. Image Characteristics from Various Imaging Devices[2,3]

device	model	codec	framerate	resolution
cctv	corporation company	H.264, H.265, MJPEG	15fps	2048 x 1536
	state institution	MP2G-2, H.264	30fps	1920 x 1080
Blackbox	INAVI QUANTUM	H.265	30fps	2560 x 1440
	Fine-drive Octagon2	H.264	30fps	2560 x 1440
Smart Phone	Samsung Galaxy S10	HEVC(H.265), H.264	30, 60fps	3840 x 2160
	iPhone 11 PRO	HEVC, H.264, MPEG-4	24, 30, 60fps	3840 x 2160
	LG V 50s ThinQ	H.264, H.265, MPEG-4	60fps	2340 x 1080
Drone	DJI	H.264, H.265	60fps	3840 x 2160

### 2.2 최근 연구 동향 및 성과

국내 영상보안시장에서 지능형 관제가 도입된 것은 2010년 초로, 움직임 분석과 규칙세트(ruleset) 기반의 패턴분석을 활용했으나, 당시의 '지능형 영상관제'는 감지 수준에 불과하였다. 분석 오류가 28.2%로 사람이 용인하기 어려운 수준이었고, 다양한 환경 변화에 대응하지 못할 정도로 오탐(false positive)도 빈번하였다<sup>4)</sup>.

2012년 이후 딥 러닝(deep learning) 기술이 진화하면서 오탐률은 2012년 18.4%에서 2015년 3.57%로 획기적인 개선을 이루었고 얼굴인식은 95% 선으로 정확해져 사람의 눈과 비슷한 수준으로 발전하였다.

2014년 이후 CCTV를 활용한 실시간 범인검거 건수는 총 10만 8천 294건으로, 14년 1천 627건에서 18년 3만 1천 142건으로 20배 이상 증가하였으며, 범죄 유형별로는 폭력범죄가 6만 6천 50건으로 가장 많았고, 절도 9천 407건, 강간 1천 661건, 강도 55건 순으로 나타나, CCTV가 해를 거듭할수록 범죄해결에 큰 역할을 담당하였다(표 2 참조)<sup>5)</sup>.

2018년 5월 22일 진주시 동성동에서 발생한 뺑소니 사망사건 수사를 위해, 경찰은 통합관제센터 CCTV를 통해 사고 현장 통과 차량을 분석, 범인택시 2000여 대를 탐문하여 용의자를 검거하였다. 진주시는 2020년 5월까지 12억 원을 투자해 긴급상황 발생 시 주변 CCTV 영상을 유관기관에 실시간 제공하는 안전망 구축을 위한 CCTV 영상관제 기반의 스마트 시티 통합 플랫폼 구축 사업을 추진하였다<sup>7)</sup>. 제천/밀양 대형 화재 이후 범국가적 차원의 재난재해 대응을 위한 시스템의 발족이 추진 중에 있으며, CCTV 등의 관제 시스템, 스마트폰 및 블랙박스 등의 제보 시스템이 주요 이슈로 부각되고 있다.

표 2. 2014년 이후 CCTV를 활용한 실시간 범인검거 현황 (경찰청)<sup>5)</sup>  
Table 2. Cases Solved in Real-Time using CCTVs since 2014 (NPA)<sup>5)</sup>

연도	범인검거 현황						
	소계	살인	강도	강간	절도	폭력	기타 형사범
14	1,627	1	14	22	737	347	506
15	10,157	2	16	242	1,445	5,250	3,376
16	20,430	3	7	354	1,817	12,358	5,891
17	28,004	0	7	483	2,285	17,388	7,841
18	31,142	0	6	403	1,786	20,852	8,095
19.6	16,934	0	5	157	1,337	9,855	5,436
합계	108,294	6	55	1,661	9,407	66,050	31,145

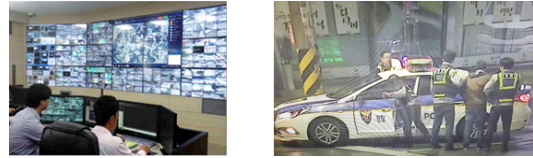


그림 1. 영상분석을 통한 국내 사건 해결 및 검거<sup>6)</sup>  
Fig. 1. Domestic Criminal Cases Solved using Image Analysis<sup>6)</sup>

### 2.3 영상 스티칭 과정

영상 스티칭은 중첩되는 영역을 가진 여러 장의 사진을 오버랩 영역(overlapped ROI)에서 이어 붙여 하나의 영상을 만드는 작업이다. 크게 그림 2와 같이 특징점 검출(Features extraction), 특징점 매칭(Matching Features), 번들 조정(Bundle Adjustment), 영상 왜곡(Warping Images), 경계선 추출(Seam Finder), 합성(Blending)의 순서대로 진행된다<sup>11)</sup>.

먼저, SIFT(Scale Invariant Feature Transform)<sup>18,9)</sup>, SURF(Speeded Up Robust Features)<sup>10)</sup>, ORB(Oriented FAST and Rotated BRIEF)<sup>11)</sup>, FAST(Features from Accelerated Segment Test)<sup>12)</sup> 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)<sup>13)</sup>을 이용해 가장 많은 정상치(Inlier)를 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, ground control point)을 사용하여 카메라의 촬영위치를 추가로 조정하는 과정을 거친 후, 여러 장의 영상이 하나의 사진으로 이어지도록 여러 장의 영상을 현재 이미지 좌표



그림 2. 기존의 영상 스티칭 알고리즘  
Fig. 2. Conventional Image Stitching Algorithm

계상에서 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 합성(Blending)을 수행한다<sup>11)</sup>.

### 2.4 물체 추적

객체 인식(object recognition)은 이미지 또는 비디오 상의 객체를 식별하는 컴퓨터 비전 기술로, 객체 인식은 딥 러닝과 머신 러닝 알고리즘을 통해 산출되는 핵심 기술이다. 사람은 사진 또는 비디오를 볼 때 인물, 물체, 장면 및 시각적 세부 사항을 쉽게 알아볼 수 있는데, 이처럼 사람이려면 당연히 할 수 있는 일을 컴퓨터가 할 수 있도록 하는 기술을 말한다<sup>14)</sup>.

딥 러닝을 사용한 객체 인식 기술 중 합성곱 신경망(CNN, convolutional neural network)과 같은 딥 러닝 모델은 객체를 식별하기 위해 해당 객체 고유의 특징을 자동으로 학습하는 데 사용된다. CNN에서는 수천 장의 훈련용 이미지를 분석하고, 고양이와 개를 구분하는 특징을 학습하여 고양이와 개의 차이점을 식별하는 방법을 학습할 수 있다<sup>14)</sup>.

최근 딥 러닝 기반의 객체 탐지 기법인 YOLO(You Only Look Once)를 사용한 연구가 보고되고 있다. YOLO는 영상 내 객체를 탐지하는 딥 러닝 기법으로 실시간 처리가 가능한 빠른 수행 속도가 장점인 기법이다<sup>15)</sup>. inception v3 또한 널리 사용되는 이미지 인식 모델로서, ImageNet 데이터셋에서 78.1% 이상의 정확성을 실현하는 것으로 나타났다<sup>16)</sup>.

## III. 제안하는 아이디어

제안하는 스티칭 알고리즘은 그림 3과 같은 기존의 스티칭 알고리즘에 ‘프레임속도 조정’ 과 ‘해상도 조정’ 과정을 추가한 것으로 다양한 촬영기기 및 영상 형식에도 문제없이 스티칭을 원활하게 수행하게 한다. 그림 3은 제안하는 아이디어의 순서도이다. 이후 물체 추적 기술을 사용하여 식별한 물체가 동일한 물체인지 확인한다.

### 3.1 프레임속도 조정

프레임속도 조정은 프레임속도가 다른 영상에 대해서 이를 같게 맞춰 주는 과정이다. 이 과정을 통해 프레임속도가 서로 다른 영상에 대해서 원활한 스티칭 영상을 만들어 줄 수 있다. 24fps(frame per second,

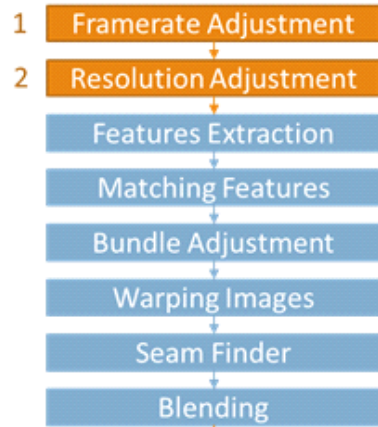


그림 3. 제안하는 영상 스티칭 알고리즘  
Fig. 3. Proposed Image Stitching Algorithm

초당 프레임 수) 영상과 30fps 영상이 있다고 가정하면, 24fps 영상에 이미지 6 프레임을 추가함으로써 30fps의 영상에서 만들어진 이미지와 프레임 수를 같게 만들어 준다. 이 때, 부족한 프레임 수를 전체 프레임 수로 나눈 후 식(1)을 사용하여 새 이미지들을 생성한다.

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

식(1)에서  $\alpha$ 와  $\beta$ 는 각각 src1과 src2 영상의 가중치를 의미하는 수이고,  $\alpha + \beta = 1$ 이 된다.

24fps 와 30fps에서 부족한 프레임은 6이므로 24를 6으로 나누고 1/6, 2/6, 3/6, 4/6, 5/6, 6/6 지점에 식(1)을 이용한 픽셀 값의 가중 평균값으로 이미지를 생성한다.

### 3.2 해상도 조정

영상의 해상도는 각 촬영기기의 특성에 맞게 다양한 해상도로 구성된다. 기존의 스티칭 알고리즘들은 서로 다른 해상도에 대해 원활한 스티칭 영상을 제공하지 않는다. 따라서, 스티칭하려는 두 영상의 해상도를 확대 또는 축소하여 서로 일치시키는 과정을 거친다. 예를 들어 그림 4와 같이, 원본 영상의 크기가  $w \times h$ 이고 확대한 영상의 크기가  $w' \times h'$ 일 때 원본 영상의 가로 픽셀의 크기가  $w$ 이고 확대한 영상의 가로 크기가  $w'$ 이기 때문에 가로 방향의 크기 변환 비율  $s_x$ 는  $s_x = w'/w$  로 계산할 수 있다. 마찬가지로 세로 방향의 크기 변환 비율  $s_y$ 는  $s_y = h'/h$  로 계산된다.

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

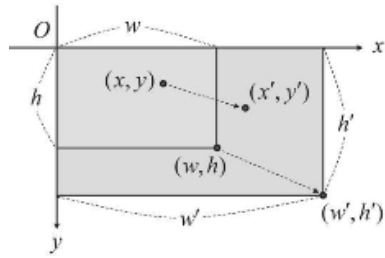


그림 4. 이미지의 크기 변환  
Fig. 4. Image Size Transformation

$$\begin{cases} x' = s_x x \\ y' = s_y y \end{cases} \quad (2)$$

식(2)에서  $s_x$ 는 가로 방향의 크기 변환 비율,  $s_y$ 는 세로 방향의 크기 변환 비율을 의미한다. 일반적으로 영상의 크기 변환을 나타내는 어파인(affine) 변환 행렬  $M$ 은 식(3)과 같다.

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

마찬가지로 식(3)에서도  $s_x$ 는 가로 방향의 크기 변환 비율,  $s_y$ 는 세로 방향의 크기 변환 비율을 의미한다.

스티칭하려는 이미지 1의 해상도가 1920x1080 이고, 이미지 2의 해상도가 640x480이라고 가정할 때, 해상도는 이 둘의 평균인 1280x780으로 정한다. 그 이유는 한쪽 이미지를 너무 키우거나 줄이게 되면 스티칭 결과 영상이 원활하게 나오지 않을 가능성이 있기 때문이다. 그래서 본 논문에서는 두 이미지 해상도의 평균값을 최대한 찾아, 스티칭하기 전의 공통 해상도로 정한다.

### 3.3 보간법(interpolation)

보간법은 결과 영상의 픽셀 값을 결정하기 위해 입력 영상에서 주변 픽셀 값을 이용하는 방식을 의미한다. 이러한 보간법 알고리즘을 사용하여 영상을 확대하거나 축소한다.<sup>[16]</sup> 보간법에는 최근방 이웃(nearest neighbor) 보간법, 양선형(bilinear) 보간법, 양입방(bicubic) 보간법, 3차(third-order polynomial) 보간법, 란초스(Lanczos) 보간법 등이 있다. 각각의 특징으로는 최근방 이웃 보간법이 가장 빠르게 동작한다는 장점이 있지만 결과 영상 화질이 좋지 않고, 양선형 보간법 및 양입방 보간법은 연산 속도가 빠르고 화질도 충분히 좋은 편이어서 널리 사용되고 있으며, 3차 보간법 또는 란초스 보간법은 화질이 양선형 보간법보

다 좋지 않지만 연산속도가 느려진다.<sup>[17]</sup>

본 논문에서는 속도와 화질을 균등하게 하기 위해 양입방 보간법을 사용한다.

## IV. 실험 결과

### 4.1 실험 환경

본 논문에서 제안한 방법은 Visual Studio 2017을 사용한 C++, OpevCV 4.1.2, YOLO v3의 환경에서 구현하였고, 실험을 위한 데이터는 삼성사의 갤럭시 노트 4, 갤럭시 s8, 갤럭시 노트 9, Apple사의 아이폰 6s, 아이폰 11Pro 등으로 촬영한 영상을 활용하였다 (세부사항은 표 3 참조)

표 3. 실험에 사용된 영상 특징  
Table 3. Image Characteristics used in Experiments

manufacturer	model	codec	framerate	resolution
Samsung	Galaxy S8	H.264, MPEG-4, JPEG	30, 60fps	2960 x 1440
	Galaxy 9	10-bit HEVC(H.265), H.264, VP9	30, 60fps	4032 x 3024
	Galaxy 4	H.264, MPEG-4, JPEG	24, 30fps	2560 x 1440
Apple	iPhone 11PRO	HEVC, H.264, MPEG-4 Part 2, Motion JPEG, HDR10	24, 30, 60fps	4032 x 3024
	iPhone 6s	H.264, MPEG-4, JPEG	24, 30fps	1334 x 750

### 4.2 실험 과정

서로 다른 프레임속도를 가진 영상들에 대해서 그림 5와 같은 작업을 수행하여 프레임속도를 맞춘다.



그림 5. 프레임속도 조정 과정  
Fig. 5. Framerate Adjustment Process

그림 6에서, 프레임속도가 낮은 1번 영상의 전후 이미지의 평균을 이용해 새로운 이미지를 만들어 주어 2번 영상과 동일한 프레임속도가 되게 한 후 영상 스티칭을 적용하였다.

기존의 스티칭 알고리즘은 해상도가 다른 이미지에 대해서 스티칭 결과가 나오지 않는다. 본 논문에서 제안한 해상도 조정을 적용한 알고리즘 결과, 그림 7에서 10까지와 같이 자연스러운 스티칭 결과 영상을 확인하였다. 또한 그림 11에서 13까지와 같이 해상도가 다른 3장의 이미지에 대해서도 자연스러운 스티칭 결과를 확인하였다.

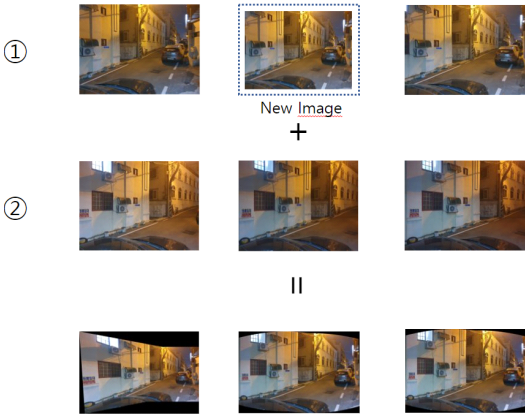
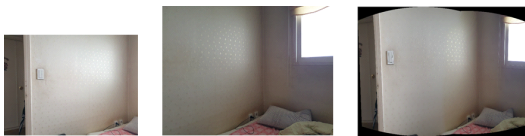


그림 6. 프레임속도 조정 예시  
Fig. 6. Example of the Framerate Adjustment



(좌: 해상도 640x480, 중: 해상도 4032x3024, 우: 스티칭 결과)  
그림 7. 실험에 사용된 영상의 예시1  
Fig. 7. Image Example 1 from the Experiments



(좌: 해상도 640x480, 중: 해상도 1920x1080, 우: 스티칭 결과)  
그림 8. 실험에 사용된 영상의 예시2  
Fig. 8. Image Example 2 from the Experiments



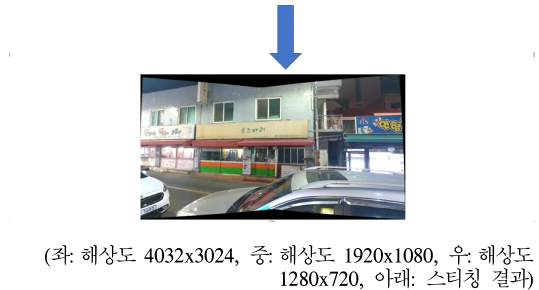
(좌: 해상도 640x480, 중: 해상도 1280x720, 우: 스티칭 결과)  
그림 9. 실험에 사용된 영상의 예시3  
Fig. 9. Image Example 3 from the Experiments



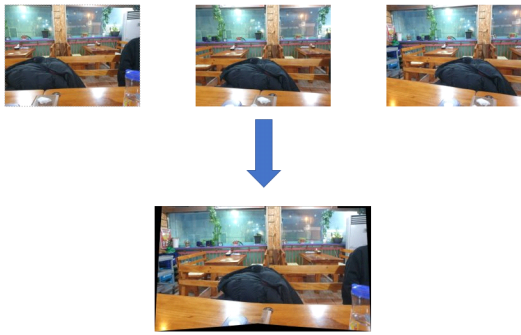
(좌: 해상도 1280x720, 중: 해상도 1920x1080, 우: 스티칭 결과)  
그림 10. 실험에 사용된 영상의 예시4  
Fig. 10. Image Example 4 from the Experiments



(좌: 해상도 640x480, 중: 해상도 1920x1080, 우: 해상도 1280x720, 아래: 스티칭 결과)  
그림 11. 실험에 사용된 영상의 예시5  
Fig. 11. Image Example 5 from the Experiments



(좌: 해상도 4032x3024, 중: 해상도 1920x1080, 우: 해상도 1280x720, 아래: 스티칭 결과)  
그림 12. 실험에 사용된 영상의 예시6  
Fig. 12. Image Example 6 from the Experiments



(좌: 해상도 4032x3024, 중: 해상도 1920x1080, 우: 해상도 640x480, 아래: 스티칭 결과)

그림 13. 실험에 사용된 영상의 예시7  
Fig. 13. Image Example 7 from the Experiments

딥 러닝 프레임워크(deep learning framework) 중 하나인 YOLO를 사용하여 물체 추적을 수행한 결과, 그림 14 및 그림 15와 같이 각 이미지에서 식별한 물체가 스티칭 이미지에 대해서도 동일한 물체를 식별하는 것을 확인하였다.



(좌: 이미지 1, 중: 이미지 2, 우: 스티칭 결과)

그림 14. 실험에 사용된 물체 인식의 예시1  
Fig. 14. Object Tracking Example 1 from the Experiments



(좌: 이미지 1, 중: 이미지 2, 우: 스티칭 결과)

그림 15. 실험에 사용된 물체 인식의 예시2  
Fig. 15. Object Tracking Example 2 from the Experiments

### V. 고찰 및 추후 과제

범죄 수사를 위한 영상 포렌식, 피사체의 위치 인식 및 추적, 시야 확보 및 스포츠/영화 관람 등에서 영상 스티칭이 많이 활용되고 있으나, 서로 다른 특징을 가진 다양한 영상들을 스티칭하는 데에 많은 어려움이 있다. 본 논문에서는 다양한 영상에서의 효과적인 이미지 스티칭 기법을 제안하였다. 서로 다른 프레임속도 영상에 대해서 각 프레임속도의 평균값을 이용하

여 새로운 이미지를 만들고, 두 영상에서 프레임속도가 큰 쪽을 기준으로 프레임속도를 맞춤으로서 스티칭 이미지를 스티칭 동영상으로 합칠 때의 문제점을 해결하였다. 그리고 기존 스티칭 알고리즘에서는 해상도 크기가 다르면 스티칭 영상이 원활하게 나오지 않게 되므로, 모든 영상의 이미지 해상도를 조정하여 스티칭 영상이 원활하게 나올수 있게 하였다. 마지막으로 스티칭 후 만들어진 영상에 대해 알려져 있는 물체 추적 알고리즘을 적용하여 각 이미지에서 식별한 물체가 동일 물체인지 여부를 확인하였다.

이러한 실용적이고 효율적인 기술을 실무에 활용하면, 다양한 기기에서 촬영되는 여러 가지의 영상 특성(해상도 및 프레임속도 등)의 변화에 제한 없이 스티칭 알고리즘을 적용할 수 있고 빠르게 특정 피사체를 식별할 수 있어, 영상 포렌식 및 피사체의 동일 여부 판독, 위치 인식 및 추적 등에 정확성과 편리함을 가져올 수 있을 것이다.

밝기 차이가 두드러지는 영상, 다양한 각도에서 찍힌 피사체, 피사체의 크기 차이가 많은 영상에 대해서 자연스러운 스티칭 영상을 도출하는 것에는 어려움이 있다. 추후 이를 보완하기 위하여 밝기 정보를 조절하는 과정을 추가하고, 다양한 각도 및 크기(촬영거리가 다른 동일 피사체 등)를 갖는 피사체에 대해 효율적으로 스티칭이 이루어지도록 하는 연구를 지속해야 할 것이다.

### References

- [1] D. Nam and J. Han, "Seam finding algorithm using the brightness difference between pictures in 360 VR," *J. Broadcast Eng.*, vol. 23, no. 6, pp. 897-898, Nov. 2018.
- [2] Samsung, *Galaxy S10 self-sufficiency (Prism White, 128G) | Samsung South Korea*(2019), Retrieved Feb. 13, 2020, from <https://www.samsung.com/sec/smartphones/galaxy-s10-g973/SM-G973NZWAKOO/>
- [3] Apple, *iPhone 11 Pro - Product Specifications - Apple (KR)*(2019), Retrieved Feb. 13, 2020, from <https://www.apple.com/kr/iphone-11-pro/specs/>
- [4] H. S. Eom, *The closing of the video security market in 2019 and the outlook for 2020*(2019), Retrieved Nov. 20, 2020, from <https://www.boannnews.com/media/view.asp?id>

x=84848&kind=0

- [5] S. K. Hwa, *Since 2014, more than 100,000 real-time criminals have been detected using CCTV*(2019), Retrieved Nov. 20, 2020, from <http://www.goodnews365.net/news/articleView.html?idxno=124644>
- [6] S. S. KIM, *Video Analysis Technology Fruiting with Intelligent CCTV*(2019), Retrieved Nov. 20, 2020, from <https://www.coworld.co.kr/news/articleView.html?idxno=49689>
- [7] News人 Jeonbuk, *Jeonju City Construction of Civil Safety Net by Sharing CCTV Video Information*(2020), Retrieved Nov. 20, 2020, from <http://www.newsinjb.com/news/articleView.html?idxno=11380>
- [8] G. L. David, "Distinctive image features from scale-invariant key-points," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91-110, Nov. 2004.
- [9] W. J. Moon, Y. H. Seo, and D. W. Kim, "Parameter analysis for time reduction in extracting SIFT keypoints in the aspect of image stitching," *J. Broadcast Eng.*, vol. 23, no. 4, pp. 559-573, Jul. 2018.
- [10] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vision and Image Understanding*, vol. 10, no. 3, pp. 346-359, 2008.
- [11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," *Int. Conf. Computer Vision*, vol. 1, pp. 2564-2571, Nov. 2011.
- [12] S. Y. Park, J. Kim, and J. S. Yoo, "Fast stitching algorithm by using feature tracking," *J. Broadcast Eng.*, vol. 20, no. 5, pp. 728-737, Sep. 2015.
- [13] A. F. Martin and C. B. Robert, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381-395, Jun. 1981.
- [14] H. Y. Jeong, *Artificial intelligence object identification, three things you must know*(2018), Retrieved Nov. 20, 2020, from [xno=12087](http://www.aitimes.kr/news/articleView.html?id</a></p>
</div>
<div data-bbox=)

- [15] H. Lee, J. Sa, H. Shin, Y. Chung, D. Park, and H. Kim, "Separation of occluding pigs using deep learning-based image processing techniques," *J. Korea Multimedia Soc.*, vol. 22, no. 2, pp. 136-145, Feb. 2019.
- [16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojha, "Rethinking the inception architecture for computer vision," *2016 IEEE CVPR*, pp. 2818-2826, Las Vegas, NV, 2016.
- [17] S. Hwang, *Computer Vision and Machine Learning with OpenCV 4*, Book Publishing Gilbut, p. 233, p. 288, Apr. 2019.

유 신 (Shin Ryou)



2022년 2월 : 서원대학교 정보  
보안학과 졸업예정  
<관심분야> 머신러닝, 딥 러닝,  
영상처리  
[ORCID:0000-0003-1157-6411]

김 한 수 (Hansoo Kim)



2002년 2월 : 서강대학교 전자  
공학과 (학사)  
2004년 2월 : 서강대학교 전자  
공학과 (석사)  
2014년 8월 : 서강대학교 전자  
공학과 (박사)  
2017년 3월~현재 : 서원대학교  
정보보안학과 조교수  
<관심분야> 영상분석, 네트워크 프로토콜 설계, 알  
고리즘 디자인  
[ORCID:0000-0003-3357-9832]