

일반논문 (Regular Paper)

방송공학회논문지 제23권 제5호, 2018년 9월 (JBE Vol. 23, No. 5, September 2018)

<https://doi.org/10.5909/JBE.2018.23.5.656>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

개선된 Seam Finder를 이용한 360 VR 이미지 스티칭 기술

손희정^{a)}, 한종기^{a)†}

Advanced Seam Finding Algorithm for Stitching of 360 VR Images

Hui-Jeong Son^{a)} and Jong-Ki Han^{a)†}

요약

스티칭 기술은 고화질의 360 VR 영상을 제작하는 과정에서 가장 중요한 요소 기술들 중의 하나이다. 스티칭 기술의 성능을 저하시키는 원인들에는 특징점 추출 과정의 오류, seam finding 과정에서 사용되는 마스크의 왜곡으로 발생하는 오류, 각 영상들의 밝기 보상 오류 등 다양한 원인들이 존재한다. 본 논문에서는 합성되는 각 영상들 간의 시차(View Disparity)가 존재함으로써 스티칭 성능이 저하되는 현상을 분석하고, 이 문제를 해결하기 위해 이음부 탐색(seam finding)의 전처리 과정에서 사용되는 효율적인 알고리즘을 제안한다. 본 논문에서 제안하는 기술을 통해 기존 방법보다 개선된 마스크들을 제작하여 효율적인 이음부 탐색(seam finding)이 수행되도록 하고, 그 결과 개선된 화질을 갖는 360 VR 영상을 얻을 수 있음을 설명한다. 본 논문에서 실시된 다양한 실험들을 통해, 제안된 기술이 시차 왜곡이 존재하는 영상 신호들을 합성하는 과정에 효율적이면서 동작 복잡도도 높지 않음을 보이고 있다.

Abstract

VR (Virtual Reality) is one of the important research topics in the field of multimedia application system. The quality of the visual data composed from multiple pictures depends on the performance of stitching technique. The stitching module consists of feature extraction, mapping of those, warping, seam finding, and blending. In this paper, we proposed a preprocessing scheme to provide the efficient mask for seam finder. Incorporating of the proposed mask removes the distortion, such as ghost and blurring, in the stitched image. The simulation results show that the proposed algorithm outperforms other conventional techniques in the respect of the subjective quality and the computational complexity.

Keyword : 360 VR, Stitching, Stereo Image, Blending, Seam Finder

a) 세종대학교 전자정보통신공학과(Sejong University, Dept. of Electrical Engineering)

† Corresponding Author : 한종기(Jong-Ki Han)

E-mail: hjk@sejong.edu

Tel: +82-2-3408-3739

ORCID:<https://orcid.org/0000-0002-5036-7199>

※ This work was partly supported by the National Research Foundation of Korea (NRF) under Grant NRF-2018R1A2A2A05023117 and partly by Institute for Information & communications Technology Promotion (IITP) under Grant 2017-0-00486 funded by the Korea government (MSIT).

· Manuscript received July 11, 2018; Revised August 20, 2018; Accepted August 21, 2018.

1. 서론

360 VR은 영화, 공연, 스포츠, 게임, 부동산, 숙박시설, 쇼핑, 소셜 커뮤니티 등에서 활용 가능한 차세대 주요 기술이다. 360 VR 기술은 컴퓨터 그래픽 기술을 활용하여 3차원 가상 공간을 구성하고, 네트워크 및 통신 기술을 이용하여 가상현실 공간과 현재 공간 사이를 실시간으로 연결하며, 사용자에게 몰입형 시청각 체험을 제공함으로써, 현장감이 넘치는 멀티미디어 서비스를 제공하는 신기술이다^[1]. 이 360 VR 기술을 효율적으로 구현하기 위해서 필요한 세부 기술들에는 3차원 공간을 2차원에 구성하는 영상처리, 방대한 정보량을 통신하기 위한 압축 기술, 사용자에게 빠르게 반응하기 위한 초고속 대용량 통신 기술 등이 필요한데, 이러한 기술적 요구조건을 만족시키는 통신 인프라 기술이 5G 이동통신기술이라고 알려져 있다^[2].

360 VR 관련 연구는 세계 여러 기관들이 진행하고 있는데, 최근에는 국제 표준화 단체 JVET(Joint Video Exploration Team) 및 MPEG(Moving Picture Expert Group) 등에서 360 VR 비디오 신호의 표현 및 부호화 기술들에 관한 표준화를 진행하고 있다^[3]. JVET에서는 360 VR을 포함하는 고해상도 HDR 비디오 영상을 H.265보다 더 효율적으로 부호화하는 기술을 표준화하고 있고, 3D 좌표와 평면사이 투영방법^[4], 360 VR 코덱 기술에 대한 객관적인 화질 평가 지표와 실험 조건 등을 규정하고 있다^[5]. 한편 MPEG에서는 MPEG-I라고 명명된 표준화 분과에서 3DoF 및 3DoF+, Windowed 6DoF, 6DoF 등으로 구분되는 Immersive Video 신호의 표현 및 부호화에 대한 표준화를 진행하고 있다^[6].

360 VR 콘텐츠를 제작 및 압축 그리고 전송 및 소비하는 전 과정에서, 첫 단계는 각 sub 카메라가 촬영한 영상들을 스티칭하여 360 VR 신호를 구성하는 단계로써, 이 부분의 성능이 나쁘면, 이후 과정인 부호화 과정, 재생 및 소비 과정 등의 성능 역시 나빠지게 된다. 따라서, 본 논문에서는 첫 단계인 이미지 스티칭 과정을 올바르게 구현함으로써 360 VR 서비스의 품질을 향상시키려는 것을 목적으로 삼고 있다.

360 VR 영상의 스티칭 과정에서, 각 세부 영상들의 밝기가 너무 밝거나 어두운 경우, 움직이는 물체가 존재하는 경

우, 각 영상들 간에 시차(view disparity)가 존재하는 경우 등에서는 합성된 영상의 품질이 저하되고, 이는 360 VR 서비스의 완성도를 떨어뜨리는 결과로 연결된다. 이와 같이 스티칭을 어렵게 하는 문제들을 해결하기 위해서, W. Li, C. B. Jin, M. Liu, H. Kim, X. Cul^[7]은 한 영상에서 여러 가지 2D 변환 중 지역마다 가장 적절한 2D 변환을 수행하는 방법을 제시하였다. 비슷한 연구로써, N. Li, Y. Xu 과 C. Wang^[8]은 한 영상에 하나의 호모그래피만 계산하는 것이 아니라, 영상 내에서 지역적으로 각각의 호모그래피를 계산하여 가장 적절한 호모그래피 변환을 수행하도록 설계했다. K. Y. Lee and J. Y. Sim^[9]은 VR 동영상 내의 움직이는 물체들을 여러 시점에서 관찰하여 물체를 구성하는 각 픽셀이 적응적으로 2D 변환되도록 변환 모델을 설계했다. 이렇게 스티칭을 어렵게 하는 문제점들 및 해결 방법들 중에서, 영상들 간의 시차(view disparity) 문제를 극복하는 문제는 다른 제약조건 문제들에 비해서 중요성이 높고 해결하기 어려운 면이 있다. 본 논문에서는 자연스러운 360 VR 영상을 합성하기 위해, 세부 영상들 간의 시차 왜곡을 교정하는 알고리즘을 제안하고자 한다.

본 논문에서는 영상 내에 존재하는 물체들 간의 시차(view disparity)로 인해 발생하는 스티칭 왜곡을 최소화하기 위해서, 복잡도가 낮은 전처리 과정을 제안하여 이음부 탐색(seam finding) 과정이 개선된 마스크를 제작하도록 하였다. 이렇게 만들어진 마스크를 사용함으로써 blending 과정에서 경계의 어긋남이나 고스트(ghost) 발생 현상을 피할 수 있었다.

본 논문의 구성은 다음과 같다. II 장에서는 360 VR 영상을 스티칭하는 과정에서 시차에 의해 발생하는 문제점을 알아본다. III 장에서는 영상 내 물체들의 깊이 정보가 스티칭 과정과 어떤 관계가 있는지를 설명하였다. IV장에서는 제안하는 이음부 탐색(seam finding) 알고리즘에 대해 자세히 설명한다. V절에서는 기존 알고리즘들의 장단점을 간단히 소개하고, 기존 방법과 제안하는 방법들을 사용하여 스티칭된 영상들을 비교 분석하며, 이러한 실험 과정에서 측정된 연산 시간 및 최대 소모 메모리 크기를 비교한다. VI절에서는 논문의 내용을 요약하고 앞으로의 연구의 적용 방향을 제시한다.

II. 360 VR 영상 스티칭 과정의 문제점 분석

1. 시차(view disparity) 발생으로 인한 스티칭 에러

360 VR 영상을 스티칭하는 과정에서 발생하는 문제들 중 하나는 동일한 물체의 다른 길보기 위치 차이가 존재하는 문제이다. 이 문제는 영상에서 물체의 공간적인 위치를 정확하게 측정하기 어렵게 한다. 영상 정보에서 정지된 물체의 길보기 위치가 변하는 것을 시차(view disparity)가 발생한다고 한다. 시차가 얼마나 발생했는가는 카메라와 물체사이의 거리에 따라 달라진다. 그림 1에서 예시로 설명하듯이, 카메라에서 가까운 물체는 촬영된 영상마다 다른 위치에 있는 것처럼 보인다. 카메라에서 멀리 존재하는 물체일수록 길보기 위치차이는 작아진다. 이는 시차가 발생한 이미지마다 측정되는 물체의 평행 이동량이 다르기 때문이다. 이로 인해 스티칭 과정에서 사용되는 영상들간의 매핑식이 카메라에서 멀리 있는 물체와 카메라에 가까이 있는 물체에 따라 달라진다. 이러한 문제로 인해, 그림 1 (d)에서

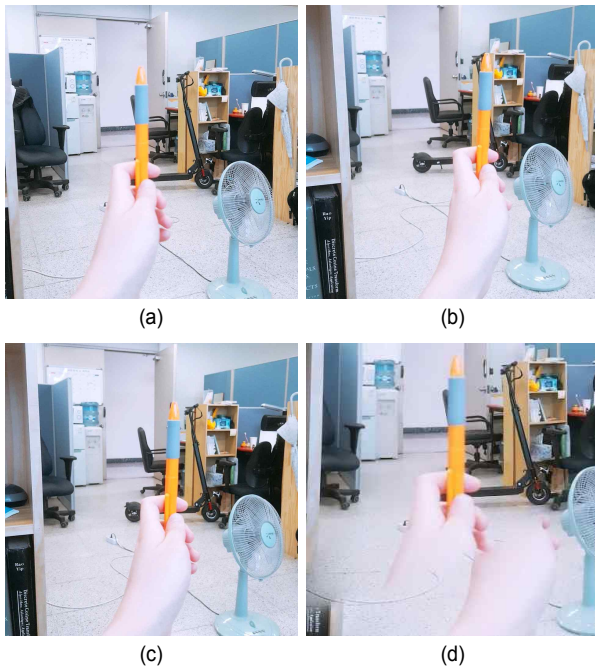


그림 1. (a),(b),(c): 다양한 각도로 촬영된 영상들에서 시차가 존재하는 물체들의 예, (d): 시차로 인한 스티칭 에러의 발생 예

Fig. 1. (a),(b),(c): Examples of objects captured with various directions, (d): Degradation resulting from view disparity in the stitched image

볼 수 있듯이, 스티칭된 이미지에서 물체가 잘리거나 고스팅이 일어나는 현상이 발생한다.

2. 기존의 스티칭 알고리즘

본 절에서는 스티칭의 전체 과정을 간단히 살펴보고 시차가 발생할 때, 이 문제가 알고리즘에서 어떤 영향을 끼치는지를 설명하고, 이 문제를 극복하기 위해서 어떤 추가과정이 필요한지 분석하고자 한다. 그림 2에서 표현한 스티칭

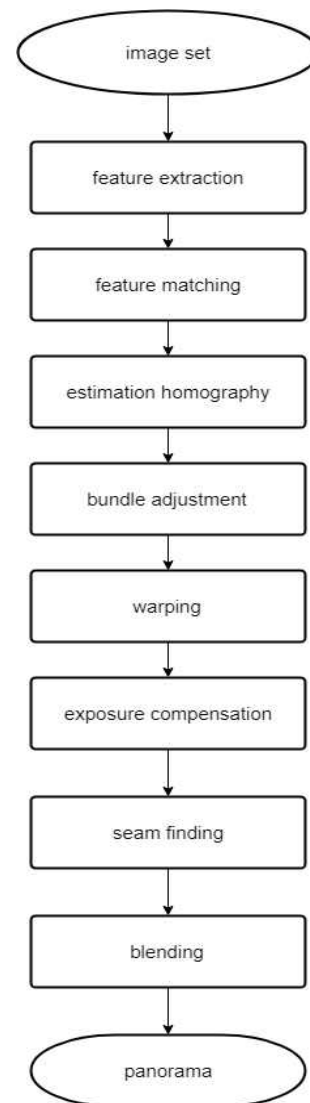


그림 2. 스티칭 알고리즘

Fig. 2. Conventional stitching algorithm

알고리즘의 첫 단계에서는, 영상별로 특징점을 추출하고, 그 다음 단계에서 이웃하는 이미지들 내의 특징점들을 서로 매칭시킨다^[10]. 이 과정에서 매칭된 특징점들 간의 관계를 가장 잘 표현하는 변환 방정식을 유도하는데, 이를 **estimation of homography**라고 한다. 이 과정을 통해 얻어진 **matrix** 방정식을 이용하여, 이웃하는 영상들을 스티칭한다. 이렇게 변환하여 합성된 영상을 만드는 과정을 **warping**이라고 한다. **warping**이 수행된 후, 전체 이미지의 밝기를 비슷하게 만드는 노출보정 과정(**exposure compensation**)^[11]을 거친다. 이렇게 합성된 영상에는 서로 다른 영상들을 합성하면서 발생한 경계의 어색함이 존재하는데, 주로 영상들의 오버랩 영역에서 발생한다. 이러한 어색함 내지 왜곡을 제거하기 위해서, 이러한 오버랩 영역에서 핵심이 되는 중요한 물체의 경계를 의미하는 **seam**을 찾고, 이를 이용하여 **blending**과정을 수행한다^[12]. 이렇게 물체의 경계를 찾는 과정을 이음부 탐색(**seam finding**)이라고 한다. 이음부 탐색(**seam finding**)은 픽셀의 에너지를 계산하여 고주파 영역과 저주파 영역을 분리한다. 고주파 영역에는 사람이나 건물 등의 주요한 물체가 분포하고 저주파 영역에는 하늘이나 벽과 같은 배경이 분포한다. 오버랩 영역에서 가장 저주파인 화소 경로를 따라 물체의 경계선이 최대한 보존되도록 경계를 결정하게 된다^[13].

앞 절에서 그림 1을 이용해서 설명한 영상들간의 시차(**view disparity**)는 **homography matrix**를 계산하는 과정에도 성능저하를 유발하고, 그 이후 과정인 **warping** 및 **blending**까지 성능 저하를 발생시킨다. 본 논문에서는 이웃하는 영상들간의 시차로 인한 왜곡 현상을 개선시키기 위해, 영상 내부 물체들의 깊이(**depth**) 정보를 이용하여 이음부 탐색(**seam finding**)을 수행한다. 시차가 발생한 경우 이미지 내 물체의 위치가 다르기 때문에 동일한 물체가 겹치지 않거나, 물체가 확대되어 저주파영역으로 인식하여 물체가 경계에서 잘리는 등 기존의 이음부 탐색(**seam finding**)이 효과적으로 작동하지 않는다. 따라서 시차가 큰 영역(**depth**가 낮은 영역)에 대해서는 이음부 탐색(**seam finding**)단계 이전의 특별한 신호처리가 필요하다. 이를 위한 제한된 알고리즘은 다음 장에서 자세히 설명한다.

III. 깊이 정보와 스티칭 과정의 관계

앞 절에서 설명한 시차(**view disparity**) 문제를 카메라로부터 거리가 가까운 (낮은 **depth** 정보를 가진) 물체와 배경 (깊은 **depth** 값을 가진 화소들)을 분리하여 처리하는 기술로 해결하고자 한다. 이를 위해서는, 먼저 화소들의 깊이 정보를 추출한 후, 이를 이용하여 중요한 물체가 왜곡되지 않도록 **seam finder**가 동작하도록 알고리즘을 제안한다. 이 이음부 탐색(**seam finding**) 과정에서 **overlap** 영역 내의 물체의 깊이값을 고려해 최종적으로 **seam**을 결정하도록 한다. 이를 위해서 본 장에서는 일반적인 스테레오 카메라의 깊이 추출방법을 스티칭 과정에 적용하는 이론적 배경을 설명하고자 한다.

일반적인 스테레오 카메라로 촬영된 영상에서는 삼각측량법을 이용하여 카메라에서 물체까지의 거리를 계산한다. 이를 위한 기본적인 구도를 그림 3과 식 (1)로 표현할 수 있다. 식 (1)에서 **B**는 렌즈 **o**와 렌즈 **o'**사이 거리, **f**는 카메라 초점거리이다. **Z**는 카메라에서 물체까지의 거리이며, **x, x'**는 각 카메라 렌즈 **o**와 **o'**에서 촬영된 영상에서의 동일한 물체의 좌표이다. **disparity**는 동일한 물체 **x**의 겹보기 위치 차이이다. 물체가 카메라에 가까울수록 **disparity**가 물체가 카메라에서 멀어질수록 **disparity**가 작아진다.

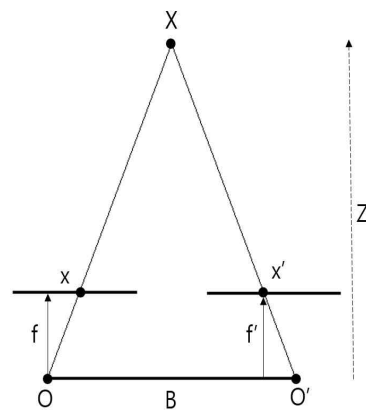


그림 3. 삼각측량법
 Fig. 3. Triangulation method

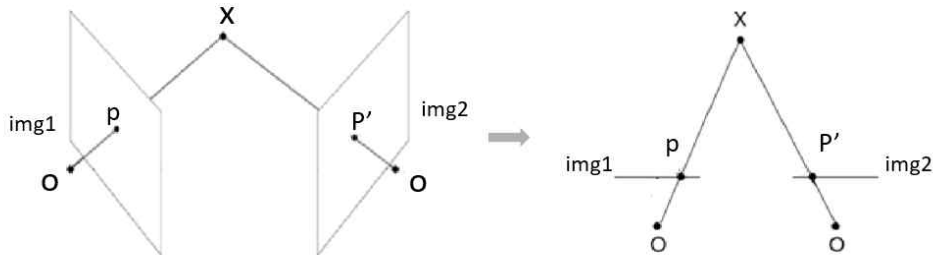


그림 4. 이미지 매칭쌍에 대한 삼각측량법 적용
Fig. 4. Triangulation method for the matched feature point pair

$$disparity = x - x' = \frac{Bf}{Z} \quad (1)$$

본 연구에서는 그림 2에서 설명한 feature extraction 과정을 통해 이웃한 영상들 내에 존재하는 동일 물체 x의 위치를 각 영상내에서 찾을 수 있고, 이렇게 매칭된 특징점 (feature point)들의 쌍을 그림 3과 수식 (1)에 적용시켜 깊이 정보를 추출하게 된다. 이러한 과정을 정합되는 이웃 영상들간의 관계 및 호모그래피 매트릭스를 이용한 관계와 연관시켜 그림 4와 식 (2)로 표현할 수 있다.

$$disparity = p - p' = p - Hp \quad (2)$$

식 (2)에서 p는 img1에서 물체x의 좌표, p'는 img2에서 물체x의 좌표이고 H는 img1과 img2 사이의 호모그래피 매트릭스를 의미한다. O는 카메라 렌즈이고 disparity는 동일한 물체 x의 겹보기 위치 차이이다. disparity가 크면 시차가 검출된 영역이라고 볼 수 있다. 이 과정을 통해 시차가 검출된 영역에서는 스티칭된 영상내에서 해당 물체들에 왜곡이 심하게 발생한다. 이 문제를 극복하기 위해, 본 논문에서는 시차가 검출된 물체는 이미지 매칭 쌍 중 어느 한 쪽의 데이터만 사용하는 기법을 제안한다.

IV. 제안된 이음부 탐색(seam finding) 알고리즘

본 장에서는 앞 장들에서 설명한 문제점을 극복하기 위해 제안된 이음부 탐색(seam finding) 알고리즘을 설명한다. 본 장에서 제안하는 알고리즘은 앞서 그림 2에서 설명

한 기존의 스티칭 알고리즘의 단계들 중에서 ‘이음부 탐색 (seam finding)’ 과정 이전에 추가적으로 수행되는 과정이다.

1. ROI 설정

앞서 설명한 그림 2의 기존 스티칭 알고리즘에서, 밝기 보상이 끝난 이웃 영상들로부터 이미지 매칭쌍들이 만들어지면, 이 영상들 간의 오버랩 영역을 계산하고, 이를 ROI (Region of Interest)로 설정한다. 이렇게 설정된 ROI 영역이 이웃하는 원 영상들의 어느 영역에 해당하는지를 계산하여, 그 영역 부분을 잘라낸 후, 저장한다. 이렇게 저장된 영역 영상들을 subimage1과 subimage2로 명명한다. 그림 5는 이웃하는 영상 두 장이 stitching되는 관계와 이로 인한 ROI 영역, 그리고 이 ROI가 원래 이웃 영상의 어느 영역에 해당하는지를 계산하고, 이에 따라 서브 이미지들을 추출하는 과정을 설명하고 있다.

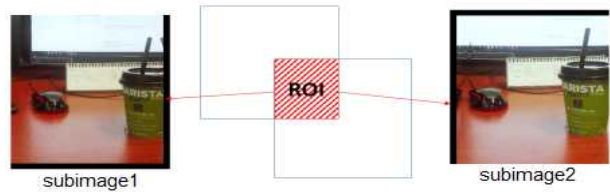


그림 5. ROI 설정과 서브이미지 추출
Fig. 5. Extraction of subimages according to ROI

2. 서브 이미지들 간의 밝기값 차이 계산

그림 5에서 설명한 subimage1과 subimage2는 그림 2의 스티칭 알고리즘을 통해, 특징점 쌍을 추출하고, 이 특징점

쌍들을 이용하여 두 영상들간의 **warping** 관계를 의미하는 호모그래피 매트릭스를 계산하고, 이에 따라 **warping**을 수행한 영상들이다. 그런데, 이 영상들 내의 화소값들 간에 차이가 존재한다면, 이는 III장에서 설명한 것처럼 해당 물체들의 깊이 정보가 호모그래피 매트릭스를 계산하는 과정에 반영되지 못했기 때문이다. 이 절에서는 ROI 영역과 관련된 **subimage1**과 **subimage2** 사이의 화소값들의 밝기값 (**Gray**) 차이를 비교함으로써, 이 ROI 영역을 **warping**하는 과정에 깊이 정보가 충분히 반영되었는지 여부를 측정하게 된다. 이때, **subimage1**과 **subimage2** 사이의 칼라 화소값들을 비교하지 않고, 칼라 화소값을 **gray** 밝기값(**Y**)으로 변환시킨 후, 이 밝기값들 간의 차이를 계산하는데, 이렇게 하는 이유는 그림 2의 **exposure compensation** 과정에서 밝기값을 보정시켰기 때문에 밝기 값들 간의 차이를 계산하는 것이 정확한 왜곡을 측정할 수 있기 때문이다.

이 과정에서 밝기값들의 차이의 절대값이 임계값 이상이면 시차(**view disparity**)가 존재한다고 판단한다. ROI 내의 화소들간의 밝기값 차이들의 평균값을 $E(disparity)$ 라고 표현하면, 임계값은 식 (3)과 같이 나타낼 수 있다. 이 식에서 **disparity**는 앞의 식 (2)에서 설명한 것처럼 **warping**된 **subimage**들 간의 각 화소별 차이 밝기값을 의미한다. $E()$ 는 두 개의 **subimage**들 전체에 대해서 화소별 **disparity** 값의 평균값을 의미한다. 이 식에서 알 수 있듯이, 시차가 많이 날수록 임계값이 작아지도록 설정하였다. 이런 과정을 거쳐 시차가 크게 존재한다고 판단된 ROI에서는 개선된 마스크를 제작한 후, 이 마스크를 이용하여 **seam finder**와 **blending**을 수행한다. 그림 6은 서브 이미지들의 밝기값과 이 밝기값들 간의 차이신호를 예제로 보이고 있다.

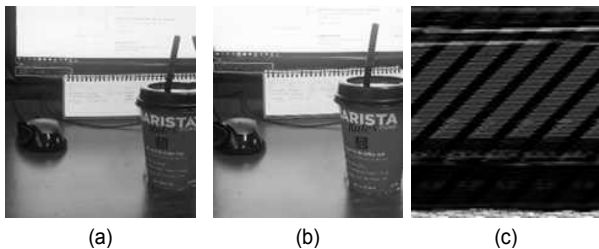


그림 6. (a) subimage1의 밝기값, (b) subimage2의 밝기값, (c) subimage1과 subimage2의 밝기값의 차이영상
 Fig. 6. Difference image between gray pixels in subimage1 and subimage2

$$threshold = \frac{255}{8 * E(disparity)} \quad (3)$$

3. 서브 이미지들과 원 영상들 간의 연속성 측정

앞의 과정을 통해 **subimage**들 간의 밝기값 차이가 크다고 판단되면, 이는 **view disparity**가 크기 때문에 발생한 현상이며, 이런 상황에서 스티칭된 영상의 왜곡을 줄이기 위해서는 두 **subimage**들 중에서 특정 서브 이미지를 선택해서 사용해야한다.

이때, 사용할 서브 이미지를 결정하기 위해 **subimage1**이 원래 **image2**와 얼마나 자연스럽게 스티칭되는지와 **subimage2**가 원래 **image1**과 얼마나 자연스럽게 스티칭되는지를 평가한다. 이를 위해서는 **subimage**들의 경계값과 원래 **image**들의 경계선에 위치한 화소값들 간의 차이를 비교하게 되는데, 정확한 측정을 위해 컬러 화소값들 간의 차이값을 계산한다. 그림 7은 **subimage1**이 원래 **image2**와 얼마나 자연스럽게 스티칭되는지를 측정하는 과정을 설명하고

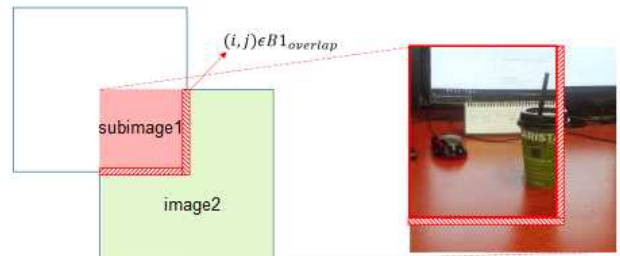


그림 7. subimage1과 image2간의 연속성 평가
 Fig. 7. Continuity Evaluation between boundaries of subimage1 and image2

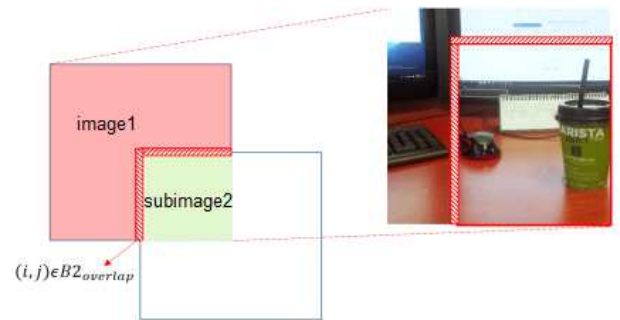


그림 8. subimage2과 image1간의 연속성 평가
 Fig. 8. Continuity Evaluation between boundaries of subimage2 and image1

$$d1 = \sum_{c \in R, G, B} \sum_{(i,j) \in B1} (|subimg1(i-1,j)^c - img2(i+1,j)^c| + |subimg1(i,j-1)^c - img2(i,j+1)^c|) \tag{4}$$

$$d2 = \sum_{c \in R, G, B} \sum_{(i,j) \in B2} (|subimg2(i-1,j)^c - img1(i+1,j)^c| + |subimg2(i,j-1)^c - img1(i,j+1)^c|) \tag{5}$$

있으며, 이때 부조화 정도를 측정하기 위해서 식 (4)를 이용한다. 또한 그림 8은 subimage2이 원래 image1과 얼마나 자연스럽게 스티칭되는 지를 측정하는 과정을 설명하고 있으며, 이때 부조화 정도를 측정하기 위해서 식 (5)를 이용한다.

위 과정을 통해 계산된 d1과 d2의 값을 비교한 후, 더 작은 d값을 가지고 있는 서브이미지를 해당 ROI를 구성할 데이터로 선택한다.

4. 마스크의 재설정

앞 절에서 설명한 과정을 수행한 후, 그 결과에 따라 image1과 image2를 blending하는 과정에서 사용되는 마스크

를 조정한다. 그 예를 그림 9에서 보여주고 있다. 그림 9의 사진은 ROI 영역으로 사용될 데이터로 subimage2가 선택된 예를 보여주고 있다. image1과 image2를 blending하는 과정에서 image2 내부의 subimage2가 ROI를 구성하게 되기 때문에, 시차가 검출된 픽셀에 해당하는 image1의 마스크의 값은 0의 값을 갖도록 세팅된다. 이 때 삭제할 픽셀에 대체 이미지로 결정된 image2의 데이터가 존재할 때만 image1의 마스크의 값을 0으로 바꾼다. 따라서 image1의 마스크 값을 0으로 바꿨을 때 두 마스크의 값이 모두 0이 되면 image1의 마스크 값을 바꾸지 않고 1로 유지한다. 이와 같은 방법으로 아래 그림9의 (b)가 시차 검출 픽셀에 해당하는 마스크의 값이 조정된 image1의 마스크이다. 그림9의 (c)는 image2의 마스크로, image2가 시차 검출 영역의 값이 1로 보존된 것을 볼 수 있다.

이렇게 조정된 마스크를 이용하여 이음부 탐색(seam finding)을 수행한다. 이음부 탐색(seam finding)을 수행하는 방법은 기존의 이음부 탐색(seam finding) 알고리즘을 그대로 이용한다. 그림 10은 이음부 탐색(seam finding) 알고리즘을 통해서 얻어지는 blending용 마스크들의 예를 보여주고 있다. 그림 10 (a)는 그림 2와 같은 기존의 알고리즘에서 사용하는 마스크들이고, (b)는 (a)의 마스크들을 이용하여 이음부 탐색(seam finding)을 수행한 결과 얻어진 blending용 마스크들을 나타내고 있다. 본 논문에서 제안하는 마스크 제작 기법으로 얻어진 마스크들을 그림 10 (c)에서 보여주고 있고, 이를 이용하여 이음부 탐색(seam finding)을 수행해서 얻어진 blending 용 마스크를 (d)에서 표시하고 있다. 그림 10 (c)에서 표시한 마스크에서처럼, 제안된 알고리즘에 의해 disparity가 큰 픽셀이 지워지고 남은 영역이 너무 자잘하게 분포하면 블렌딩 이후 픽셀이 깨진 것처럼 보이고 이미지 매칭쌍의 두 마스크의 값이 모두 사용된 경우 고스팅을 일으킬 수 있다. 따라서 (c)마스크 셋으로 이음부 탐색(seam finding)을 수행하여 너무 작은 영역은 병합하고 양쪽 마스크의 데이터가 모두 사용되는 영역이 발생하지 않도록 한다. seam finder에 제안된 알고리즘을 거

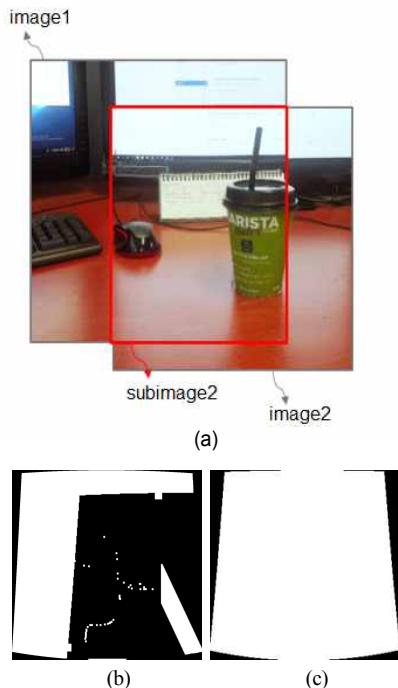


그림 9. (a) ROI 영역에서 subimage2가 선택된 경우, (b) image1을 위한 시차 검출 영역 픽셀 값이 지워진 마스크, (c) image2를 위한 시차 검출 영역 픽셀값이 보존된 마스크

Fig. 9. (a) Example of ROI constructed from image1 and image2, (b) Mask for image1, (c) Mask for image2

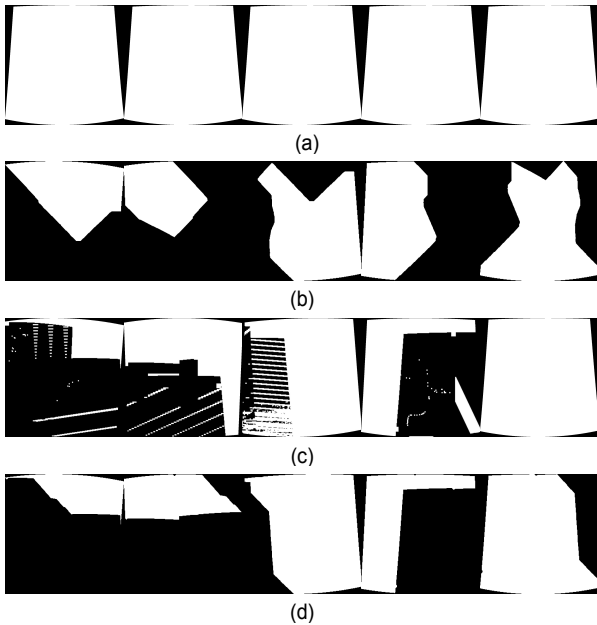


그림 10. (a) 기존 방법에서 사용되는 마스크 set, (b) (a)를 이용하여 이음부 탐색(seam finding)을 수행한 결과 마스크 set, (c) 제안하는 기술로 만든 마스크 set, (d) (c)를 이용하여 이음부 탐색(seam finding)을 수행한 결과 마스크 set

Fig. 10. (a) Conventional mask set, (b) Mask set resulting from seam finder using mask set of (a), (c) Mask set generated from the proposed algorithm, (d) Mask set resulting from seam finder using mask set of (c)

친 마스크 셋을 전달하면, seam finder에서 두 마스크의 값이 모두 1인 픽셀이 존재하는 충돌 영역에서 seam을 결정한다. 충돌영역의 픽셀은 각 마스크의 충돌하지 않는 픽셀까지의 거리를 구해 가까운 쪽의 마스크에 합쳐진다. 이와 같은 과정을 통해 마스크의 병합 및 분리가 완료된다. 시차

가 보상된 최종 마스크가 그림 10의 (d)마스크 셋이다.

그림 10에서 표시한 다양한 마스크 set들을 사용해서 영상들을 합성한 결과를 그림 11에서 나타내고 있다. 그림11 (a)는 그림10 (b)의 기존의 마스크 set을 이용해서 합성된 결과 영상이고, 그림11 (b)는 제안된 알고리즘이 적용된 그림 10 (d)의 마스크 set을 이용하여 합성한 결과 영상이다. 두 영상은 합성된 영상에서 이미지의 경계가 어떻게 다른지 보이기 위해 의도적으로 블렌딩을 수행하지 않고 제시하였다. 그림 11 (c)와 (d)는 그림 11 (a)와 (b)를 만드는 과정에 blending을 추가한 결과 영상이다. 이때 합성된 영상의 품질을 자세히 살펴보기 위해서 특정 영역을 확대해서 표시하고 있다.

5. 제안된 알고리즘의 요약 및 정리

제안된 아이디어의 알고리즘을 요약하자면 다음과 같다. 밝기 보상이 완료된 이미지 set의 오버랩 영역을 관심영역으로 설정한다. 두 관심영역을 추출한 영상을 gray scale로 변환한 뒤 각 픽셀의 밝기 값의 차를 이용해서 disparity를 계산한다. 이렇게 계산된 disparity가 문턱값보다 클 때, 제안하는 알고리즘이 사용되고, 문턱치보다 작을 때는 기존 방법을 사용하여 스티칭이 수행된다. 제안하는 알고리즘을 사용하는 경우, 이미지1의 관심영역이 이미지2의 관심영역 밖의 부분과 어울리는지 아니면 이미지2의 관심영역이 이미지1의 관심영역 밖의 부분과 어울리는지 오버랩 경계에서 RGB 변화량을 통해 판단한다. 예를 들어 RGB 변화량

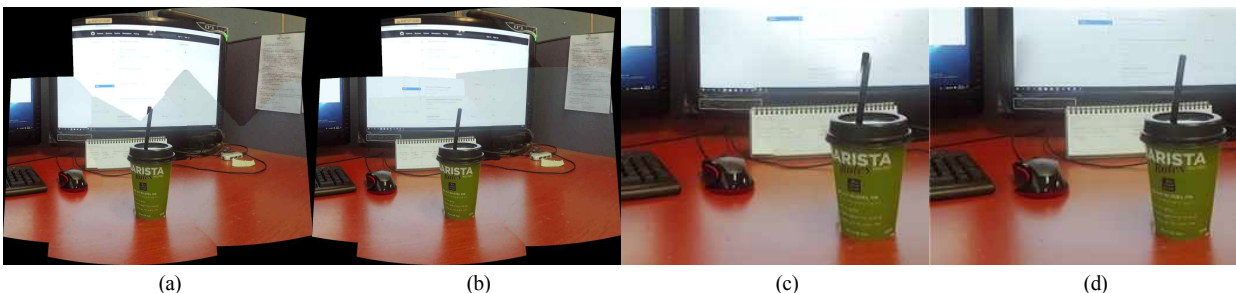


그림 11. (a) Blending 과정 없이 그림 10 (b)의 마스크를 이용하여 스티칭된 영상, (b) Blending 과정 없이 그림 10 (d)의 마스크를 이용하여 스티칭된 영상, (c) Blending 과정을 적용하면서 그림 10 (b)의 마스크를 이용하여 스티칭된 영상, (d) Blending 과정을 적용하면서 그림 10 (d)의 마스크를 이용하여 스티칭된 영상

Fig. 11. (a) The stitched image based on the mask of Fig. 10 (b) without blending, (b) The stitched image based on the mask of Fig. 10 (d) without blending, (c) The stitched image based on the mask of Fig. 10 (b) with blending, (d) The stitched image based on the mask of Fig. 10 (d) with blending

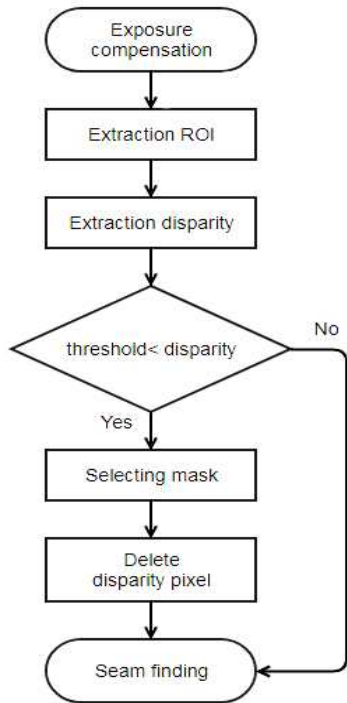


그림 12. 제안된 아이디어의 알고리즘
Fig. 12. Proposed algorithm

이 작은 경우가 이미지1의 관심영역과 이미지2의 관심영역 밖 부분을 사용한 경우라면, 이미지2의 마스크에서 disparity가 제안된 threshold보다 큰 픽셀을 삭제한다. disparity가 threshold 이상인 픽셀은 삭제되고 threshold 이하인 픽셀은 마스크에서 삭제되지 않은 마스크가 전달되어 기존 이음부 탐색(seam finding) 방법이 수행된다. 이러한 과정을 그림 12에서 도식적으로 설명하고 있다.

V. 실험 결과

본 장에서는 다양한 알고리즘들을 이용하여 스티칭을 실행했을 때, 스티칭된 결과 영상들을 비교하고, 연산 시간과 최대 소모 메모리 크기를 비교한다. 본 논문에서 제안하는 알고리즘의 성능을 기존 방법들의 성능들과 비교하기 위해서 다음과 같은 두 개의 기존 알고리즘들을 구현하여 성능을 분석하였다.

본 실험에서 고려된 기존 알고리즘들 중에 하나인 ‘보로노이(Voronoi) 그래프 컷 알고리즘’^[14]은 특징점들을 seed로 clustering하여 보로노이(Voronoi) 다이어그램을 구성한다. 보로노이 다이어그램은 주요물체와 배경을 구분하며, 이 영역 간 경계가 seam이 되어 주요물체를 보존하는 방법이다. 소요되는 메모리나 연산시간에 비해 주요물체를 잘 보존하지만 ‘최대 유량(Max-Flow) 그래프 컷 알고리즘’에 비해 오버랩 경계에서는 물체가 잘리는 단점이 있다. 추가로 고려된 기존 알고리즘인 ‘최대 유량 그래프 컷 알고리즘’^[15]에서는 각 픽셀을 노드로 그래프를 구성한다. 픽셀의 에너지를 간선의 유량으로 고려하며, 주요 물체를 구성하는 노드에서 배경을 구성하는 노드까지 최대 유량을 흘려 보내면 배경을 구성하는 노드까지 도달하지 못하고 간선들을 채우게 된다. 그 간선들은 최소 손실 비용으로 고려되며, 최소 비용 간선을 삭제함으로써 주요 물체와 배경을 구분한다. 물체를 온전하게 보존하는 장점이 있지만, 이미지 셋의 모든 이미지마다 이미지의 화소로 트리를 구성하고 최대 유량을 계산하게 되므로 메모리와 연산 시간이 많이 소모된다. ‘보로노이 그래프 컷 알고리즘’^[14]은 연산 시간과 소모 메모리는 적은 편이지만 물체의 어긋남이 심한 편이고 ‘최대 유량 그래프 컷 알고리즘’^[15]은 물체는 잘 보존되지만 연산 시간과 소모 메모리 크기가 매우 많다. 또한 ‘보로노이 그래프 컷 알고리즘’보다 ‘최대 유량 그래프 컷 알고리즘’이 적용될 수 있는 이미지 세트의 종류가 더 적다. 즉 ‘보로노이 그래프 컷 알고리즘’이 적용된 결과 영상을 확인할 수 있는 경우에도 ‘최대 유량 그래프 컷 알고리즘’이 적용된 결과영상을 확인할 수 없는 경우가 있다. 본 논문에서는 비교하려는 모든 알고리즘들을 적용할 수 있는 실험 영상들에 대해서만 실험을 실시하였다. 본 연구의 실험에서는 (a) ‘보로노이 그래프 컷 알고리즘’만 적용하는 실험, (b) ‘최대 유량 그래프 컷 알고리즘’만 적용하는 실험, (c) 제안하는 마스크 제작 알고리즘과 ‘보로노이 그래프 컷 알고리즘’을 모두 적용하는 실험 등을 실시하였다.

본 실험에서 사용된 테스트 영상들은 삼성 갤럭시 S6의 Samsung SM-G920K 카메라로 빛이 충분한 실내에서 촬영되었다. 초점거리는 4.30mm이고 노출 시간은 1/20초로 통일되어 있다. 그림 13의 영상은 1080x1080 ‘연구실 풍경’ jpg 영상 3장을 스티칭한 결과영상이다. 그림 14는 5312x

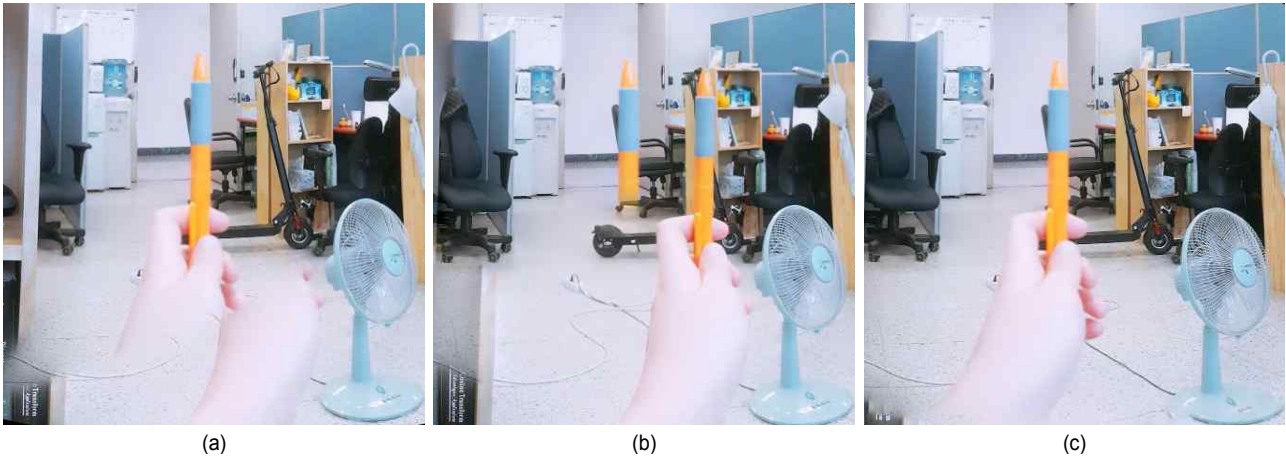


그림 13. '연구실 풍경' 영상에 대해 기존 기술들과 제안 기술을 적용하여 스티칭된 이미지들, (a):보로노이 그래프 컷 알고리즘, (b):최대 유량 그래프 컷 알고리즘, (c):제안된 아이디어와 보로노이 그래프 컷 알고리즘
 Fig. 13. 'Research Laboratory' Images resulting from various stitching algorithms, (a) Voronoi Cut Algorithm, (b) Max-flow Graph Cut Algorithm, (c) The proposed algorithm combined with Voronoi Graph Cut Algorithm



그림 14. '복도 풍경' 영상에 대해 기존 기술들과 제안 기술을 적용하여 스티칭된 이미지들, (a):보로노이 그래프 컷 알고리즘, (b):최대 유량 그래프 컷 알고리즘, (c):제안된 아이디어와 보로노이 그래프 컷 알고리즘
 Fig. 14. 'Aisle' Images resulting from various stitching algorithms, (a) Voronoi Graph Cut Algorithm, (b) Max-flow Graph Cut Algorithm, (c) The proposed algorithm combined with Voronoi Graph Cut Algorithm

2988 '복도 풍경' jpg 영상 71장을, 그림 15은 5312x2988 '의자' jpg 영상을 스티칭한 결과 영상의 일부이며, 그림 16은 '거실' jpg 영상들을 스티칭한 결과 영상들이다.

그림 13에서 볼 수 있듯이, 기존 알고리즘을 사용한 그림 13의 (a)에서는 펜과 손이 잘린 것이 보이고, (b)에서는 펜이 이중으로 나타나고 있다. 제안된 알고리즘이 적용된 영상 (c)에서는 물체와 손이 온전히 보존되었음을 볼 수 있다.

그림 14에서 볼 수 있듯이, 기존 알고리즘을 적용한 그림

14의 (a)와 (b)에서는 기둥의 윗 부분이 어긋나고 그림의 왼쪽 벽면의 포스터와 문이 어긋난 것을 확인할 수 있다. 제안된 알고리즘이 적용된 (c)에서는 기둥과 문, 포스터 등의 물체가 온전하게 보존됨을 확인할 수 있다.

그림 15에서는 제안된 알고리즘이 적용된 영상 (c)에서는 보로노이 그래프 컷 알고리즘이 적용된 (a) 영상과 최대 유량 그래프 컷 알고리즘이 적용된 영상 (b)와 달리 물체 전체가 나타나는 것을 알 수 있다. 그렇지만, 제안하는 방법



그림 15. '의자' 영상에 대해 기존 기술들과 제안 기술을 적용하여 스티칭된 이미지들, (a):보로노이 그래프 컷 알고리즘, (b):최대 유량 그래프 컷 알고리즘, (c):제안된 아이디어와 보로노이 그래프 컷 알고리즘
 Fig. 15. 'Chair' Images resulting from various stitching algorithms, (a) Voronoi Graph Cut Algorithm, (b) Max-flow Graph Cut Algorithm, (c) The proposed algorithm combined with Voronoi Graph Cut Algorithm



그림 16. '거실' 영상에 대해 기존 기술들과 제안 기술을 적용하여 스티칭된 이미지들, (a):보로노이 그래프 컷 알고리즘, (b):최대 유량 그래프 컷 알고리즘, (c):제안된 아이디어와 보로노이 그래프 컷 알고리즘
 Fig. 16. 'living room' Images resulting from various stitching algorithms, (a) Voronoi Graph Cut Algorithm, (b) Max-flow Graph Cut Algorithm, (c) The proposed algorithm combined with Voronoi Graph Cut Algorithm

을 사용한 (c) 결과에서도 기존 기술을 적용한 결과 영상 (a)와 마찬가지로 경계에서 책상이 끊어지는 등 여전히 개선해야할 점이 존재하는 것을 알 수 있었다. 이점은 후속 연구를 통해 개선해야할 점이라고 생각한다.

앞의 실험 결과 영상들 중에서, 그림 15는 영상 내에 고주파 성분이 많기 때문에, 저주파 성분이 많은 영상들인 그림 13과 그림 14의 결과가 그림 15보다 화질개선 효과가 분명함을 확인할 수 있었다. 그림 15 내의 의자 부분에서 왜곡이 심하게 발생했는데, 그 이유는 다음과 같다. 그림 15에서 두 영상의 오버랩 영역의 경계에서 고주파 성분이

많으면 두 RGB 변화량이 비슷한 값을 가져 subimage들 중 에서 어떤 영상이 스티칭에 더 적절한지 판단하기 어려운 상태가 되어 왜곡이 심하게 발생하였다. 같은 이유로 그림 13와 같이 시차가 발생하는 물체가 가까울 때(depth가 낮을 때)가 물체가 멀 때(depth가 깊을 때)보다 오버랩 영역의 경계에서 고주파 성분이 적어져 시차검출 영역의 대체 이미지로 올바른 subimage가 선택되었다. 따라서 제안된 아이디어가 적용된 알고리즘의 기능이 효과적으로 기능했음을 알 수 있었다.

그림 16은 시차가 적게 발생한 이미지 셋에 대한 실험

결과 영상이다. (a)와 (b)영상은 기존 알고리즘들이 적용된 결과이고 (c)는 제안된 아이디어가 적용된 결과 영상이다. 시차가 적은 경우 제안된 아이디어에 의해 삭제되는 픽셀이 적기 때문에 제안된 아이디어를 적용하지 않고 기존 그래프 컷 알고리즘이 적용된 영상과 같은 결과영상을 얻는다.

그러나 제안 방법에서는 두 ROI 영상의 밝기 값의 차이로만 시차왜곡 여부를 결정하기 때문에 충분한 밝기 보상이 이루어져야 한다. 밝기 보상이 충분하지 않거나 잘못 수행된 경우 그림 16의 실험 이미지 set과 같이 실제 영상에서 disparity가 없어도 시차가 발생했다고 판단되어 물체의 굽김, 고스팅 등의 부작용이 발생할 수 있다. 이와 같은 이유로 그림 16의 (c)에서도 (a)와 달리 오디오 부분에서 고스팅이 일어난 부분이 발생함을 볼 수 있다.

표 1, 표 2, 표 3은 그림 13, 그림 14, 그림 15의 결과를 얻기 위한 실험에서 소모된 연산 시간과 최대 메모리 크기를 비교한 것이다. 스티칭하려는 사진 개수가 적을 때는 세 알고리즘 모두 비슷한 연산시간과 메모리 양을 소모했지만, 사진의 개수가 증가할수록 ‘최대 유량 그래프 컷 알고리즘’은 다른 알고리즘들 보다 2배 이상의 메모리를 사용했고,

표 1. 그림 13의 알고리즘 별 연산 시간과 메모리 소비량 비교
 Table 1. The time and memory consumed to stitch images by various algorithms for images of Figure 13

	Voronoi Graph Cut Algorithm	The proposed algorithm combined with Voronoi Graph Cut Algorithm	Max-flow Graph Cut Algorithm
Consumed Time (sec)	0.015	0.039	0.269
Consumed Memory (MB)	50.3	51.8	56.9

표 2. 그림 14의 알고리즘 별 연산 시간과 메모리 소비량 비교
 Table 2. The time and memory consumed to stitch images by various algorithms for images of Figure 14

	Voronoi Graph Cut Algorithm	The proposed algorithm combined with Voronoi Graph Cut Algorithm	Max-flow Graph Cut Algorithm
Consumed Time (sec)	2.171	6.295	264.756
Consumed Memory (MB)	202.0	204.0	532.1

표 3. 그림 15의 알고리즘 별 연산 시간과 메모리 소비량 비교
 Table 3. The time and memory consumed to stitch images by various algorithms for images of Figure 15

	Voronoi Graph Cut Algorithm	The proposed algorithm combined with Voronoi Graph Cut Algorithm	Max-flow Graph Cut Algorithm
Consumed Time (sec)	2.246	6.836	236.670
Consumed Memory (MB)	195.8	205.6	538.0

훨씬 긴 연산 시간을 소모했다. 제안된 기술과 ‘보로노이 그래프 컷 알고리즘’을 적용했을 경우, 복잡도나 메모리 소모량이 ‘보로노이 그래프 컷 알고리즘’만을 사용했을 경우와 거의 비슷했다.

그림 13, 그림 14, 그림 15, 그리고 표 1 ~ 3의 결과를 근거로 판단할 때, 연산시간이나 메모리를 적게 소모하면서, 시차 (view disparity)에 강인한 스티칭 알고리즘은 제안된 기술과 ‘보로노이 그래프 컷 알고리즘’을 결합한 방법이었다.

VI. 결 론

본 논문에서는 360 VR의 중요성과 연구동향에 대해 살펴보고 VR 영상을 구성할 때 발생하는 문제점들을 알아보았다. 360 VR 영상의 품질을 저하 시키는 요인 중 하나인 시차가 스티칭의 과정인 이음부 탐색(seam finding)에서 어떻게 에러를 발생시키는지 분석하고, 이를 해소하기 위해 필요한 처리 과정의 방향을 제시하였다. 시차로 인한 스티칭 에러의 개선 방법으로 시차가 크게 발생하는 영역에서는 개선된 마스크를 제작하는 방법을 소개하였다. 기존 알고리즘과 제안한 기술들이 적용된 실험결과 영상들, 그리고 연산시간과 최대 소모 메모리 양을 비교한 표들을 제시하여 스티칭 에러가 개선됨을 보였다.

참 고 문 헌 (References)

[1] Chang-Hoon Kang, "Flow of next generation broadcast video, Present and Future of VR contents," The Korea Contents Association Review,

- Vol. 14, No. 2, pp. 14~18, 2016, June.
- [2] J. G. Andrews et al., "What Will 5G Be?," in IEEE Journal on Selected Areas in Communications, vol. 32, no. 6, pp. 1065-1082, June, 2014.
- [3] ITU-T/ISO/IEC JVET, "Results of the Joint Call for Evidence on Video Compression with Capability beyond HEVC", JVET-G1004-v2, July, 2017.
- [4] ITU-T/ISO/IEC JVET, "Algorithm descriptions of projection format conversion and video quality metrics in 360Lib", JVET-E1003, January, 2017.
- [5] ITU-T/ISO/IEC JVET, "Common Test Conditions and Evaluation Procedures for HDR/WCG Video Coding", JVET-D1020, October, 2016.
- [6] M. Domański, O. Stankiewicz, K. Wegner and T. Grajek, "Immersive visual media – MPEG-I: 360 video, virtual navigation and beyond," 2017 International Conference on Systems, Signals and Image Processing (IWSSIP), Poznan, pp. 1-9, May, 2017.
- [7] W. Li, C. B. Jin, M. Liu, H. Kim and X. Cui, "Local similarity refinement of shape-preserved warping for parallax-tolerant image stitching," in IET Image Processing, vol. 12, no. 5, pp. 661-668, May, 2018.
- [8] N. Li, Y. Xu and C. Wang, "Quasi-Homography Warps in Image Stitching," in IEEE Transactions on Multimedia, vol. 20, no. 6, pp. 1365-1375, June 2018.
- [9] K. Y. Lee and J. Y. Sim, "Stitching for Multi-View Videos With Large Parallax Based on Adaptive Pixel Warping," in IEEE Access, vol. 6, pp. 26904-26917, 2018.
- [10] M. Brown, D. G. Lowe, "Automatic Panoramic Image Stitching using Invariant Features", International Journal of Computer Vision, Volume 74 Issue 1, Pages 59 - 73, August 2007.
- [11] M. Uyttendaele, A. Eden and R. Szeliski, "Eliminating ghosting and exposure artifacts in image mosaics," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, pp. II-509-II-516 vol.2, December, 2001.
- [12] Vladan Rankov, Rosalind J. Locke, Richard J. Edens, Paul R. Barber, Borivoj Vojnovic, "An algorithm for image stitching and blending", SPIE 5701, Three-Dimensional and Multidimensional Microscopy: Image Acquisition and Processing XII, March 2005.
- [13] Muthukrishnan.R, M.Radha, " Edge detection techniques for image segmentation", International Journal of Computer Science & Information Technology (IJCSIT) Vol 3, No 6, Pages 259 - 267, December, 2011.
- [14] Chetan Arora, Subhashis Banerjee, Prem Kalra, and S. Maheshwari. An efficient graph cut algorithm for computer vision problems. In Computer Vision ECCV 2010, volume 6313 of Lecture Notes in Computer Science, pages 552-565. Springer Berlin / Heidelberg, 2010.
- [15] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 9, pp. 1124-1137, Sept. 2004.

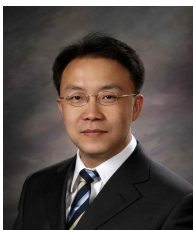
— 저 자 소 개 —

손 희 정



- 2016년 ~ 현재 : 세종대학교 전자정보통신공학과 재학 중
- ORCID : <https://orcid.org/0000-0002-6346-0712>
- 주관심분야 : VR, 자동차 비전, 드론 통신

한 종 기



- 1992년 : KAIST 전기및전자공학과 공학사
- 1994년 : KAIST 전기및전자공학과 공학석사
- 1999년 : KAIST 전기및전자공학과 공학박사
- 1999년 3월 ~ 2001년 8월 : 삼성전자 DM연구소 책임연구원
- 2001년 9월 ~ 현재 : 세종대학교 전자정보통신공학과 교수
- 2008년 9월 ~ 2009년 8월 : University California San Diego (UCSD) Visiting Scholar
- ORCID : <https://orcid.org/0000-0002-5036-7199>
- 주관심분야 : 비디오 코덱, 영상 신호처리, 정보 압축, 방송 시스템