

일반논문 (Regular Paper)

방송공학회논문지 제23권 제6호, 2018년 11월 (JBE Vol. 23, No. 6, November 2018)

<https://doi.org/10.5909/JBE.2018.23.6.896>

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

# 360 VR을 구성하는 영상들 간 밝기 차이를 이용한 seam finding 알고리즘

남 다 윤<sup>a)</sup>, 한 종 기<sup>a)‡</sup>

## Seam Finding Algorithm using the Brightness Difference Between Pictures in 360 VR

Da-yoon Nam<sup>a)</sup> and Jong-Ki Han<sup>a)‡</sup>

### 요 약

360 VR 영상을 구성하는 과정에서 합성되는 영상들 사이의 경계(seam)를 정확하고 효율적으로 추출하는 것은 스티칭된 영상의 화질에 매우 중요한 영향을 끼친다. 기존의 seam finding 기술들인 보로노이(Voronoi) 알고리즘, dynamic programming, graph cut 방법들은 시차(view disparity)가 존재하거나 이동하는 물체가 존재하는 상황에서는 ghost와 같은 왜곡을 발생시킨다. 본 논문에서는 기존 알고리즘들과 달리 물체를 가로지르지 않고, 물체를 피해 배경을 가로질러 접합선을 추출하는 ‘밝기 대비를 이용한 마스크 추출 방법’을 제안하였다. 이때, 배경과 물체와의 밝기차이를 이용하여, 물체의 형상을 따라 이어지는 접합선을 추출했다. 본 논문의 실험 결과를 통해, 제안하는 기술을 사용하면 왜곡을 감소시키는 위치에서 접합선이 생성되고, 스티칭된 영상의 화질이 개선되는 것을 확인하였다.

### Abstract

Seam finding algorithm is one of the most important techniques to construct the high quality 360 VR image. We found that some degradations, such as ghost effect, are generated when the conventional seam finding algorithms (for examples, Voronoi algorithm, Dynamic Programming algorithm, Graph Cut algorithm) are applied, because those make the inefficient masks which cross the body of main objects. In this paper, we proposed an advanced seam finding algorithm providing the efficient masks which go through background region, instead of the body of objects. Simulation results show that the proposed algorithm outperforms the conventional techniques in the viewpoint of the quality of the stitched image.

Keywords: Seam finder, 360 VR, Image Stitching, Panorama Image

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

‡ Corresponding Author : 한종기(Jong-Ki Han)

E-mail: [hjk@sejong.edu](mailto: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 September 12, 2018; Revised October 29, 2018; Accepted October 29, 2018.

Copyright © 2016 Korean Institute of Broadcast and Media Engineers. All rights reserved.

“This is an Open-Access article distributed under the terms of the Creative Commons BY-NC-ND (<http://creativecommons.org/licenses/by-nc-nd/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited and not altered.”

## 1. 서론

이미지 스티칭(Image Stitching)<sup>[1,2]</sup> 기술은 하나의 영상으로 담기지 않는 풍경을 중첩되도록 촬영한 여러 장의 사진을 이어 붙여 파노라마 영상으로 만드는 기술을 의미한다. 현재 가장 주목받는 기술 중 하나인 360°VR 역시 스티칭 과정이 기본적으로 필요하다.

스티칭 과정의 기본적인 구성은 다음과 같다. 사진의 크기, 회전, 노이즈, 밝기에 불변하는 특징을 가진 ‘특징점(Feature)’들을 찾아내는 ‘특징점 검출(Feature Extraction)’ 과정, 영상들 간 닮은 특징점끼리 매칭하는 ‘특징점 매칭(Feature Matching)’ 과정, 매칭된 특징점들을 동일한 지점에 위치시켜 공통의 중첩된 영상을 가지고 있는 인접한 사진들끼리 이어지도록 하는 ‘영상(Warping)’ 과정, 영상의 경계선을 흐릿하게 만드는 ‘블렌딩(Blending)’ 순서로 실행된다. 스티칭 과정에서 블렌딩까지 거치면, 결과 사진이 완성된다. 또한 스티칭 과정 중 와핑까지만 완료해도, 하나로 이어진 영상을 얻을 수 있다. 하지만, 이 경우에는 인접한 영상과 영상 사이에 밝기 차, 동적 피사체, 시차 등의 여러 원인들 때문에 영상이 이어지더라도, 모자이크 효과처럼 영상과 영상과의 구분이 명확하게 드러나는 왜곡이 발생한다. 그렇기 때문에 스티칭의 마지막 과정으로 블렌딩 기술을 적용해서 여러 장의 사진이 자연스럽게 이어지도록 하는 접합선(Seam)을 기준으로 두 영상을 부드럽게 연결될 수 있도록 한다. 블렌딩의 방법들에는 대표적으로 주파수별로 feathering 하는 피라미드 블렌딩(Pyramid Blending)<sup>[3]</sup>, 포아송 방정식을 이용한 포아송 블렌딩(Poisson Blending)<sup>[4]</sup> 등이 있다. 모든 블렌딩 알고리즘은 접합선을 기준으로 영상을 부드럽게 접합시키는 것이다. 블렌딩 과정 안에 접합선을 찾는 과정이 포함된 알고리즘도 있지만, 주로 사용되는 피라미드 블렌딩의 경우 접합선을 찾는 과정이 포함되어 있지 않다. 따라서 피라미드 블렌딩의 경우, 미리 추출한 접합선 정보를 추가적으로 입력해주어야 한다. 그리고, 이렇게 두 장의 영상을 이어 붙이는 접합선의 정보를 가지고 있는 영상이 마스크이다. 마스크에 의해 최종 파노라마 영상이 크게 달라질 수 있다. 이러한 블렌딩 과정 전에 마스크를 추출하는 기능을 seam finder에서 수행한다. 만약 성능이 좋지 않은 seam finder를 사용하여 스티칭 영

상을 만든다면, 물체의 끊어짐과 어긋남이 발생하여 와핑의 결점들이 부각될 수 있고, 오버랩 영역 안에서 영상이 겹쳐져서 보이는 고스팅(Ghosting)<sup>[5]</sup>이 발생할 수도 있다. 고성능의 seam finder 일수록 접합선 중심으로 영상 간 불일치가 적고, 고스팅이 보이지 않을 것이다. 기존에 seam finder 알고리즘으로는 Voronoi segment를 이용하는 보로노이(Voronoi)<sup>[6]</sup>, 저주파 성분을 찾는 다이나믹 프로그래밍(Dynamic Programming)<sup>[7]</sup>과 그래프 컷(Graph Cut)<sup>[8]</sup>, 오버랩되는 영상간 최소 오차를 찾는 최소오차경계블렌딩(Minimal Error Boundary Cut)<sup>[9,10]</sup>, 최소오차경계블렌딩[9]에서 가중치를 적용한 가중치 블렌딩 기법<sup>[10]</sup> 등이 있다. 하지만 기존의 seam finder들이 추출하는 접합선은 영상 속 물체를 관통하는 경우가 있다. 이 경우, 영상 안의 물체를 가로지르는 선을 기준으로 영상이 정합되기 때문에, 이로 인해 스티칭된 영상 내의 물체의 어긋남이나 끊어짐이 발생할 수 있다. 본 논문은 눈에 띄는 물체를 관통함을 통해 생기는 어긋남을 줄이기 위한 ‘밝기 대비 마스크 추출’ 알고리즘을 제안한다. 배경과 물체와의 밝기 대비를 통해 물체의 형상을 판단하고, 물체를 피해 배경으로 접합선을 만드는 seam finder 기술을 제안한다.

본 논문의 II장에서는 일반적인 스티칭 과정과 블렌딩에서 마스크가 필요한 이유를 설명하고, III장에서는 기존의 seam finder 알고리즘들을 설명하고 비교 및 분석한다. IV장에서는 제안하는 seam finder의 마스크 추출 과정을 세부적으로 설명한다. V장에서는 기존의 알고리즘들과 제안 알고리즘의 결과 영상들을 비교하여 성능 분석하고, VI장에서 결론을 맺는다.

## II. 스티칭 과정 및 마스크의 기능

### 1. 이미지 스티칭 과정

이미지 스티칭(Image Stitching)은 중첩되는 영역을 가진 여러 장의 사진을 오버랩 영역(Overlap Roi)에서 이어 붙여 하나의 영상을 만드는 작업이다. 스티칭은 간단히 (a) 특징점 검출(feature extraction), (b) 특징점 매칭(feature matching), (c) 영상 와핑(warping), (d) 경계선 블렌딩(blending)

이렇게 4단계로 구분된다. 스티칭의 첫 단계에서는 SIFT (Scale Invariant Feature Transform)<sup>[11,12]</sup>, SURF(Speeded Up Robust Features)<sup>[13]</sup>, ORB(Oriented BRIEF)<sup>[14]</sup>, FAST<sup>[15]</sup> 등의 특징점 검출 알고리즘으로 특징점을 추출(feature extraction)한다. 영상의 크기, 회전, 밝기 등에 불변하는 영상 고유의 특징을 가진 점을 특징점이라 한다. 일반적으로 코너점을 특징점으로 결정한다. 그리고 검출된 특징점과 주변 픽셀들의 관계를 각도와 같은 정보로 표현하는 기술자(Descriptor)를 특징점 마다 저장한다. 기술자의 비교를 통해, 현재 영상의 특징점과 가장 닮은 중첩하는 영상의 특징점을 찾아, 일대일로 매칭(Feature Matching)시킨다. 다음 일반적으로 RANSAC<sup>[16]</sup>을 이용해 가장 많은 내외점(Inlier)을 가진 매칭 모델을 선택하고, 이 모델을 벗어나는 이상치(Outlier)를 제거한다. 매칭 모델을 통한 이미지 간의 매칭쌍의 관계를 호모그래피 행렬로 저장한다. 이 호모그래피 행렬을 통해 영상마다 회전행렬과 초점거리를 구한다(estimator)<sup>[17]</sup>. 여러 장의 영상이 하나의 사진으로 이어지도록 여러 장의 영상을 현재 이미지 좌표계상에서 구형, 실린더형, 평면형, 어안형 등의 좌표계 상으로 픽셀을 매핑시키고, 다시 평면 이미지로 투영시켜 저장한다(warping). 이때

와핑 과정에서 파노라마의 목적에 따라 구형 투영(Spherical Projection), 실린더 투영(Cylindrical Projection), 평면 투영(Plane Projection) 등 와핑 방법이 달라진다. 와핑을 하고나면, 여러 장의 사진이 한 장의 사진 안에 위치되지만, 포스트잇을 붙여놓은 듯 영상 간 영상의 차이가 왜곡처럼 보이는 파노라마 영상이 만들어 진다. 이러한 영상 간의 오차가 적은 접합선을 결정(seam finder)하고, 접합선을 중심으로 현재 영상에서 인접한 영상으로 자연스럽게 전환시켜 주는 블렌딩(blending)을 한다. 이와 같은 일반적인 스티칭 과정을 그림 1에서 도식적으로 설명하고 있다.

## 2. 블렌딩 과정에서 사용되는 마스크

마스크(mask)는 0 또는 1, 두 가지 정보만 가지는 영상이다. 마스크는 현재 영상과 인접 영상을 블렌딩할 때, 현재 영상의 차단시킬 영역과 통과시킬 영역을 결정하는 이진영상이다. 차단시킬 영역과 통과시킬 영역이 구분되는 경계선은 인접한 영상과 이어지는 접합선이 된다. 즉, 이미지가 합성되는 접합선(seam)의 정보가 마스크에 담겨 있다. 그림 3의 이진화 영상이 그림 2 와핑 영상의 마스크이다. 그림 3

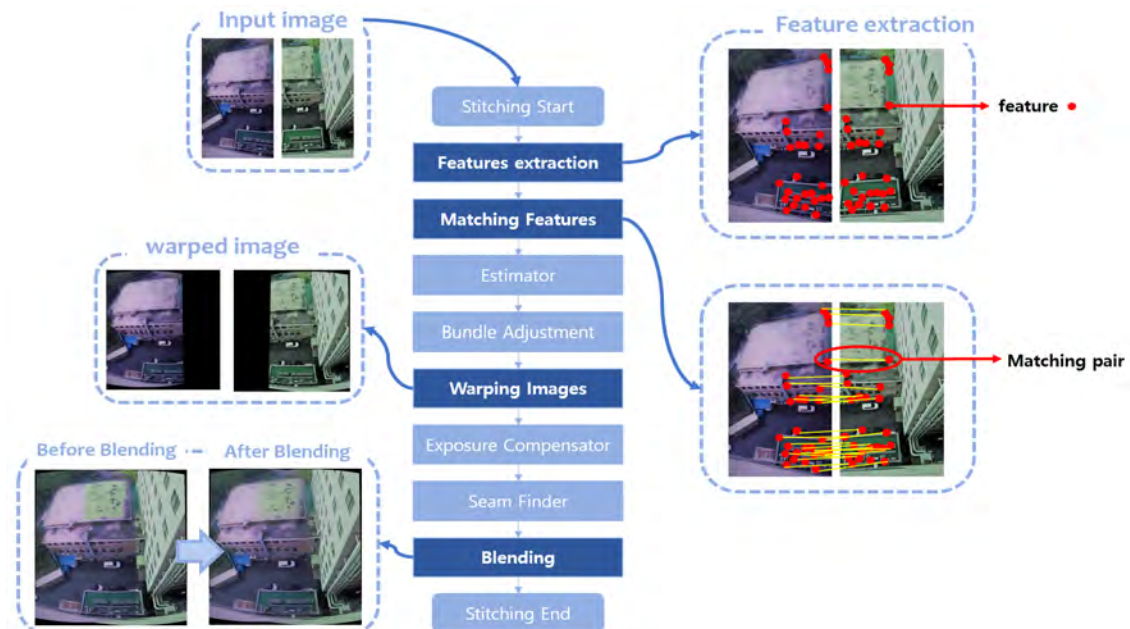
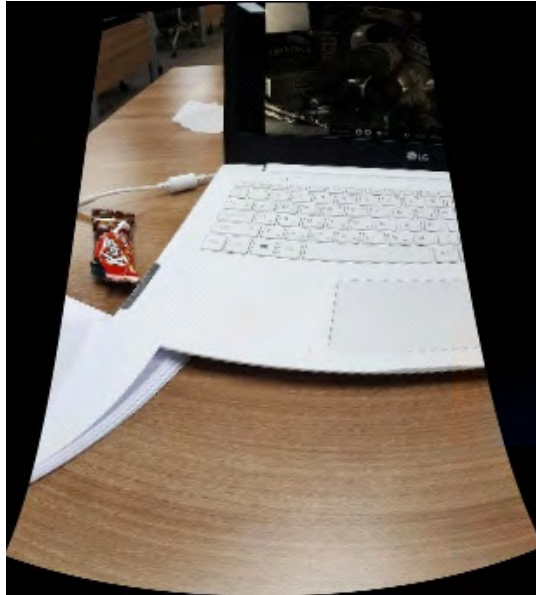


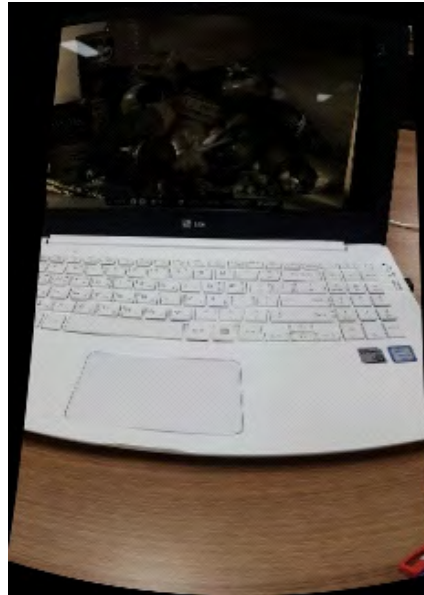
그림 1. 복수개 영상들의 합성 과정  
Fig. 1. Flowchart of the image stitching algorithm

의 하얀색 영역에는 1이 저장되어 있고, 이는 그림 2의 영상에서 해당 영상의 화소값을 통과시키는 영역이다. 반면,

그림 3의 검은색 영역에는 0이 저장되어 있고 그림 2의 영상에서 해당 영역을 차단시키는 역할을 한다. 그리고, 마

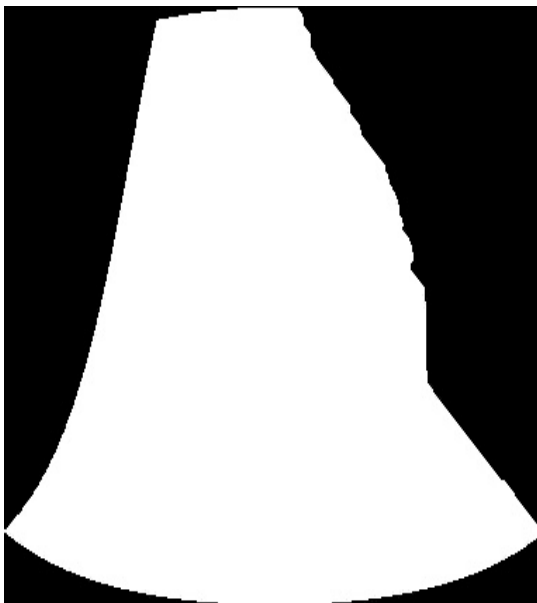


(a)



(b)

그림 2. 스티칭되기 전에 Warping된 사진들  
Fig. 2. Warped images for the stitching process



(a)

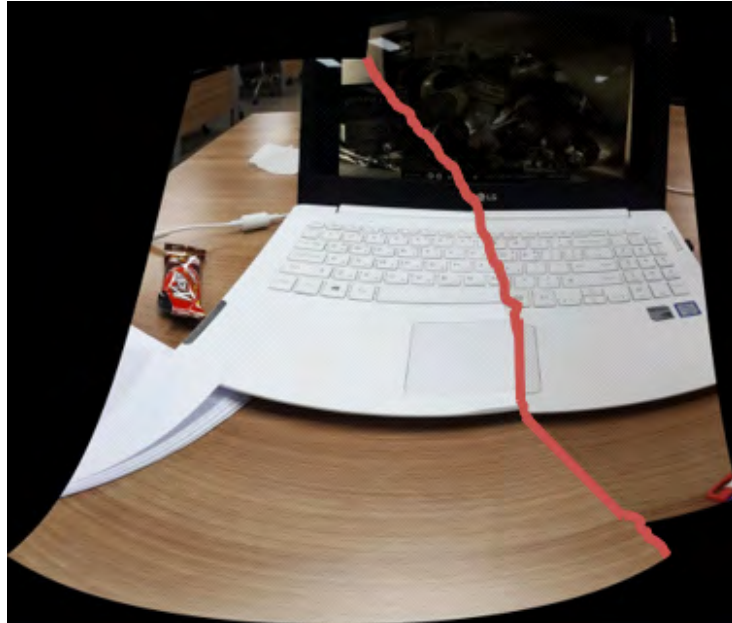


(b)

그림 3. Warping된 사진들을 stitching하기 위한 Mask들  
Fig. 3. Masks for stitching the warped pictures

스크의 하얀색 영역에서 검은색 영역으로 바뀌는 경계선은, 그림 2의 (a) 영상과 (b) 영상을 이어주는 접합선 역할을 한다. 그림 3의 마스크를 그림 2에 적용해서 블렌딩을 하면,

그림 4 (a)과 같이 스티칭 영상을 얻게 된다. 이 그림에서 가운데를 관통하는 빨간색 선은 접합선(seam)을 나타낸다. 그림 4 (b)는 마스크 없이 블렌딩할 경우 얻어지는 결과영



(a) The image stitched by using masks of Fig.3



(b) The stitched image without mask

그림 4. 마스크 사용한 경우와 사용하지 않은 경우들의 스티칭된 이미지들 비교  
Fig. 4. Comparison between the stitched images using mask and no mask



상이다. 그림 4 (a) 영상과 그림 4 (b) 영상을 비교함으로써 마스크의 역할을 확인할 수 있다.

마스크는 스티칭 영상을 구성하는 과정에서 영상들을 합성하는 접합선을 결정한다. 만약 마스크가 없다면, 접합선에 대한 아무런 정보를 얻을 수 없게 된다. 이 경우에 왜곡된 영상을 그대로 블렌딩할 것이다. 즉, 모든 영상이 통과되어 블렌딩되어 스티칭 영상이 만들어 지기 때문에, 그림 4 (b)처럼 이미지가 다중으로 분산되어 보이는 고스팅(Ghosting)이 발생한다. 왜곡을 하더라도, 영상마다 동일한 피사체가 같은 위치에 놓여져 있지 않을 수 있기 때문에 마스크를 사용하지 않고 블렌딩하면, 제 각각 다른 위치에 존재하는 피사체로 인해 다중으로 흐릿하게 보이는 것이다. 이런 고스팅 현상을 없애기 위해서는 스티칭된 영상의 특정 영역을 구성할 때, 여러 장의 사진의 정보들이 모두 반영되면 곤란하고, 한 장의 사진만이 통과되고, 나머지 사진들은 모두 차단되어야 한다. 그리고 통과 영역과 차단 영역을 구별하고, 영상과 영상을 이어주는 것이 접합선(seam)이다. 이런 접합선의 정보를 담은 마스크를 생성하는 알고리즘이 seam finder이다.

### III. 기존 seam finder 알고리즘들 분석

기존에 알려진 seam finder 알고리즘은 3가지가 있다. 보

로노이(Voronoi)<sup>[6]</sup>, 다이내믹 프로그래밍(Dynamic Programming)<sup>[7]</sup>, 그래프 컷(Graph Cut)<sup>[8]</sup> 있다. 이 중 보로노이 경계선 추출은 영상 정보와 상관 없이 오직 왜곡된 이미지의 모양으로 경계선을 결정하는 방식이고, 다이내믹 프로그래밍과 그래프 컷은 영상 내부의 화소값들의 변화가 급격한 영역을 피해서 저주파 속성을 가진 경계선 찾는 방식이다. 이 중 ‘보로노이’와 다이내믹 프로그래밍의 일반적인 과정을 살펴보고, 기존 seam finder 알고리즘의 기능을 확인하고자 한다.

#### 1. Voronoi Seam Finder

한 영상에서 오버랩 영역을 모두 잘라내고 남은 영역은 어떠한 영상과도 겹치지 않고 현재 영상에서만 볼 수 있는 고유한 영역이다. Voronoi seam finder는 이 영역에 최대한 가까이 위치한 경계선을 생성하는 알고리즘이다. 이 알고리즘을 그림 5에서 도식적으로 표현하고 있다.

가장 먼저 현재 영상 안에서 다른 영상이 위치할 수 있는 영역에 해당하는 마스크 내 픽셀값들을 모두 0으로 저장하고, 그 외의 모든 픽셀값들에는 1을 저장한다. 이 과정을 모든 영상들을 위한 마스크들에 적용시킨다면 영상마다 마스크로 사용될 이진화 영상을 얻을 수 있다.

그 다음, 이진화 영상을 거리변환하는데, 거리변환이란 현재의 픽셀과 그 주변의 0의 값을 갖는 가장 가까운 픽셀과의 거리를 현재 픽셀 위치에 저장시키는 것이다. 거리변환

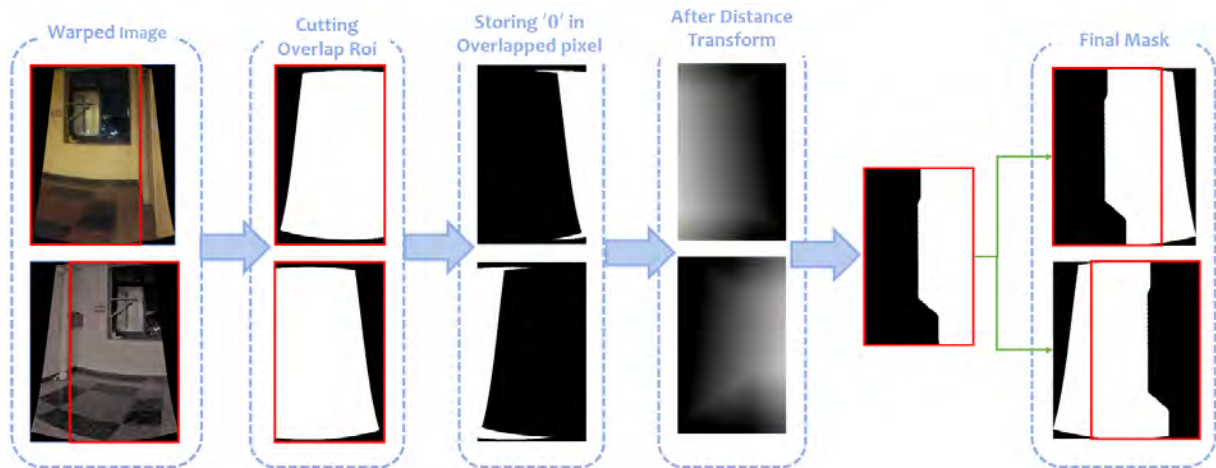


그림 5. 보로노이 알고리즘의 과정  
 Fig. 5. Process of Voronoi algorithm

영상을 얻으면, 현재 영상의 거리변환 영상을 오버랩되는 영상의 거리변환 영상과 같은 위치의 픽셀끼리 비교한다. 현재 영상의 거리변환 영상이 더 작은 값을 가지면 현재 영상의 마스크에 0을 저장하고, 더 큰 값을 가지면 원래의 현재 마스크 값을 유지시킨다. 즉, 현재의 픽셀이 현재영상의 고유영역에 가까운지, 다른 영상의 고유영역에 가까운지를 비교해서 제일 가까운 영상의 마스크에만 같은 위치의 픽셀에 1의 값을 갖게 하고, 그 외 나머지 모든 영상의 마스크는 같은 위치의 픽셀에 0의 값을 갖게 하는 것이다.

Voronoi는 오버랩 영역 안에 겹치는 영상이 많을 때, 비교적 깔끔한 경계선을 추출할 수 있고, 간단한 알고리즘이기 때문에 실행 시간도 짧아 효율적이다. 하지만, 경계선이 이미지의 주파수, 밝기 등의 정보와 아무 관련 없이 경계선을 만들기 때문에, 경우에 따라 파노라마 영상 안에 물체의 어긋남이 심하게 나타날 수도 있다.

## 2. Dynamic programming

다이나믹 프로그래밍 알고리즘은 영상의 가장 저주파인 지점을 따라 경계선을 추출하는 알고리즘이다. 이 알고리

즘의 과정을 그림 6에서 도식적으로 설명하고 있다. 이 과정에서는 와핑된 영상을 흑백으로 변화시켜  $u$  열 및  $v$  행에 위치한 픽셀의 intensity 값인  $I(u,v)$ 를 이용하여, 수평인  $u$  축 방향과 수직인  $v$  축 방향으로의  $I(u,v)$ 의 변화율을 계산한다. 이를 통해,  $u$  열 및  $v$  행에 위치한 픽셀의 에너지 값  $e(u,v)$ 을 구할 수 있다. 모든 픽셀에 대해서 식(1)과 같이 에너지를 계산하고, 현재 픽셀 값이 저장된 위치에 에너지 값  $e(u,v)$ 을 저장한다.

$$e(u,v) = \left| \frac{\partial I(u,v)}{\partial u} \right| + \left| \frac{\partial I(u,v)}{\partial v} \right| \quad (1)$$

모든 픽셀에 대해 각각 에너지를 저장한 후, 위에서 아래 방향으로 이동하면서, 한 칸 위에 인접한 3개의 픽셀 중 가장 작은 에너지와 현재 위치 픽셀의 에너지를 더해, 누적 에너지 값인  $M(u,v)$ 을 계산한다. 현재 에너지 값  $e(u,v)$ 이 저장된 위치를 누적 에너지 값  $M(u,v)$ 으로 업데이트 시킨다.

그리고 영상의 마지막 행에 도달할 때까지 이 과정을 모든 픽셀에 대해 반복한다.

$$M(u,v) = e(u,v) + \min(M(u-1,v-1), M(u-1,v), M(u-1,v+1)) \quad (2)$$

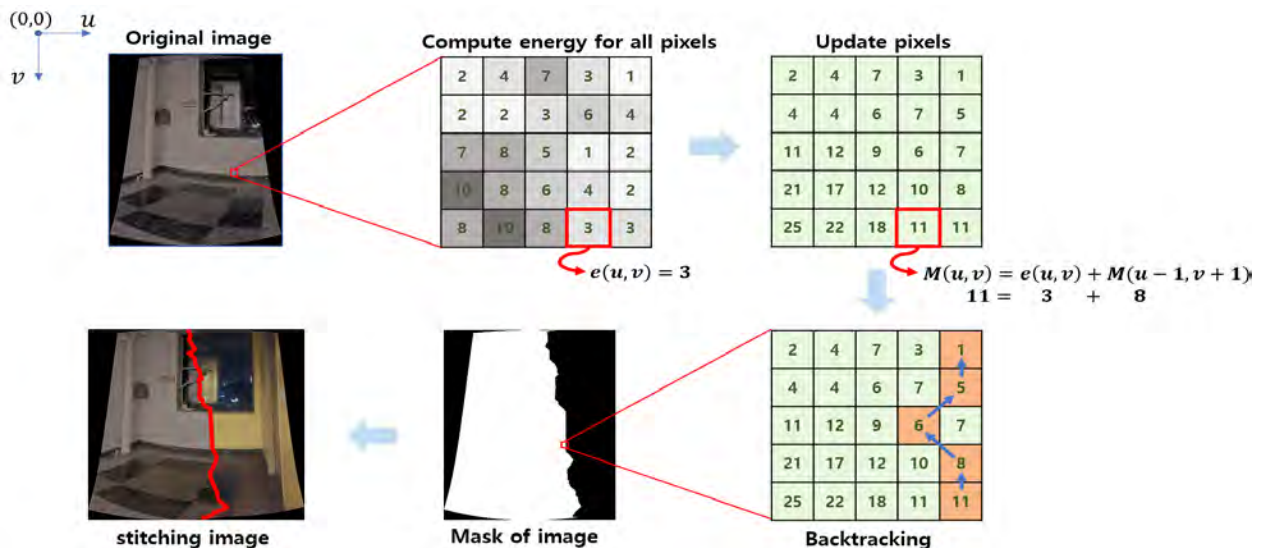


그림 6. 다이나믹 프로그래밍 알고리즘의 과정  
Fig. 6. Process of Dynamic programming algorithm

모든 픽셀 위치에서 누적 에너지 값으로 업데이트 했으면, 마지막 행에서부터 바로 위의 인접한 세 개의 픽셀의 누적 에너지 값 중 가장 작은 값을 선택하면서 위로 올라간다. 이렇게 역추적하면서 첫번째 행에 도달하면, 하나의 수직 경로가 만들어 지는 것이다. 여러 수직 경로를 찾고, 경로의 에너지를 기준으로 정렬한다. 가장 에너지가 작은 경로를 경계선(seam)으로 삼아 최종 마스크를 만든다.

그 외에도 Dynamic programming과 비슷하게 에너지함수를 구해서 저주파의 경계선을 찾는 Graph cut 알고리즘이 있다. 그리고 오버랩되는 영상끼리 비교해서, 같은 위치의 픽셀에서 오차가 가장 적은 픽셀을 연결하면서 경계선을 만드는 최소오차블렌딩 기법<sup>[9]</sup>이 있다.

그림 7은 기존 seam finder 알고리즘들을 적용시켜 스티칭된 영상들이다. 그림 7에서는 접합선을 눈에 보이도록 확



(a) The stitched image by using Voronoi



(b) The stitched image by using Dynamic programming



(c) The stitched image by using graph cut

그림 7. 기존 seam finder 알고리즘들을 적용하여 스티칭된 사진들 (블렌딩 과정은 사용하지 않음)

Fig. 7. The stitched pictures by using the conventional seam finder algorithms, where the blending process is not applied to compare the performances of the seam finder algorithms



인하기 위해, 경계선을 흐릿하게 만드는 기능을 하는 블렌딩 과정을 적용하지 않고 스티칭한 결과 영상들이다. 그림 7의 가운데를 관통하는 빨간색 선은 접합선(seam)을 나타낸다. Seam finder을 통해, 물체의 어긋남, 엇갈림, 잘림 등이 눈에 보이는 경계선이 만들어 질 수도 있고, 자연스럽게 사진이 연결되는 경계선이 만들어 질 수도 있다. 고스탕이 적고, 눈에 띄지 않는 경계선을 만드는 것이 성능이 좋은

seam finder라고 할 수 있다. 본 논문에서 제안하는 seam finder 역시 경계선의 잘못된 위치로 물체의 엇갈림과, 사라짐이 발생하는 문제를 해결하기 위한 알고리즘이다.

### 3. 기존 스티칭 알고리즘들의 문제점

스티칭 프로그램은 입력 영상에 따라 스티칭 영상 출력



Parking Lot



Column in the building

그림 8. 시차에 의한 스티칭 에러  
Fig. 8. Stitching error resulted from Parallax



Motorcycle in the road



Walking people

그림 9. 피사체의 움직임에 의한 스티칭 에러  
Fig. 9. Stitching error resulted from moving objects

에 실패할 수도, 부자연스러운 영상이 출력될 수도 있다. 스티칭 에러의 원인은 여러 가지 존재한다. 첫째 원인은 원거리와 근거리의 물체들의 간의 시차가 다르기 때문에 다른 각도에서 같은 카메라로 같은 영상을 찍을 지라도 상대적인 위치 차이가 존재하는 현상이다. 이러한 현상을 그림 8에서 보여주고 있다. 두 번째 원인은 동적 피사체의 위치 이동이다. 카메라에 담겨지는 피사체가 동적이거나, 피사체의 형태가 수시로 바뀌는 경우, 물체의 일부가 잘리는 스티칭 영상을 얻기 쉽다. 또한 동적이지 않지만, 카메라 그림자, 카메라 삼각대, 카메라 플래시 등 카메라와 함께 움직이는 경우에도 실제와는 큰 차이가 보이는 파노라마 영상이 발생한다. 이러한 현상을 그림 9에서 보여주고 있다. 그 외 각종 요인들에 의해 스티칭으로 생성된 파노라마 영상 안의 물체는 어긋나거나, 이중으로 보이거나, 잘려질 수 있다. 특히 중요한 원인으로 카메라 노출의 차이가 다르다는 점이다. 카메라를 이동시킬 때, 렌즈로 들어오는 빛의 양이 모두 다르다. 두 영상의 밝기 차이가 클 경우 스티칭 영상에서 밝기가 급격히 변하는 결과가 나타날 것이다. 본 논문에서 제안하는 블렌딩 마스크 추출 방식은 앞서 설명한 첫번째 원인과 두 번째 원인으로 인한 문제점을 보완하기 위한 마스크 추출 방식이다.

#### IV. 제안하는 마스크 추출 및 seam finding 알고리즘

제안하는 마스크 추출방법의 기본아이디어는 블렌딩하려는 두 영상 내의 물체의 형상을 이용하여 만들어진 이진화 영상들을 기반으로 마스크를 만드는 것이다. 영상 간 중첩하는 물체를 관통하는 경계선이 아니라, 물체의 형상을 따라 물체 밖에서 그려진 집합선을 가진 마스크를 생성하는 것이다. 두 영상이 포개지는 영역 안에 동일한 물체가 유사한 밝기를 갖도록 조절하고, 이러한 영상의 밝기 대비를 이용해 배경과 명확하게 구분되는 물체 형상을 추출하고, 물체의 형상에 간격을 둔 집합선을 생성한다. 영상 간 시차나 움직임이 존재하더라도 잘리거나 어긋남이 적어지도록 물체의 전체 부분을 한 영상으로만 나타나게 하였다. 본 논문에서 제안하는 밝기 대비를 이용한 마스크 추출

알고리즘 및 seam finding 알고리즘을 flowchart로 표시하면 그림 10과 같다. 제안하는 seam finder는 연결되는 모든 영상의 평균 밝기를 비슷하게 맞추고, 연결되는 두 영상을 선택해서 오버랩 영역만 남기고 나머지는 모두 제거한다. 그리고 밝기 정보만을 남겨 흑백영상으로 전환하고, 임계값을 기준으로 이진화시킨다. 이런 이진화된 영상을 다듬어 최종 마스크로 옮긴다. 이것을 모든 연결된 영상 쌍에서 반복한다.

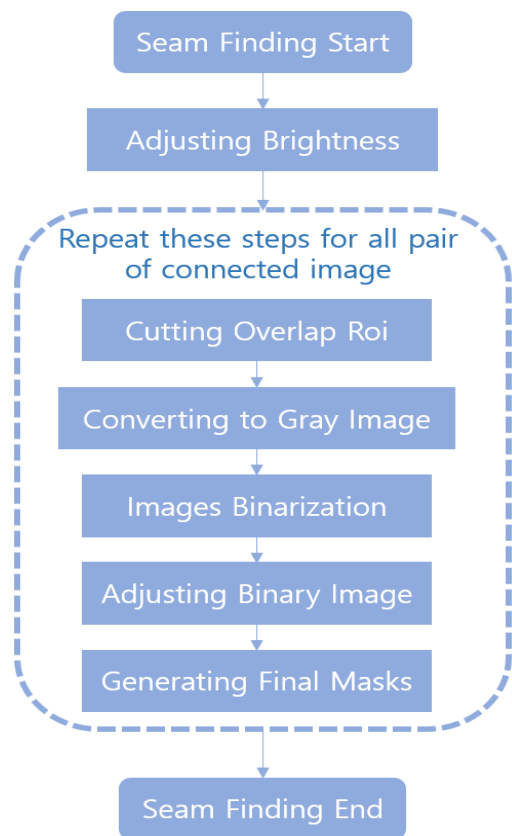


그림 10. 제안 방법의 알고리즘  
 Fig. 10. Algorithm of the proposed seam finder

마스크 추출은 영상들끼리 결합되는 경계선을 찾는 과정으로 와핑(warping) 후, 블렌딩(blending) 전에 필요한 과정이다. 그래서 다음의 마스크 추출 과정에서 사용되는 입력 영상은 와핑까지 완료한 영상이다. 본 장에서 제안하는 내용을 그림으로 표현하면 그림 11과 같고, 이 그림의 각 단계를 설명하면 다음과 같다.

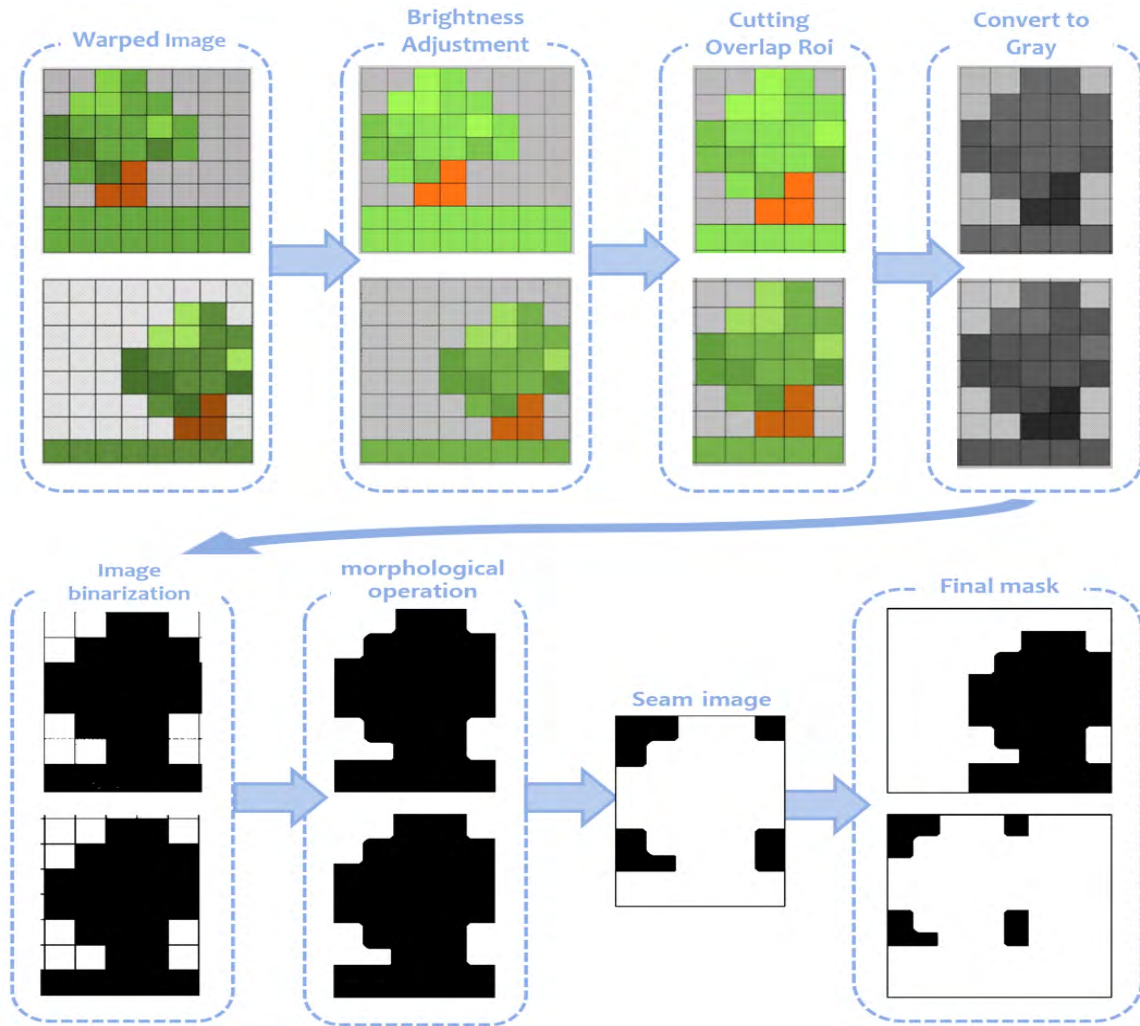


그림 11. 밝기 대비를 이용한 마스크 추출  
 Fig. 11. Mask extraction by using the difference of brightness

### 1. 영상 밝기 조정

본 논문에서 제안 방법은 영상 내의 밝기 대비를 이용하여 물체와 배경을 구분하는 마스크를 생성하는 방법이기 때문에, 같은 피사체끼리 밝기가 비슷하여야 한다. 즉, 외곽 영상의 밝기를 비슷하게 분포시켜주어야 한다. 영상들의 밝기를 균일하게 맞추기 위해, 픽셀 강도(intensity)을 업데이트시켜 주어야 한다. 밝기 강도의 업데이트를 하고 나면, 현재 영상의 밝기 평균이 전체 영상의 밝기 평균과 비슷해

진다.

빛의 삼원색  $R(x,y)$ ,  $G(x,y)$ ,  $B(x,y)$ 의 값으로 구성된 현재 영상의 x행 및 y열의 픽셀의 강도  $I(x,y)$ 를 식(3)와 같이 계산한다.

$$I(x,y) = \sqrt{R(x,y)^2 + G(x,y)^2 + B(x,y)^2} \quad (3)$$

픽셀 강도  $I(x,y)$ , 전체영상들의 밝기평균  $P_{avg}$ , 현재 영상의 밝기평균  $I_{avg}$ 을 이용하여, 업데이트 시키기 위한 가중

치  $\alpha$ 를 아래 식(5) 과 같이 계산한다. 이때,  $NP$ 는 현재 영상의 픽셀 개수를 나타낸다.

$$I_{avg} = \frac{\sum_x \sum_y I(x,y)}{NP} \quad (4)$$

$$\alpha = (1 - I(x,y) + P_{avg} - I_{avg})^2 + \sqrt{\frac{1 + P_{avg}}{1 + I(x,y)}} \quad (5)$$

$\alpha$ 을  $R(x,y)$ ,  $G(x,y)$ ,  $B(x,y)$  에 곱해  $R(x,y)$ ,  $G(x,y)$ ,  $B(x,y)$  값을 업데이트 시키면, 모든 영상의 평균 밝기가 비슷해진다. 이 경우, 밝기의 균등화에 초점을 맞춰 R, G, B 3개의 색상채널에 모두 같은 가중치를 적용했기 때문에 색조(Hue)의 변화를 고려하지 않은 과정이다. 이후 과정(3. 컬러에서 흑백으로 변환)에서 불필요한 색조 성분을 제거하여 오로지 밝기 관련된 성분만 남게 할 것이다.

## 2. 오버랩 영역 추출

경계선(seam)은 두 영상이 포개어지는 오버랩 영역 안에서 결정되어야 한다. 따라서 경계선을 찾기 전, 두 영상의 오버랩 영역만 잘라내어 따로 저장해야 한다.

## 3. 컬러에서 흑백으로 변환

색상 요소 중 색조는 ‘제한하는 마스크 추출과정’에 필요하지 않으므로, 컬러영상이라면 밝기와 관련된 성분만 남기고 색조를 제거해 주어야 한다. 이를 위해 RGB 색상표를 HSV 색상표로 변환하고, 영상 내부의 모든 픽셀의 색조(Hue)을 같은 상수값으로 설정한다. 본 논문에서는 0으로 설정하였다. 그 다음, HSV에서 GRAY로 변환시켜, 흑백영상을 만든다.

## 4. 영상의 평균, 표준편차 계산

흑백 영상 이진화를 위한 임계값을 설정하기 위해, 전 단계의 결과인 흑백영상의 밝기 강도 평균  $I_{avg}$  과 표준편차  $I_{div}$ 를 계산해야 한다.

$$I_{div} = \sqrt{\frac{\sum(I(x,y) - I_{avg})^2}{NP}} \quad (6)$$

## 5. 영상의 이진화

이제, 흑백영상을 0과 1의 값으로만 구성된 이진화 영상(Binary image)를 만들어야 한다. 그리고 0과 1 중 값을 결정할 때 기준이 되는 임계값 threshold는 아래의 식(7)과 같이 구할 수 있다. 밝기 강도가 Threshold 보다 같거나 작은 픽셀의 위치는 0의 값이 저장되고, 밝기강도가 Threshold 보다 큰 픽셀의 위치에는 1의 값이 저장된다.

$$\text{threshold} = I_{avg} + I_{div} \quad (7)$$

## 6. 이진화 영상 보정

위의 과정으로 만들어진 이진화 영상은 아직 마스크로 사용하기에 부적합하다. 이진화 영상을 보면, 작은 틈이나 구멍이 보인다. 이런 것들이 많으면 물체의 어긋남을 부각시킬 수 있고, 매우 자잘하게 틈 또는 구멍이 많다면 고스팅까지도 발생할 수 있다. 때문에 작은 크기의 폐곡선 형태의 경계선들은 제거해 주어야 한다. 이 기능을 실행시키기 위해 모폴로지 연산<sup>18)</sup>중 팽창과 침식을 이용한다.

먼저, 31정도의 커널 크기로 이진화 영상을 침식시켜, 현재 1의 값을 가진 픽셀의 주변 31개의 픽셀이 1의 값을 갖도록 한다. 이번에는 모폴로지 연산 중 팽창을 이용해, 작아진 경계선을 원래의 크기로 돌아오게 한다. 그리고 지름이 픽셀 31개의 크기보다 작은 구멍들은 아예 없어져버릴 것이다. 즉, 같은 크기로 영상을 침식 후 팽창을 하면서 작은 크기의 구멍들이 메워지게 된다.

자잘한 구멍이나 틈을 메우고 나서, 현재 어두운 물체의 가장자리를 따라 경계선이 생성된 상태이다. 기본적으로 블렌딩을 하기 위한 경계선은 주변으로부터 에너지가 작은 저주파 영역이어야 자연스러운 블렌딩영상을 만들어낼 수 있다. 물체의 가장 자리와 같은 밝기 대비가 급격한 곳에서 블렌딩된다면, 오히려 물체의 가장자리가 굉장히 부자연스러워진다. 또한 시차에 의해 어긋나는 부분은 마스

크를 통해 차단시켜야 할 필요가 있다. 이를 위해 물체의 가장자리에서 어느정도 간격을 가진 경계선을 만들어야 한다. 그리고, 크기가 11인 커널로 영상을 다시 침식시켜 이 기능을 수행한다.

마지막으로 *low pass filter*를 통과시켜 경계선을 더 자연스럽게 연결되도록 하면, 이진화 영상을 최종 마스크로 만들 준비가 끝난다.

## 7. 최종 마스크 추출

와핑영상 *image1* 과 *image2*에 대하여 앞의 6단계까지 완료한 결과영상들을 *subimage1*, *subimage2* 라 하면 비트연산을 통해서 새로운 마스크 영상 *seamimage*을 만든다. 두 영상의 같은 위치의 픽셀 중 하나라도 0의 값을 가지면 1을 새로운 마스크의 같은 위치 픽셀에 저장한다.

$$\begin{aligned} & \textit{seamimage} \\ & = OR(NOT(\textit{subimage1}), NOT(\textit{subimage2})) \quad (8) \end{aligned}$$

*seamimage*에서 1의 값을 갖는 위치에서는 *image2*의 마스크에 0의 값이 저장되고, *seamimage*에서 0의 값을 갖는 위치에서 *image1*의 마스크는 0이 저장된다.

그리고 마스크의 와핑 영상에 가중치를 주어, 통과 또는 차단시키는 역할을 한다. 만약 가중치가 1 이라면 영상은 그대로 통과되어 파노라마 영상으로 적용되지만, 가중치가 0이라면 마스크와 같은 위치의 영상은 차단되어 파노라마 영상에 보이지 않는다. 이러한 원리로, 한쪽 영상에서 통과되면, 그 위치의 모든 영상을 차단시켜야 한다. 다시 말해, 블렌딩하려는 두 영상의 오버랩 영역에서는 두 영상의 마스크는 같은 값을 가지면 안 된다. 이것이 *seamimage*를 두 영상의 마스크로 적용시킬 때, 왼쪽 영상에는 그대로, 오른쪽 영상에는 값을 반전시켜 적용하는 이유이다. 같은 이유로 오버랩 영역 밖의 영역의 마스크는 1의 값을 가지게 한다. 위와 같은 알고리즘을 통해 생성한 마스크로 블렌딩하면 배경과 물체를 구분하거나, 밝기 차가 심한 경계를 접합선으로 가지는 파노라마 영상을 얻을 수 있다.

## V. 실험 결과

본 장에서는 기존의 *seam finding* 알고리즘과 제안하는 *seam finding* 알고리즘을 스티칭 프로그램에 적용했을 때, 결과 영상들을 비교해서 다양한 기술들의 성능을 분석하였다. 기존의 알고리즘으로 *voronoi*, *dynamic programming*, *graph cut* 이 세가지 알고리즘을 이용하였다. 그림 12와 그림 13은 의도한 대로 물체를 피해가는 접합선이 만들어지는지를 확인하기 위해, 블렌딩을 제거하고 각각의 영상의 색상을 변화시켜 접합선을 눈에 보이도록 한 파노라마 영상이다. 그림 14는 제안하는 논문의 접합선이 의도한대로 생성될 때, 이 접합선이 실제로 스티칭 결과에 도움이 되는지를 확인하기 위해, 블렌딩 알고리즘까지 같이 적용해서 얻은 파노라마 영상이다. 실험 결과 그림들에서 (a) 영상은 ‘*voronoi*’ 알고리즘을, (b) 영상은 ‘*Dynamic programming*’ 알고리즘을, (c) 영상은 ‘*graph cut*’ 알고리즘을, (d) 영상은 제안한 알고리즘을 적용한 결과들을 보여준다. 본 실험의 입력 영상은 모두 삼성 갤럭시 S6 SM-G920K 카메라로 촬영한 영상이다. 그림 12는 540x960 ‘복도’ jpg 영상 2장을 스티칭한 영상이고, 그림 13과 그림 14는 960 x 540 ‘9층 연구실’ jpg 영상 78장을 스티칭한 결과 영상의 일부분을 나타낸 것이다.

그림 12는 영상의 경계선을 더 부각시킬 수 있도록, 입력 영상을 붉은 색과 녹색 계열로 색조정하고 블렌딩을 제거하여 얻은 파노라마 영상이다. *Voronoi*, *Dynamic programming*, *Graph cut* 모두 창문과 바닥을 그대로 통과하는 경계선이 생겨나지만, 제안하는 *seam finder*의 경우 창문과 바닥의 가장자리를 따라가는 경계선을 가진다. 그림 12를 통해 구현한 *seam finder*가 물체를 배경과 구분시키는 *seam*이 생성됨을 확인할 수 있다.

그림 12와 유사하게, 그림 13도 블렌딩을 제거하여, 영상 간의 밝기 차이를 이용해 경계선(*seam*)을 눈에 보이도록 하였다. 그림 13의 (a), (b), (c)에서 볼 수 있듯이 기존 *seam finder*를 적용할 경우, 의자, 가방, 사람 얼굴까지 관통하는 경계선이 만들어진다. 이에 반해, 제안하는 알고리즘을 구현한 결과 영상 (d)의 스티칭 결과는 의자와 가방을 지나가



지 않으면서, 바닥에 경계선이 만들어 진다. 그림 12와 그림 13의 (d) 영상을 통해, 제안한 마스크 추출 방식이 물체의

형태를 유지하는 경계선을 만들어 내겠다는 의도대로 제대로 기능함을 확인 할 수 있다.



(a) the image stitched by voronoi



(b) the image stitched by dynamic programming



(c) the image stitched by graph cut



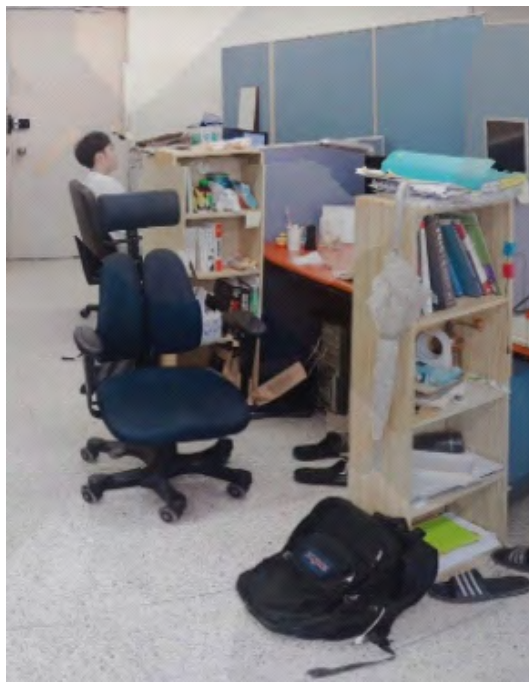
(d) the image stitched by the proposed seam finder

그림 12. Seam finder들의 성능을 비교하기 위해 블렌딩 과정을 사용하지 않고 스티칭된 영상들의 비교

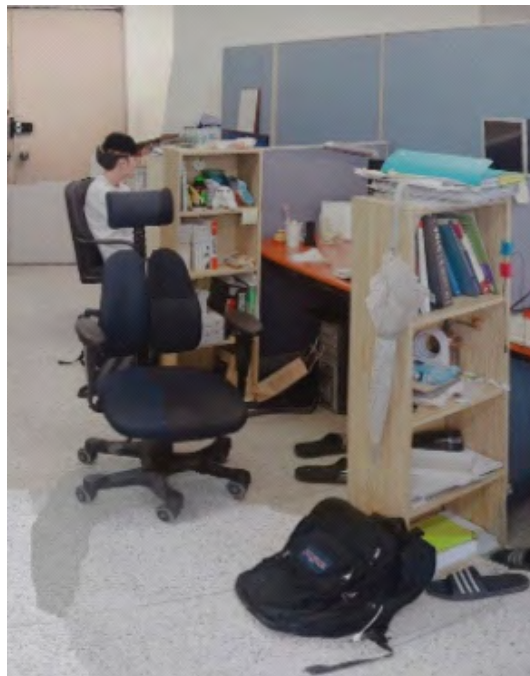
Fig. 12. The images stitched by using a variety of seam finding algorithms, where the blending process is not applied to compare the performances of the seam finder algorithms



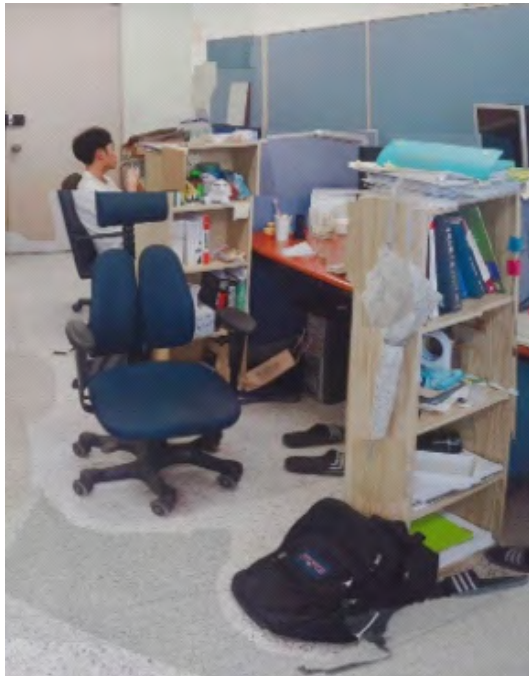
(a) the image stitched by voronoi



(b) the image stitched by dynamic programming



(c) the image stitched by graph cut



(d) the image stitched by the proposed seam finder

그림 13. Seam finder들의 성능을 비교하기 위해 블렌딩 과정을 사용하지 않고 스티칭된 영상들의 비교

Fig. 13. The images stitched by using a variety of seam finding algorithms, where the blending process is not applied to compare the performances of the seam finder algorithms



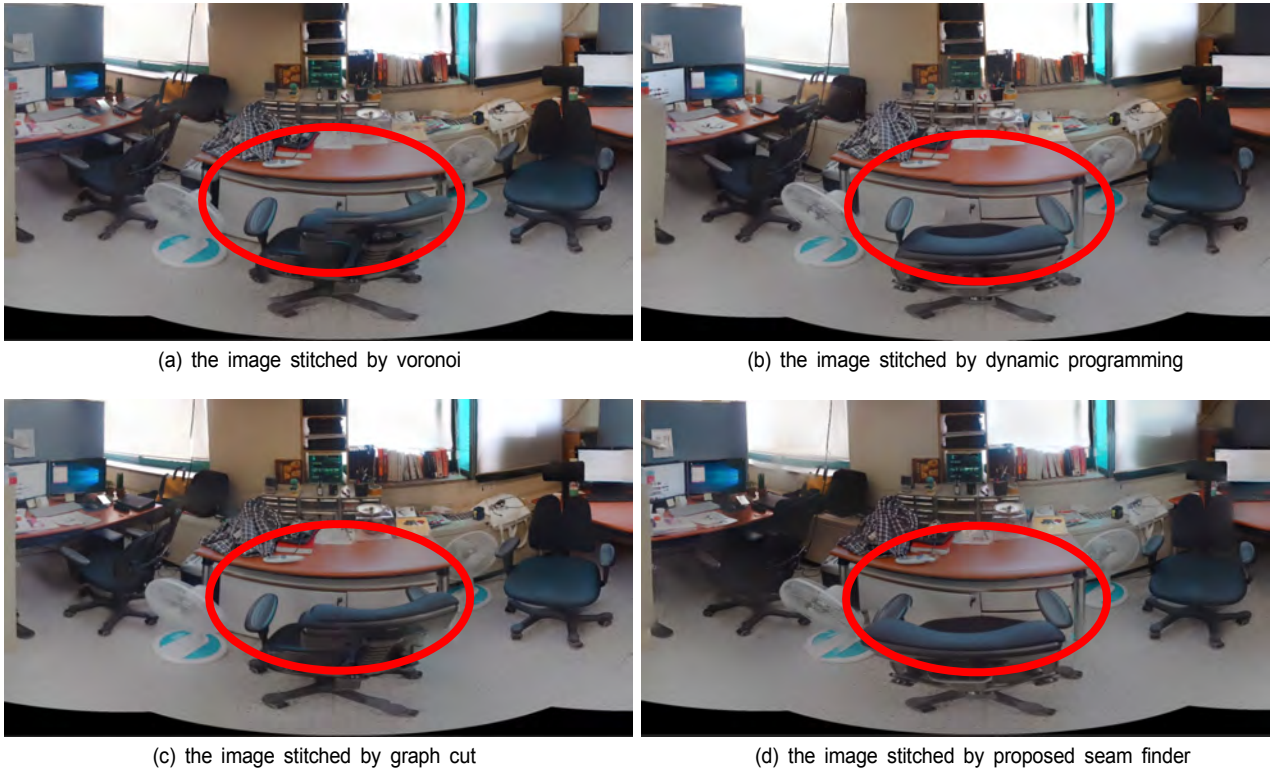


그림 14. 블렌딩 과정을 적용하는 조건에서 다양한 Seam finder들을 이용하여 스티칭된 영상들의 비교  
 Fig. 14. The images stitched by using a variety of seam finding algorithms, where the blending process is applied

그림 14는 블렌딩 과정까지 적용한 경우, 스티칭된 영상들을 비교하는 실험 결과들이다. 그림 14는 시차(view disparity)로 인해, 의자의 상대적인 위치가 다르다. 이렇게 시차가 크게 발생한 물체 부근에서는 기존의 seam finder로 자연스러운 결과를 얻을 수 없다. 그림 14의 (a), (b), (c)처럼 기존 seam finder 기법을 적용하면, 이미지 안에 동일한 대상이 중첩될 때, 그 물체를 통과하는 접합선(seam)이 만들어져서 엇갈림이 더 부각될 수도 있다. 그림 14의 (d)영상에서 확인할 수 있듯이, 제안하는 seam finder 기법을 적용했을 때, 물체가 크게 어긋나는 것을 줄여줄 수 있다.

또한 제안하는 seam finder 기법은 배경과 대상을 분리시켜 비교적 눈에 띄는 어긋남을 줄이기 위함이지만, 배경과의 분리 없이 물체와 물체가 맞닿아 있는 영상을 정합하는 경우가 있을 것이다. 일반적으로 물체의 가장자리는 급격한 밝기 변화가 일어나는 곳이기 때문에, 제안한 seam finder를 사용하면, 그림 15와 같이 대상의 가장자리의 안쪽이나 바깥쪽을 따라 접합선이 만들어 질 것이다. 이를 통해 물체의 가장자리



그림 15. 배경과 물체의 구분이 명확하지 않는 영상들의 스티칭 결과  
 Fig. 15. The stitched image when there is no clear division between background and objects

에서 어긋나거나 끊김을 방지하는 효과가 있다. 하지만, 배경과 물체 사이의 접합선과 달리, 불필요한 접합선이 생겨 오히려 깔끔하지 못한 영상을 얻을 수 있다는 한계점이 있다.

## VI. 결 론

기존의 seam finder는 dynamic programming 같이 주변 픽셀과의 변화율이 가장 작은 픽셀을 이어 접합선을 만들거나, voronoi 같이 오버랩되지 않는 영역에서 가장 가깝도록 영역을 분할한 경계선을 접합선으로 만들거나, 최소오차경계와 같이 현재 영상의 픽셀 값과 같은 위치의 중첩되는 영상의 픽셀의 값 차이가 가장 적은 픽셀을 찾아 연결하는 방식으로 접합선을 생성하였다. 이러한 기존 seam finder 들은 멀리 있는 풍경을 지면과 나란한 방향으로 촬영한 사진의 경우, 굉장히 효과적으로 기능한다. 하지만 근경과 원경에 시차가 크고, 생동감 있는 사진을 찍을 경우 seam finder 에서 추출한 접합선으로 인해 동일한 물체끼리 끊어지고, 일부가 사라지고, 어긋나는 스티칭 에러가 발생하여, 부자연스러운 영상을 얻을 수 있다. 본 논문에서는 기존 seam finder 알고리즘의 스티칭 에러의 원인이 물체를 관통하는 접합선을 seam으로 추출했기 때문이라고 분석하였다. 본 논문에서는 물체를 가로지르지 않고, 물체를 피해 배경을 가로질러 접합선을 추출하는 ‘밝기 대비를 이용한 마스크 추출 방법’을 제안하였다. 눈에 띄게 보이는 어긋남을 줄이기 위해, 배경과 물체와의 밝기차이를 이용하였고, 이러한 밝기 차이를 이용하여, 물체의 형상을 따라 이어지는 접합선을 추출했다. 다양한 실험을 통해, 본 논문에서 제안하는 기술을 사용했을 경우, 의도한대로 접합선이 생성되고, 스티칭 영상의 화질을 개선시키는 것을 확인하였다.

하지만 본 논문에서 밝기 차이가 적은 영상보다, 밝기 차이가 명확한 영상에서 접합선 경계의 끊어짐과 어긋남이 시각적으로 더 민감하게 보인다는 점에 주목하여, 밝기 차이가 적은 영상의 접합선 추출은 고려하지 않았다. 즉, 접합선 추출을 위한 비교 성분으로 밝기가 유일하기 때문에, 물체와 후경의 밝기가 비슷한 경우의 접합선 추출에는 어려움이 있다. 또한, 평평한 배경과 물체 사이뿐만 아니라 물체들로만 구성된 영상의 경우 깔끔한 접합선을 얻는 것에도 어려움이 있다. 이를 보완하기 위한 다음 연구에서는, 접합선을 추출하는 과정에 오차, 색상 등의 성분까지 고려하는 과정을 추가하고, 밝기 정보를 이용할 때 지역적인 밝기 분포에 따른 임계값 설정을 추가하여 seam finder를 개선할 예정이다.

## 참 고 문 헌 (References)

- [1] Matthew Brown and David G. Lowe, "Automatic Panoramic Image Stitching using Invariant Features," *International Journal of Computer Vision*, Vol. 74, No. 1, pp. 59-73, 2007
- [2] R. Szeliski, "Image Alignment and Stitching: A Tutorial," *Foundations and Trends in Computer Graphics and Vision*, Vol 2, Issue 1, January 2004.
- [3] Burt, P. J. and Adelson, E. H. "A multiresolution spline with applications to image mosaics," *ACM Transactions on Graphics*, Vol. 2, No. 4, pp.217-236, October 1983
- [4] Patrick Perez, Michel Gangnet and Andrew Blake, "Poisson Image Editing," *ACM Transactions on Graphics*, Vol 22, Issue 3, pp. 313-318, July 2003
- [5] Matthew Uyttendaele, Ashley Eden and Richard Szeliski, "Eliminating Ghosting and Exposure Artifacts in Image Mosaics," *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol 2, pp. 509-516, 2001
- [6] Jun Pan, Mi Wang, Deren Li, and Jonathan Li, "Automatic Generation of Seamline Network Using Area Voronoi Diagrams With Overlap," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 47, Issue. 6, pp 1737-1744, April 2009
- [7] Shai Avidan and Ariel Shamir, "Seam Carving for Content-Aware Image Resizing," *ACM Transactions on Graphics*, Vol. 26, Issue. 3, No 10, pp.1-10, July 2007
- [8] Yuri Y. Boykov and Marie-Pierre Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images," *International Conference on Computer Vision*, Vol. 1, pp.105 - 112, July 2001
- [9] Alexei A. Efros and William T. Freeman, "Image quilting for texture synthesis and transfer," *Proceedings of ACM SIGGRAPH 2001*, pp.341 - 346, 2001
- [10] Myeongah Cho, Junsik Kim, and Kyuheon Kim, "Three-Dimensional Rotation Angle Preprocessing and Weighted Blending for Fast Panoramic Image Method," *Journal of Broadcast Engineering*, Vol. 23, No. 2, pp 235- 245, March 2018
- [11] David G. Lowe, "Distinctive Image Features from Scale-Invariant Key-points," *International Journal of Computer Vision*, Vol. 60, No. 2, pp.91-110, November 2004.
- [12] Won-Jun Moon, Young-Ho Seo, and Dong-Wook Kim, "Parameter Analysis for Time Reduction in Extracting SIFT Keypoints in the Aspect of Image Stitching," *Journal of Broadcast Engineering* Vol. 23, No. 4, pp 559 -573, July 2018
- [13] Herbert Bay, Andreas Ess, Tinne Tuytelaars and Luc Van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, Vol. 10, Issue 3, pp.346 - 359, 2008
- [14] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," *International conference on computer vision*, Vol. 1, pp. 2564 - 2571, November 2011
- [15] Siyoung Park, Jongho Kim, and Jisang Yoo, "Fast Stitching Algorithm by using Feature Tracking," *Journal of Broadcast Engineering*, Vol. 20, No. 5, pp 728-737, September 2015
- [16] Martin A. Fischler and Robert C. Bolles, "Random sample consensus:

A paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, Vol. 24, No. 6, pp.381 - 395, June 1981

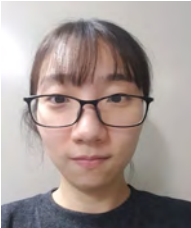
[17] Heung-Yeung Shum and Richard Szeliski, “Construction of Panoramic

Image Mosaics with Global and Local Alignment,” *International Journal of Computer Vision*, Vol. 36, pp.953 - 956, 1998

[18] R. Gonzalez and R. Woods, *Digital Image Processing*, Addison-Wesley , pp 512 - 550, 1992

---

## 저 자 소 개



남 다 윤

- 2016년 3월 현재 : 세종대학교 전자정보통신공학과 학사과정
- ORCID : <https://orcid.org/0000-0002-6573-4932>
- 주관심분야 : 영상 신호처리



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