

정규논문 (Regular Paper)

방송공학회논문지 제20권 제5호, 2015년 9월 (JBE Vol. 20, No. 5, September 2015)

<http://dx.doi.org/10.5909/JBE.2015.20.5.728>

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

## 특징점 추적을 통한 다수 영상의 고속 스티칭 기법

박 시 영<sup>a)</sup>, 김 종 호<sup>a)</sup>, 유 지 상<sup>a)†</sup>

### Fast Stitching Algorithm by using Feature Tracking

Siyoung Park<sup>a)</sup>, Jongho Kim<sup>a)</sup>, and Jisang Yoo<sup>a)†</sup>

#### 요 약

스티칭 기법은 여러 영상에서 추출한 특징점의 디스크립터를 생성하고, 특징점들간의 정합 과정을 통해 하나의 영상으로 만드는 것이다. 각각의 특징점은 128 차원의 정보를 가지고 있고, 특징점의 개수가 증가 할수록 데이터 처리 시간이 증가하게 된다. 본 논문에서는 비디오 영상을 입력 했을 때 고속 파노라마 생성을 위한 특징점 추출 및 정합 기법을 제안한다. 빠른 속도로 특징점 추출을 위해서 FAST(Features from Accelerated Segment Test) 기법을 사용한다. 특징점 정합과정은 기존의 방법과는 다른 새로운 방법을 제안한다. Mean shift를 통해 특징점이 포함된 영역을 추적하여 벡터(vector)를 구하고 이 벡터를 사용하여 추출한 특징점들을 정합하는데 사용한다. 마지막으로 이상점(outlier)을 제거하기 위해 RANSAC(RANdom Sample Consensus) 기법을 사용한다. 입력된 두 영상의 호모그래피(homography) 변환 행렬을 구하여 하나의 파노라마 영상을 생성한다. 실험을 통해 제안하는 기법이 기존의 기법보다 속도가 향상되는 것을 확인하였다.

#### Abstract

Stitching algorithm obtain a descriptor of the feature points extracted from multiple images, and create a single image through the matching process between the each of the feature points. In this paper, a feature extraction and matching techniques for the creation of a high-speed panorama using video input is proposed. Features from Accelerated Segment Test(FAST) is used for the feature extraction at high speed. A new feature point matching process, different from the conventional method is proposed. In the matching process, by tracking region containing the feature point through the Mean shift vector required for matching is obtained. Obtained vector is used to match the extracted feature points. In order to remove the outlier, the RANdom Sample Consensus(RANSAC) method is used. By obtaining a homography transformation matrix of the two input images, a single panoramic image is generated. Through experimental results, we show that the proposed algorithm improve of speed panoramic image generation compared to than the existing method.

Keyword : Panorama, Stitching, FAST, Mean shift

---

a) 광운대학교 전자공학부(Electronics Engineering, Kwangwoon University)

† Corresponding Author : 유지상(Jisang Yoo)

E-mail: jsyoo@kw.ac.kr

Tel: +82-02-940-5112

※ 본 연구는 미래창조과학부의 정보통신방송연구개발사업의 일환으로 수행하였음. [라우드니스 기반의 방송음량 기술 및 실내 환경 소음의 스트레스 평가 기술 개발 (2014-044-055-002)]

· Manuscript received June 10, 2015; Revised September 1, 2015; Accepted September 24, 2015.

## I. 서론

파노라마 영상은 제공된 여러 개의 영상을 다양한 처리를 통해 같은 영역을 찾고 스티칭(stitching)하는 과정을 거쳐 생성되며 고해상도의 넓은 시야각을 제공 한다. 그 응용 분야도 전시관과 스트리트 뷰 등으로 점차 확대되고 있다<sup>[1]</sup>. 근래에는, 파노라마 AV 기술과 인터페이스 기술을 결합하여, 3D의 입체감 및 UHD의 사실감과는 또 다른, 극대화된 현실감을 제공하는 기술들이 나오고 있다<sup>[2]</sup>.

파노라마 서비스를 제공하기 위해서는 크게 파노라마 영상 획득 기술과, 생성 기술 및 재생기술이 필요하다. 특히 한 대 이상의 카메라에서 획득된 다수의 영상으로부터 파노라마 영상을 생성하기 위해서 영상간의 기하학 보정을 위한 스티칭 기술과 색상 보정을 위한 블렌딩(Blending) 기술이 필요하다. 스티칭 기술은 입력된 영상에서 동일한 영역을 찾는 작업이 필요하고, 입력 받은 두 영상이 동일한 영역을 포함하고 있더라도 그 영역의 크기가 다르고, 회전 및 이동 된 상태와 조명 변화도 고려해야 한다. 기존에는 동일한 영역을 찾기 위한 방법으로 SIFT(Scale Invariant Feature Transform)<sup>[3]</sup>와 SURF(Speeded Up Robust Feature)<sup>[4]</sup> 기법이 많이 이용되고 있다.

SIFT의 경우 DoG(Difference Of Gaussian)<sup>[5]</sup>을 통해 주변보다 밝기 값이 어둡거나 밝은 영역의 중앙의 위치를 찾아 특징점을 검출 하고, 각각의 특징점에 대해 주변의 화소들을 이용하여 주 방향 성분과 128 또는 64 차원 벡터를 통해 디스크립터(Descriptor)를 형성한다. SURF의 경우 Haar wavelet<sup>[5]</sup>을 통해 특징점을 검출하고 특징점의 주변의 값들을 통해 64 또는 32 차원 벡터를 형성한다. 위에서 구한 디스크립터를 사용하여 서로 다른 두 영상의 정합 과정이 이루어진다. 하나의 특징점마다 128 또는 64 개의 정보를 가지고 있기 때문에 특징점이 많을수록 빠른 시간 안에 스티칭 하는데 문제가 있다.

본 논문에서는 파노라마 영상을 생성하기 위하여 고속의 스티칭 기법을 새롭게 제안한다. 입력이 비디오 영상일 경우 생기는 문제를 해결하기 위하여 아래와 같은 방법을 제안한다. 한 영상에서 특징점을 추출 할 수 있는 방법은 다양한데 그 중 FAST(Features from Accelerated Segment Test)

기법<sup>[6]</sup>을 이용한다. FAST는 SIFT의 DoG 방법이나 SURF의 Haar wavelet 방식보다 특징점을 추출하는 속도 측면에서 월등하며, 특징점으로서 중요한 요소 중에 하나인 높은 반복성(Repeatability)을 가지고 있다. 일차적으로 영상을 스티칭 하기 위해 사용하는 SIFT 또는 SURF 대신에 FAST를 사용하여 수행시간을 감소 시킨다. 그리고 비디오 영상의 경우 앞 뒤 프레임 간에 움직임이 작고 조명 변화 측면에서도 많은 차이가 없는 점을 고려하여 디스크립터 정보를 이용한 정합 방법을 대신하여, Mean shift 기법을 이용하여 특징점을 추적한다<sup>[7]</sup>. 추적한 방향을 토대로 앞 뒤 영상의 전체적인 이동 방향을 예측하여 정합과정을 진행함으로써 상당한 시간을 줄일 수 있다. 이후 기존의 방식과 비슷하게 RANSAC을 통해 이상점을 제거<sup>[8]</sup>하고 호모그래피 변환<sup>[9]</sup>을 통해서 기존의 방식보다 수행 시간을 단축 할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 입력으로 비디오 영상이 들어 올 경우 제안한 각각의 기법에 대한 설명과 전체적인 수행 방법에 대해 설명한다. 3장에서는 SIFT와 SURF에 의해 생성되는 파노라마 기법에 대한 수행시간과 제안한 기법의 수행 시간을 비교하고 결과를 제시한다. 4장에서는 결론을 내린다.

## II. 제안하는 고속의 영상 스티칭 기법

본 논문에서 제안하는 고속의 영상 스티칭(Stitching) 기

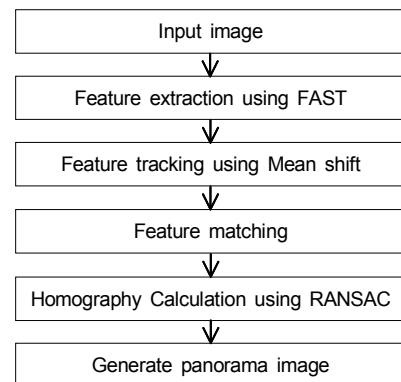


그림 1. 제안하는 고속 스티칭 기법의 흐름도  
Fig 1. Flowchart of the proposed fast stitching algorithm

법의 흐름도는 그림 1과 같다. 먼저 비디오 영상을 입력 받고 FAST 기법을 이용하여 일정한 간격의 프레임에서 특징점을 추출한다. 추출된 특징점을 기반으로 mean shift 기법을 적용하여 이동 방향을 추적하고 이 결과를 정합과정에 적용한다. 잘못 정합된 특징점을 제거하기 위해 RANSAC 기법을 적용한 후 두 영상의 호모그래피 관계를 구하여 하나의 파노라마 영상을 생성하게 된다.

## 1. FAST 기법을 이용한 특징점 추출

FAST(Features from Accelerated Segment Test) 기법은 기존의 특징점 추출 기법인 Harris corners<sup>[10]</sup>, SIFT(Scale Invariant Feature Transform)의 DoG(Difference Of Gaussian), SURF(Speeded Up Robust Feature)의 Haar wavelet과 SUSAN(Smallest Uni-Value Segment Assimilating Nucleus Test)<sup>[11]</sup> 기법 등과 특징점 추출 시간을 비교했을 때 상당히 빠르다는 장점이 있다. FAST를 이용하여 특징점을 추출하는 방법은 다음과 같다. 그림 2는 특징점을 결정하는 중앙 화소와 주변 화소의 관계를 보여주고 있다<sup>[6]</sup>.

먼저 그림 2와 같이 영상의 한 화소  $p$ 를 중심으로 거리가 3인 원을 형성한다. 형성된 원에 걸쳐있는 16개의 화소들(그림 2에서 원이 지나는 화소)과 중심 화소  $p$ 의 밝기 값을 비교 한다. 16개 화소의 밝기 값과  $p$  화소의 밝기 값의 차이가 임계값(Threshold) 이상인 값들이 연속적으로  $N$ 개 이상 존재할 경우 중심의  $p$  화소를 특징점으로 정의한다. [6]에서는 임계값을 중심 화소  $p$ 의 밝기 값에 따라 가변적으로 적용하여 밝기에 무관하게 특징점을 추출할 수 있도록 하였으며,  $N$ 이 9일 경우 특징점의 반복성(Repeatability)이 가장 높음을 보인다.

신속한 특징점 판별을 위해서 원을 구성하는 16개 화소 중  $N$ 개의 연속하는 화소가 서로 비슷한 밝기 값을 가져야 하는 특성을 고려해서 단순히 한 방향으로 검색하는 것이 아니라 가장 먼저 1번 화소의 밝기 값을 비교하고 다음 9번 화소, 5번 화소 그리고 13번의 화소의 밝기 값을 확인 한다.  $N$ 이 9일 경우, 1번과 9번 화소의 밝기가 중앙 화소 값보다 밝지만 5번 화소의 밝기가 중앙 화소 값보다 어두울 경우 9개가 연속적으로 밝거나 어두울 수 없기 때문에 나머지

비교 과정을 생략하여 속도를 개선 할 수 있다. 반대로 5번 화소의 밝기 값이 밝을 경우 9개가 연속적으로 밝기가 밝을 확률이 생기기 때문에 다른 화소의 밝기 값을 비교한다<sup>[6]</sup>.

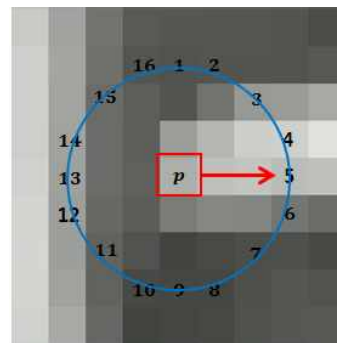


그림 2. 화소  $p$ 를 중심으로 하는 원 위의 16개 화소  
Fig 2. 16 pixels on a circle with the center pixel  $p$

위의 과정을 모든 화소에 대해 반복적으로 수행하기 때문에 입력 받은 영상의 크기가 커지면 처리 속도 또한 증가하는 문제가 있다. 이 문제를 해결하고자 결정 트리 구조를 이용한다. 트리 구조를 만들기 위해 우선 각각의 화소들의 관계를 4가지 상태로 구분한다. 식 (1)에서 화소들의 관계에 따른 상태를 보여준다.

$$S_{p \rightarrow x} = \begin{cases} d, & I_{p \rightarrow x} \leq I_p - t \\ s, & I_p - t \leq I_{p \rightarrow x} \leq I_p + t \\ b, & I_p + t \leq I_{p \rightarrow x} \\ u, & \text{unknown} \end{cases} \quad (1)$$

여기서  $I_p$ 은 중앙화소의 밝기 값을 나타내고,  $I_{p \rightarrow x}$ 는 주변 16개의 화소의 밝기 값을 나타내고  $t$ 는 임계값을 나타낸다.  $S_{p \rightarrow x}$ 는 주변 16개의 상태를 설정해주는 값으로써 중앙 화소의 밝기 값보다 작을 경우  $d$ (darker)가 되고 차이가 일정 임계값 사이일 경우  $s$ (similar)가 된다. 중앙 화소의 밝기 값보다 클 경우  $b$ (brighter)가 된다. 마지막으로 비교하지 않은 상태일 경우  $u$ (unknown)가 된다<sup>[6]</sup>.

그림 3은 식 (1)을 가지고 16개의 상태를 나타낸 후, 트리를 구성하여 깊이 탐색 방식(Depth first search)으로 전체 화소의 특징점 추출을 하는 과정을 보여준다.

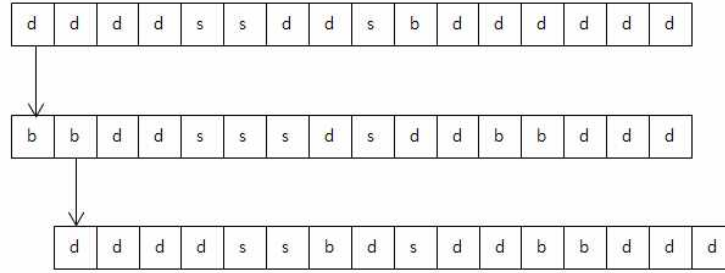


그림 3. 결정 트리 구조  
Fig 3. Decision tree structure

가장 먼저 모든 화소 간의 상태를 **unknown**으로 지정한다. 이후 그림 3처럼 먼저 임의의 화소  $p$ 에 대하여 주변 16개의 화소와 비교과정을 거친다. 그리고 깊이 탐색 방식에 의해서 첫 번째 화소가 중심이 되어 다시 주변 16개의 화소들과 비교하는 과정을 반복한다. 식 (1)의 과정을 영상의 모든 화소를 중심으로 주변 16개의 화소들과 비교하여 상태를 정의해야 한다. 상태 비교는 두 화소의 값 차이를 통해 결정되기 때문에 결정 트리 구조를 통해서 두 화소 간의 상태가 **s(similar)**일 경우 식 (1)의 과정 없이 **s**로 저장하게 되고, **d(dark)**일 경우 **b(bright)**로 저장한다. 두 화소 비교 과정에서 이러한 경우가 적어도 50%를 차지하기 때문에 결정 트리 구조를 통해 수행 시간을 줄일 수 있다.

FAST<sup>[6]</sup>를 이용한 특징점 추출 기법은 주변의 밝기 값과의 차이를 통해서 특징점을 추출하기 때문에 SIFT의 DoG의 특징점 검출 방식과 비슷하지만 빠른 속도를 자랑한다. 하지만 FAST의 문제점은 SIFT의 DoG나 SURF의 Haar wavelet과는 다르게 **scale**의 변화에 취약하기 때문에 추출된 특징점들

이 모여 있는 경우가 많다. 영상의 특징점을 정합할 경우 서로 모여 있는 특징점들을 가지고 디스크립터(descriptor)로 사용하거나 정합과정을 진행할 때 오류가 생길 확률이 높아진다. 이를 해결하고자 NMS(Non Maximum Suppression)방법을 통해서 모여 있는 특징점들 중에서 값이 가장 큰 하나의 특징점을 선택하게 된다. 특징점 하나를 선택하기 위해서는 각각의 특징점마다 값을 부여해주는 과정이 필요하다. 식 (2)는 특징점의 수치화 과정을 보여준다<sup>[6]</sup>.

$$V = \max(\sum_{x \in S_{right}} |I_{p \rightarrow x} - I_p| - t, \sum_{x \in S_{dark}} |I_p - I_{p \rightarrow x}| - t) \quad (2)$$

여기서  $I_{p \rightarrow x}$ 와  $I_p$ 는 각각 주변 화소의 밝기 값과 중앙 화소의 밝기 값을 나타낸다. 먼저 식 (1)에 따라 주변 16개의 화소의 상태를 결정한다. 그 중 **b(bright)** 상태에 해당하는 화소의 경우와 **d(dark)** 상태에 해당하는 화소의 경우에 대해 각각 식 (2)를 적용하여 더 큰 값을 해당 특징점의 값으로 부여한다<sup>[6]</sup>.



그림 4. FAST를 이용한 특징점 추출한 예; (a) NMS 적용하지 않을 경우 (b) NMS 적용한 경우  
Fig 4. Examples of feature extraction using FAST; (a) Not apply the NMS (b) Apply the NMS

수치화된 특징점들을 가지고 NMS 과정을 통해 중심 값이 가장 큰 부분만을 특징점으로 추출한다. 그림 4는 N이 9이고, 임계값이 20일 때 그림 4.(a)는 NMS 기법을 적용하지 않은 경우이고 그림 4.(b)는 NMS 기법을 적용한 경우이다. 정수기의 좌측 상단의 특징점이나 거울 상단의 특징점의 개수가 줄어든 모습을 볼 수 있다.

## 2. Mean shift를 이용한 특징점 추적

특징점 정합과정은 크게 두 가지 조건에 의해서 수행 시간이 결정 된다. 첫 번째는 영상에서 추출 되는 특징점의 수와 관련이 있다. 특징점의 개수는 영상의 크기, 조명 변화에 따라 다르겠지만 일반적으로 수십 개의 특징점이 추출된다. 두 번째는 두 영상의 특징점들을 서로 정합시켜 주기 위해서 추출한 특징점들에 대해 디스크립터(Descriptor)를 형성하고 디스크립터를 통해 특징점들 사이를 비교할 수 있는 벡터들을 사용하여 특징점을 정합시켜주는 과정이다. 기존의 기법으로 특징점을 정합하는 과정에 걸리는 수행 시간을 위에서 언급한 두 가지 조건을 가지고 계산한다면 오랜 수행 시간이 필요하다. 본 논문에서는 특징점 정합에 필요한 수행 시간을 줄이기 위한 방법을 제안 한다. 일반적으로 영상에서 물체를 추적하는 알고리즘은 Optical flow<sup>[12]</sup>, Mean shift<sup>[7]</sup> 등 여러 가지 기법이 존재 한다. 제안한 방법에서는 일정한 영역의 히스토그램을 이용하여 이동 방향을 추적하는 기법인 Mean shift를 사용한다.

Mean shift는 데이터 분포의 정점이나 무게 중심을 찾는 기법이다<sup>[13]</sup>. 이 기법은 영상에서 물체의 이동을 추적하기 위해서 우선 물체가 포함된 영역에 대한 색상 히스토그램(histogram) 분포를 구한다. 그 다음 이전 영상에서 구한 히스토그램 정보를 가지고 다음 영상에서 이전 영역과 같은 크기로 모든 영역에 걸쳐서 히스토그램 정보가 가장 비슷한 영역을 찾는다. 입력되는 영상에서 모든 영역에 대해 각각 히스토그램을 구하고 비교하는 과정은 시간이 오래 걸린다. 히스토그램 역전사(Histogram back-projection) 기법과 Mean shift를 결합한 방법을 사용하여 실시간으로 추적하는 것이 가능하다<sup>[7]</sup>.

제안한 특징점 정합 과정은 먼저 이전 영상에서 찾고자

하는 영역을 설정해 주는 작업이 필요하다. 특징점 추적이 가능하기 위해서는 설정한 영역이 이전 프레임과 다음 프레임에서 모두 존재해야 되고, 알고리즘을 적용하기 전에는 영상의 이동 방향을 예측하지 못하기 때문에 영상의 중심에 가까운 영역일수록 영상의 이동방향과 상관없이 다음 프레임에 나올 확률이 높다. 따라서 특징점들 중에서 영상 중심에 가까운 특징점을 선택하여 이 특징점이 포함된 일정한 크기의 사각형 영역을 설정해 준다. 다음은 설정한 영역을 Mean shift에 적용하는 과정으로 식 (3)을 이용해 설정한 영역을 히스토그램 확률분포로 변환해 준다.

$$H_{model} = \{q_u\}_{u=1,n} \quad \sum_{u=1}^n q_u = 1 \quad (3)$$

여기서  $q_u$ 는 각 밝기 값의 확률 값을 나타낸다.  $H_{model}$ 은 설정한 영역의 밝기 값의 확률 히스토그램 정보를 가지고 있다. 식 (4)는 다음 영상의 밝기 값들을 확률 값으로 변화 시켜준다.

$$w(x) = \sqrt{\frac{H_{model}(I(x))}{H_{backprojection}(I(x))}} \quad (4)$$

여기서  $I(x)$ 는 화소의 밝기 값이고,  $H_{backprojection}(I(x))$ 는 다음 영상의 전체 영역에 대한 히스토그램 역전사 값이다. 식 (4)를 통해서 다음 영상의 값들을 확률 값으로 변환시켜 준다. 식 (5), (6)은 확률 분포를 이용해 이동방향을 변환 시켜준다.

$$\Delta x = \frac{\sum w(x)(x - x_{old})K(x - x_{old})}{\sum w(x)K(x - x_{old})} \quad (5)$$

$$x_{new} = x_{old} + \Delta x \quad (6)$$

여기서  $K$ 는 커널 함수로서 배경의 영향을 줄이기 위한 용도로 방사형의 함수가 사용된다.  $\Delta x$ 는 식 (4)에 의해서 확률이 높은 방향으로의 변화량을 보여주고,  $x_{new}$ 는 변화량에 따른 이동을 보여준다.



그림 5. 특징점 추적; (a) 이전 영상에서 설정한 영역 (b) 다음 영상에서 찾은 이동 영역

Fig 5. A feature tracking; (a) Setting a region in the previous image (b) Moving a region found at the next image

식 (5)과 (6)을 반복적으로 수행하면 한 지점으로 수렴하게 되어 설정한 영역과 가장 비슷한 영역을 다음 영상에서 찾을 수 있다. 색상 히스토그램을 사용하기 때문에 영역이 클수록 정확한 값을 찾을 수 있지만, 이에 따라 확률 계산과정이 오래 걸려 영역을 추적하는 시간이 차이가 날 수 있기 때문에 상황에 따라 가변적으로 영역의 크기를 설정해야 한다. 그림 5(a)는 이전 영상에서 특징점을 중점으로 임의로 설정한 영역을 보여 주고 있고, 그림 5(b)는 다음 영상에서 mean shift를 통해 찾은 영역을 보여 준다.

### 3. 특징점 정합

입력으로 들어오는 비디오 영상에서 일반적으로 각 프레임 간의 움직임의 변화는 크지 않다. 프레임 간의 변화가 작기 때문에 영상의 전체적인 이동 방향을 같다고 볼 수 있다. 2-2절에서 구한 영역의 이동 벡터를 통해 정합 과정을 진행한다. 식 (7)은 영역의 이동방향 벡터를 계산하는 식이다.

$$\vec{V1} = P1_{dst} - P1_{src} \quad (7)$$

여기서  $P1_{dst}$ 는 이전 영상에서 설정한 영역의 왼쪽 상단 좌표를 나타내고  $P1_{src}$ 는 다음 영상에서 찾은 영역의 왼쪽 상단 좌표를 나타낸다.  $\vec{V1}$ 는 영역의 이동방향 벡터로  $P1_{dst}$ 와  $P1_{src}$ 의 차이를 통해 구할 수 있다.

$\vec{V1}$ 를 현재 프레임 각각의 특징점에 더한 후 다음 프레임에서 변경된 위치를 중심으로 영역을 설정한다. 설정된 영역에 특징점이 존재 할 경우 정합 과정을 거친다. 먼저  $\vec{V1}$ 를 가지고 이전 프레임 영역을 설정하기 위한 특징점에 대한 정합과정을 수행한다. 식 (8)은 특징점의 이동방향 벡터를 계산하는 식이다.

$$\vec{V2} = P2_{dst} - P2_{src} \quad (8)$$

여기서  $P2_{src}$ 는 이전 영상에서 설정된 영역의 특징점 좌표를 나타내고  $P2_{dst}$ 는 추적 과정을 통해 다음 영상에서



그림 6.  $\vec{V1}$ ,  $\vec{V2}$ 를 이용한 특징점 정합

Fig 6. Feature matching using  $\vec{V1}$  and  $\vec{V2}$

정합된 특징점의 좌표이다.  $\overrightarrow{V2}$ 는 좌표  $P2_{src}$ 와  $P2_{dst}$  간의 이동방향 벡터를 나타낸다.

정합하는 과정에서 생길 수 있는 문제는 FAST로 추출한 특징점들이 가까운 위치에 존재할 경우 한 영역 안에 다수의 특징점이 존재할 수 있다는 것이다. 이 경우 위에서 구한  $\overrightarrow{V2}$ 를 통해 가까운 특징점과 정합한다. 그림 6은 위에서 구한  $\overrightarrow{V1}$ ,  $\overrightarrow{V2}$ 를 이용하여 특징점 간의 정합 과정을 보여주고 있다.

#### 4. RANSAC을 통한 호모그래피 계산

두 영상을 하나의 영상으로 만들어 주기 위해서는 하나의 카메라 좌표계를 중심으로 호모그래피(Homography) 변환을 적용해야 한다. 호모그래피 행렬은 정합된 특징점 네 쌍을 가지고 구할 수 있다. 행렬을 구할 네 쌍의 특징점들을 선택하는 과정에서 이상점(Outlier)이 스티칭을 하기 위한 특징점 쌍으로 이용될 경우 잘못된 호모그래피 행렬을 구할 수 있다. 이러한 이상점들을 제거하기 위하여 RANSAC 기법을 사용한다. RANSAC 기법은 특징점들의 쌍 중에서 무작위로 4개의 쌍을 뽑아서 호모그래피 행렬을 생성한 후 이전 프레임을 호모그래피 변환한다. 호모그래피 변환에

의해 이전 영상 특징점들의 좌표가 변환된다. 식 (9)는 호모그래피 좌표 변환 식이다.

$$\begin{bmatrix} X_2 \\ Y_2 \\ 1 \end{bmatrix} = H \begin{bmatrix} X_1 \\ Y_1 \\ 1 \end{bmatrix} \quad (9)$$

여기서  $X_1$ ,  $Y_1$ 은 이전 프레임의 특징점의 좌표이고,  $X_2$ ,  $Y_2$ 는 호모그래피 행렬에 의해 변경된 좌표이다.  $X_2$ 와  $Y_2$ 가 정합되었던 좌표와 유클리디언 거리를 통해서 가장 차이가 작은 특징점 네 쌍을 찾아서 영상을 스티칭 한다.

### III. 실험 방법 및 결과

파노라마 영상을 생성하기 위해 본 논문에서는 1920x1080 해상도의 Samsung사의 갤럭시 노트로 직접 획득한 비디오를 실험 영상으로 사용한다. 실험 영상은 각각 정수기와 건물이 포함된 영상이다. 실험 환경은 Microsoft사의 Microsoft Visual Studio C++ 2010과 OpenCV 2.4.8 라이브러리를 이용하여 구현하고, 3.40GHz의 인텔 i5 쿼드코어 프로세서를 이용한다.

표 1. 특징점 추출 수행 시간 및 특징점 개수 비교

Table 1. Feature extraction processing time and number of feature points

Mode	DoG(SIFT)	Haar wavelet(SURF)	Proposed algorithm
Water Purifier(time)	2071.9ms	135.8ms	3.23ms
Water Purifier(feature count)	746	542	284
Building(time)	4553ms	176.81ms	5.179ms
Building(feature count)	3739	1995	1236

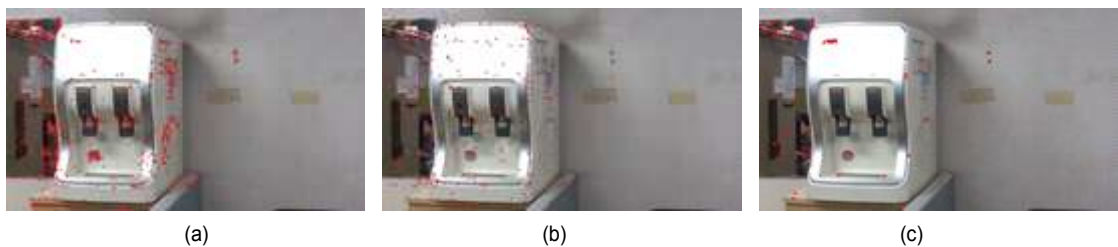


그림 7. 다양한 기법을 통하여 추출된 특징점들; (a) DoG (b) Haar wavelet (c) FAST

Fig 7. Extracted features using various methods; (a) DoG (b) Haar wavelet (c) FAST



표 1은 특징점 추출을 위해 제안한 FAST와 기존에 주로 사용하는 SIFT의 DoG와 SURF의 Haar wavelet 부분의 수행 시간을 비교하고 각 기법에 따라 추출된 특징점 개수를 비교한다. 기존의 DoG와 Haar wavelet를 사용하는 대신에 FAST를 이용했을 때 수행 시간을 표 1과 같이 줄일 수 있다. 특징점의 개수는 상대적으로 적은 것을 볼 수 있다.

그림 7은 같은 해상도의 정수기 입력 영상에 대해 각기 다른 기법을 통해 특징점을 추출한 영상이다.

표 2는 mean shift 이동 방향 검출을 이용한 정합 방법과 기존의 SIFT와 SURF를 통한 정합 시간의 비교 결과이다. 특징점 정합 과정의 수행시간에 영향을 미치는 주된 요인은 특징점 개수, 디스크립터 차원과 정합 기법이 될 수 있다. SIFT와 SURF에서 사용하는 기법은 특징점 정합을 위해 디스크립터를 형성하고 이를 비교 요소로 사용하지만 제안한 방법은 입력이 비디오 영상일 경우 디스크립터를 사용하지 않고 이동방향을 기반으로 정합 과정을 수행하였기 때문에 표 2와 같이 빠른 수행 시간의 결과를 얻을 수 있었다. 표 1에서 같은 특징점 추출 기법이더라도 특징점 추출 개수가 다름을 볼 수 있다. 표 1과 표 2에 의해서 특징점 추출 개수가 많으면 특징점 정합 수행 시간에도 영향을 끼쳐 더 많은 수행 시간이 걸리는 것을 알 수 있다.

표 2. 특징점 정합 수행 시간 비교

Table 2. Comparison of the feature matching time

Mode	SIFT	SURF	Proposed algorithm
Water Purifier(time)	355.7ms	112.71ms	14.89ms
Building(time)	1326.1ms	413.7ms	26.97ms

표 3은 제안한 기법의 각 단계별 수행 시간과 전체 파노라마 생성 시간을 보여준다. 각 단계는 2장에서 제안한 흐름도에 따라 구성하였고 두 영상을 하나의 파노라마 영상으로 만드는데 걸리는 수행시간이다.

표 3 제안하는 기법의 단계별 수행 시간

Table 3 Step by step processing time of the proposed method

Process	Water Purifier(time)	Building(time)
Feature extraction using FAST	3.23ms	5.179ms
Feature tracking using Mean shift	14.3ms	9.95ms
Feature matching	0.59ms	17.02ms
Homography Calculation using RANSAC	196.1ms	177.43ms
Total	218.22ms	209.579ms



그림 8. 비디오 영상을 사용한 파노라마 영상 생성

Fig 8. Panoramic image generation using video sequence



그림 8은 비디오 영상을 입력으로 주었을 때 생성한 파노라마 영상을 보여주고 있다.

기존 방식의 경우 시간이 오래 걸리지만 움직임 추정하는 방식이 아닌 디스크립터 생성 및 정합 과정을 거치기 때문에 비슷한 영역이 있으면 얼마든지 파노라마 영상을 생성할 수 있다. 반면에 제안하는 기법의 경우 특징점 영역의 움직임 추정을 통한 정합 과정이기 때문에 비슷한 영역이 있더라도 움직임이 없으면 파노라마 영상을 생성하지 못한다. 하지만 비디오 영상으로 입력이 주어지면, 움직임을 놓칠 가능성이 적고 표 1과 표 2에 따라 수행시간을 단축 할 수 있기 때문에 더 효과적으로 파노라마 영상을 생성할 수 있다.

#### IV. 결 론

본 논문에서는 비디오 영상이 입력으로 들어 올 경우 기존에 사용하는 방법 대신 수행 시간을 줄이기 위한 새로운 스티칭 기법을 제안하였다. 제안하는 기법에서는 비디오 영상은 프레임간에 움직임이 많이 없다고 가정하였다. 입력으로 들어온 영상에 대해 FAST 기법을 통해 빠른 시간 안에 특징점을 추출 할 수 있었다. 기존에 사용하였던 디스크립터 생성과 이를 이용한 정합 방법보다 mean shift를 이용한 특징점 영역의 이동방향을 추정하여 두 영상간의 특징점 정합 과정을 수행하여 기존의 방식보다 빠른 속도를 얻을 수 있었다.

#### 참 고 문 헌 (References)

- [1] Y. J. Cho, J. M. Seok, S. Y. Lim, S. W. An, J. I. Seo, and J. H. Chan, "Post-UHD Realistic media, high quality panoramic AV technology", *Electronics and Telecommunications Trends*, vol. 20, no. 3, pp. 33-46, June 2014.
- [2] Y. J. Lee, Y. J. Joe, M. S. Ki, S. Y. Lim, H. G. Lee and J. H. Cha, "High quality human fusion type panorama services", *The Korean Institute of Communications and Information Sciences*, vol. 28, no. 6, pp. 11-20, 2011, 5.
- [3] D. Lowe, "Distinctive image features from scale-invariant keypoints", *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, November 2004.
- [4] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool, "Speeded-up robust feature", *Computer Vision and Image Understanding*, vol. 10, no. 3, pp. 346-359, June 2008.
- [5] L. M. J. Florack, B. M. Ter Haar Romeny, J. J. Koenderink, M. A. Viergever, "General intensity transformations and differential invariants", *Journal of Mathematical Imaging and Vision*, Vol. 4, no. 2, pp. 171-187, May 1994.
- [6] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection", *European Conference on Computer Vision*, Graz, Austria, pp. 430-443, May 2006.
- [7] D. Comaniciu, V. Ramesh and P. Meer, "Real-time tracking of non-rigid objects using mean shift", *Proc. 2000 IEEE Conference Computer Vision and Pattern Recognition*, vol. 2, pp. 142-149, June 2000.
- [8] M. A. Fischler and R. 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.
- [9] E. Dubrofsky, "Homography estimation", *UNIVERSITY OF BRITISH COLUMBIA*, March 2009.
- [10] C. Harris and M. Stephens, "A combined corner and edge detector", *proceedings of the 4th Alvey Vision Conference*, pp. 147-151, August 1988.
- [11] S. M. Smith and J. M. Brady, "Susan - a new approach to low level image processing", *International Journal of Computer Vision*, vol. 23, no. 1, pp. 45-78, May 1997.
- [12] J. Tompkin, *Optical flow an introduction*, University College London((UCL) Computer Science Dept., machine Vision - Practical 2, March 2008.
- [13] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis", *IEEE Computer Society*, vol. 24, no. 5, pp. 603-619, May 2002.

---

저 자 소 개

---



박 시 영

- 2015년 2월 : 광운대학교 전자공학과 학사
- 2015년 3월 ~ 현재 : 광운대학교 전자공학과 석사
- ORCID : <http://orcid.org/0000-0003-1043-1499>
- 주관심분야 : 컴퓨터 비전, 영상 식, 영상 신호 처리



김 종 호

- 2015년 2월 : 광운대학교 전자공학과 학사
- 2015년 3월 ~ 현재 : 광운대학교 전자공학과 석사
- 주관심분야 : 영상 인식, 영상 신호 처리



유 지 상

- 1985년 2월 : 서울대학교 전자공학과 학사
- 1987년 2월 : 서울대학교 전자공학과 석사
- 1993년 5월 : Purdue Univ. EE, ph>D
- 1997년 9월 ~ 현재 : 광운대학교 전자공학과 교수
- 주관심분야 : 웨이블릿 기반 영상처리, 영상압축, 영상인식, 비선형 신호처리