

# Efficient Image Stitching Using Fast Feature Descriptor Extraction and Matching

Sang-Burm Rhee<sup>†</sup>

## ABSTRACT

Recently, the field of computer vision has been actively researched through digital image which can be easily generated as the development and expansion of digital camera technology. Especially, research that extracts and utilizes the feature in image has been actively carried out. The image stitching is a method that creates the high resolution image using features extract and match. Image stitching can be widely used in military and medical purposes as well as in variety fields of real life. In this paper, we have proposed efficient image stitching method using fast feature descriptor extraction and matching based on SURF algorithm. It can be accurately, and quickly found matching point by reduction of dimension of feature descriptor. The feature descriptor is generated by classifying of unnecessary minutiae in extracted features. To reduce the computational time and efficient match feature, we have reduced dimension of the descriptor and expanded orientation window. In our results, the processing time of feature matching and image stitching are faster than previous algorithms, and also that method can make natural-looking stitched image.

**Keywords :** Image Stitching, Feature Descriptor, SURF

## 빠른 특징점 기술자 추출 및 정합을 이용한 효율적인 이미지 스티칭 기법

이 상 범<sup>†</sup>

### 요 약

최근 디지털 카메라 기술의 발전으로 이미지를 쉽게 생성할 수 있어 이를 활용한 컴퓨터 비전분야의 연구가 활발하게 진행되고 있다. 특히 디지털 이미지에서 특징점을 추출하고 이를 활용하는 연구가 활발하게 진행되고 있다. 이미지 스티칭은 여러 이미지에서 특징점을 추출하고 이를 정합하여 하나의 고해상도 이미지를 생성하는 것으로 군사용, 의료용뿐만 아니라 실생활의 다양한 분야에서 활용되고 있다. 본 논문에서는 특징점 기술자의 차원을 효과적으로 감소시켜 정확하면서도 빠르게 정합점을 찾을 수 있는 SURF 기반의 빠른 특징점 기술자 추출 및 정합을 이용한 효율적인 이미지 스티칭 기법을 제안한다. 추출된 특징점에서 불필요한 특징점을 분류하여 특징점 기술자를 생성한다. 이때 특징점 기술자의 연산량을 줄이면서도 효율적인 정합을 위해 기술자의 차원을 줄이고 방향 윈도우를 확장하였다. 실험 결과 특징점 정합 및 전체 이미지 스티칭 속도가 기존의 알고리즘보다 빠르면서도 자연스러운 스티칭된 이미지를 생성할 수 있었다.

**키워드 :** 이미지 스티칭, 특징점 기술자, SURF

### 1. 서 론

최근 이미지에서 특징점을 추출하고 이를 활용하는 분야에 대한 연구가 활발하게 진행되고 있다. 그 중에서도 특히 이미지 스티칭 기법은 영상처리 분야에서 많은 관심을 가지고 있는 연구 분야이다. 이미지 스티칭은 두 개 이상의 이

미지에서 중첩된 부분을 서로 정합하여 넓은 화각을 가지고 있는 하나의 고해상도 이미지를 생성하는 방법이다[1,2,3]. 일반적으로 이미지 스티칭 기법은 항공 및 위성으로부터 얻어진 이미지를 정합하여 정찰 등의 군사적 목적으로 사용되고 있다. 뿐만 아니라 가상 현실 인식, 물체 인식 및 의료 분야에서도 다양하게 활용되고 있다.

이미지 스티칭에서 핵심적인 부분은 여러 이미지에서 중첩된 부분을 찾고 중첩된 부분에서 각각의 이미지에 대한 상호관계를 도출하는 것이다[4,5]. 상호관계 도출을 위해서는 입력 이미지에서 정확한 특징점을 찾고 다른 이미지에서 대응되는 정합점을 찾는 것이 가장 중요하다. 정확한 특징점

\* 이 연구는 2010년도 단국대학교 대학연구비의 지원으로 연구되었음.

† 중신회원: 단국대학교 공과대학 학장

논문접수: 2012년 10월 15일

수정일: 1차 2012년 12월 13일

심사완료: 2012년 12월 13일

\* Corresponding Author: Sang-Burm Rhee(sbrhee@dankook.ac.kr)

추출은 이미지 스티칭에서 중요한 요소이지만 연산량이 많기 때문에 전체 이미지 정합 성능에 큰 영향을 미치게 된다. 이 이미지에서 특징점을 많이 추출하게 되면 정합점을 찾을 수 있는 정보가 풍부하여 정확한 이미지 스티칭을 수행할 수 있다. 그러나 많은 특징점 추출로 인해 특징점 기술자 생성 및 이를 바탕으로 수행되는 정합 과정에서 서로 비교해야 할 대상이 많아지게 되므로 속도가 저하되는 문제점을 가지고 있다. 반대로 이미지에서 특징점이 적게 추출된다면 정합점을 찾기 위한 기술자 생성 정보가 빈약하게 되어 정확한 정합점을 찾기 어려워지는 문제점을 야기할 수 있다. 이러한 문제를 해결하기 위해서는 이미지에서 필요한 특징점을 효율적으로 추출하면서도 연산량을 줄일 수 있는 기법이 필요하다.

본 논문에서는 의미 있는 특징점을 이용한 고속 이미지 스티칭 기법을 확장하였다[11]. 선별된 특징점을 통해 기술자의 차원을 축소하여 연산량을 감소시키고, 방향 윈도우 크기를 확장하여 정확한 방향 정보를 추출 함으로써 빠르면서도 효율적인 이미지 스티칭할 수 있는 기법을 제안한다.

## 2. 기존의 이미지 스티칭 알고리즘

이미지 스티칭은 컴퓨터 비전 분야에서 활용도가 높은 분야이다. 일반적으로 이미지를 정합하기 위해서는 특징점 기반의 알고리즘을 사용한다. 특징점 기반의 이미지 스티칭은 먼저, 입력 이미지에서 관심점을 찾고 이에 대한 특징점을 추출한다. 관심점은 일반적으로 코너, 블랍(Blob) 및 교차점 등이 될 수 있다. 두 번째는 추출된 관심점 주변을 특징 벡터로 표현하여 특징점 기술자를 생성한다. 생성된 특징점 기술자를 이용하여 입력 이미지와 참조 이미지에서 관심점과 대응되는 정합점을 찾는다. 대응되는 정합점을 찾기 위해서 기술자 사이의 거리 정보를 활용한다. 마지막으로 추출된 정합점을 이용하여 입력 이미지와 참조 이미지를 서로 정합하여 하나의 고해상도 이미지를 생성한다. 이러한 이미지 스티칭 시스템은 Fig. 1과 같은 과정으로 이루어진다.

이미지 스티칭에서 중요한 요소인 특징점 추출을 위해서 사용되는 방법은 일반적으로 FAST(Fast Feature from Accelerated Segment Test), SIFT(Scale-Invariant Feature Transform), PCA-SIFT(Principle Component Analysis-SIFT), SURF(Speeded-Up Robust Features) 및 C-SURF(Clustering-SURF) 등이 사용되고 있다[6][11].

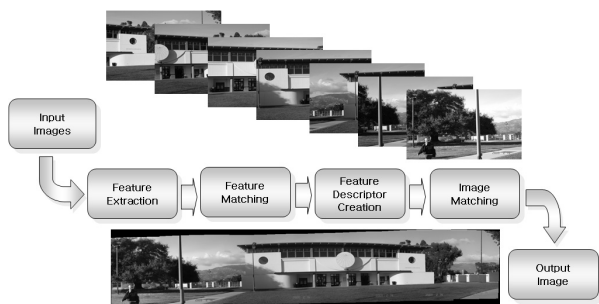


Fig. 1. The image stitching system

FAST는 이미지의 모든 화소에 대하여 주변 16개 화소를 검사하여 모서리 점인지 아닌지를 판별한다[7]. 중심점을 p를 중심으로 테두리 16개의 화소를 검사하여 중심점 화소값보다 밝거나 어두운 픽셀이 연속으로 n개 이상 나타나면 이 점을 모서리로 분류한다. [7]. 따라서 고속으로 모서리를 추출하기 때문에 기존의 방법에 비해 빠른 연산이 가능하며, 변형에 대한 높은 반복성을 지니게 된다. 그러나 노이즈 환경에서 기존의 특징점 추출 방법에 비해 불안정한 특성을 보이는 단점을 가지고 있다.

SIFT는 이미지의 크기, 회전 및 조명에 대한 변화에 강인한 특징점을 추출하는 방법으로 성능이 우수하다[1,8,9]. 그러나 128 차원의 고차원적인 특징 때문에 이미지의 크기가 커질수록 계산해야 하는 데이터량이 증가하게 된다[9]. 데이터량이 증가하게 되면 연산 시간이 증가 되는 문제점이 발생하게 된다. 따라서 기존의 SIFT 방법의 문제점을 개선하기 위해서 SIFT에 PCA 방법을 적용하여 사용한 PCA-SIFT방법이 제시 되었다[9]. PCA는 전체 이미지의 데이터 집합 요소들의 데이터의 평균값, 데이터 표준편차 값과 차원의 데이터들의 공분산의 통계적 성질을 이용한 2차 통계적 방법이다[10]. 전체 이미지의 데이터를 분산이 큰 몇 개의 고유방향(Eigen Vector)에 대한 축으로 선형 투영시켜서 데이터의 차원을 줄이는 방법으로, 128차원을 사용하는 SIFT 특징점 기술자보다 1/4 적은 차원으로 특징점 기술자를 구성하기 때문에 데이터량이 많아져도 연산량이 증가되지 않아 SIFT 보다 연산량 및 오차 발생률을 줄일 수 있는 장점을 가지게 된다.

SURF는 이미지에서 크기 및 회전 등의 왜곡에 강인한 특징점을 추출하고 추출된 특징점의 기술자들을 생성하는 방법이다[8]. 일반적으로 SIFT 방법이 강인한 특징점을 추출하지만 알고리즘의 복잡도가 높고, 연산량이 많아서 특징 추출 속도가 느린 단점을 가지고 있다. 그러나 SURF는 SIFT 방법과 비교하여 특징점 추출 측면에서는 유사한 성능을 보이지만, 속도적인 측면에서 크게 향상된 성능을 보인다. 이와 같은 방법으로 특징점을 추출하게 되면 크기 및 회전 등의 왜곡에 강한 특징을 갖게 된다.

C-SURF는 이미지에서 불필요한 특징점을 제거하여 효율적인 특징점 기술자를 생성하는 방법이다[11]. 이미지에서 군집되지 않은 특징점은 잡음 및 오류 등으로 간주되므로 클러스터링 필터를 이용하여 불필요한 특징점을 제거함으로써 특징점 기술자를 생성시에 연산량을 줄일 수 있다.

기존의 특징점 추출 방법은 다양한 환경에서 우수한 성능을 보이고 있다. 그러나 실시간 응용성을 갖기에는 계산 복잡도가 높기 때문에 연산량이 증가하게 되어 특징점 추출 시스템의 성능 저하를 유발하게 된다. 이를 극복하기 위해서 빠른 연산 속도와 우수한 특징점 추출을 가능하게 하는 알고리즘 개발이 필요하다.

## 3. 제안하는 이미지 스티칭 알고리즘

이미지 스티칭에서 중요한 요소는 정확하면서도 연산량이

적은 특징점 기술자를 생성하는 것이다. 이를 위해서 추출된 특징점에서 잡음 등과 같은 불필요한 특징점을 분류하고 기술자를 생성하여 대응되는 정합점을 찾는다[11]. 이때 주변 환경의 변화 및 스케일 변화에 강인하면서도 효율적인 특징점 기술자 생성이 필요하다. 본 논문에서는 특징점 기술자의 차원을 줄여 속도를 향상 시켰다. 이때 차원 축소로 인하여 손실되는 정보는 방향 윈도우의 크기를 확장함으로써 보다 많은 정보를 활용하여 주 방향을 계산하여 정보 손실을 감소 시키는 알고리즘을 제안하고 있다. 마지막으로 추출된 특징점에 정합 알고리즘을 적용하여 효율적인 이미지 스티칭 기법을 제안하고 있다.

3.1 특징점 추출

SURF방법에서는 특징점을 추출하기 위해 2차 미분을 근사화한 박스 필터를 이용한 회전 연산을 반복적으로 수행한다[12]. 박스 필터의 회전 연산은 필터 내부에 존재하는 화소값의 합을 계산하는 작업을 수행한다. 이때 적분 이미지(Integral Image)를 사용하여 빠른 속도로 회전 연산을 수행할 수 있다. 따라서 스케일 공간 생성시 소요되는 연산량을 감소시킬 수 있다[13]. 또한 SURF 기반의 특징점 추출에서는 고속으로 특징점을 추출하기 위해 헤이시안 행렬 기반의 검출기를 사용한다[12,14]. 헤이시안 행렬 기반의 검출기는 특징점 추출의 정확성이 높다[15]. 원본 이미지의 스케일을 변화시키지 않고 박스 필터의 크기만 증가시켜 다양한 스케일 변화에 강인한 특징점을 추출할 수 있다 [12,14]. 그러나 추출된 특징점을 이용하여 이미지 스티칭을 수행하면 잡음 등으로 인하여 불필요한 연산을 수행하게 된다. 불필요한 연산을 줄이기 위해서 추출된 특징점들로부터 필요한 특징점만 분류할 필요가 있다. 필요한 특징점을 선별하기 위해서 추출된 특징점들을 클러스터링 필터를 통과하여 불필요한 특징점을 제거시킨다.[11].

3.2 기술자의 차원 축소 및 성능 향상

이미지 스티칭의 속도를 향상 시키기 위해서는 특징점 기술자의 연산량을 감소시켜야 한다. SURF 알고리즘의 기술자는 64차원으로 성능은 우수하나 실시간으로 사용하기에는 느린 단점을 가지고 있다[12]. 이러한 문제를 해결하기 위해서 본 논문에서는 특징점 기술자의 차원을 축소하여 속도를 향상시킬 수 있다. 그러나 축소된 차원으로 인해서 정보의 손실이 발생하게 되어 정확한 특징점 기술자 생성이 어려워지게 된다. 정보의 손실을 감소시키기 위해서 특징점에 대한 주 방향을 찾는 방향 윈도우의 크기를 확장하여 방향 정보의 손실을 줄일 수 있다. 따라서 본 논문에서는 차원 축소를 인하여 속도가 향상되고 방향 윈도우의 확장으로 정확한 기술자를 생성할 수 있는 향상된 특징점 기술자 생성을 제안한다.

1) 특징점 기술자의 성능 향상

클러스터링 필터를 사용하여 추출된 특징점들이 회전에 강인하면서도 정확한 정보를 갖기 위해 특징점에 대한 방향

정보 추출 과정이 필요하다. 방향 정보 추출 과정은 추출된 특징점 후보군에서 특징점의 주 방향을 찾아야 한다. 특징점의 주 방향은 선택된 특징점들을 중심으로  $6\sigma$  스케일 반경의 영역 내에서 수평( $x$ ), 수직( $y$ ) 방향으로 하 웨이블릿 응답을 계산하여  $dx$ ,  $dy$  를 계산한다. 특징점에 대한 주 방향을 방향 윈도우(Orientation Window)를 사용하여  $360^\circ$  방향으로 검색하여 방향 윈도우 안에서 수직, 수평 성분의 벡터 계산한다. 이때  $60^\circ$  도의 방향 윈도우를 사용하지 않고 보다 많은 정보를 활용하여 주 방향을 계산하기 위해서 방향 윈도우를  $90^\circ$ 로 확장하여 측정하였다. Fig. 2는  $90^\circ$ 로 정의한 방향 윈도우를 통해서 생성된 방향 정보 벡터를 나타내고 있다.

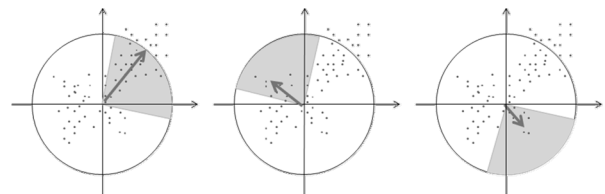


Fig. 2. Orientation Assignment with window size of  $\pi / 2$

이때 수직 수평 성분은 방향 윈도우 범위 내에서 벡터의 합으로 표현되며 가장 큰 벡터를 방향 벡터로 정의한다. 특정 방향을 가리키는  $x$  축,  $y$  축 응답의 합을 구하고 합쳐진 응답은 새로운 벡터로 생성한다. 이 중에서 가장 긴 벡터를 주 방향으로 선택하고, 나머지 벡터들은 사용하지 않는다.

2) 특징점 기술자의 차원 축소

기술자 생성에 앞서 먼저 위치가 결정된 특징점들이 회전에 불변하는 특성을 갖기 위해 특징점에서 방향 정규화를 수행한다. 방향 정규화에서는 주방향을 찾기 위해 탐지된 특징점들을 중심으로 특징점이 찾아진 스케일 공간의 스케일 정보 's'를 사용한다. 특징점 기술자를 생성은 방향 윈도우를 통해 결정된 주 방향과 스케일 정보를 활용하여 특징점을 중심으로 사각형 윈도우 영역을 구성한다. 이때 윈도우의 크기는 15s로 하며, 윈도우의 방향은 주 방향에 따라 위치시킨다. 관심영역을 Fig. 3과 같이 사각형 형태의  $3 \times 3$  부분 영역으로 분할한다.

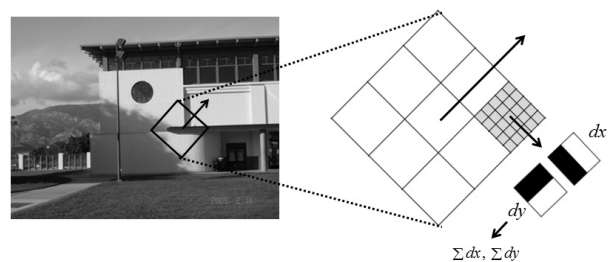


Fig. 3. Reduction of dimension in feature descriptors

각각의 부분영역에서  $5 \times 5$  로 생성된 서브 윈도우에서 규칙적으로 샘플링한 하 웨이블릿 응답을 계산한다. 윈도우를  $3 \times 3$  나눈 부분 영역에 수식 (1)과 같이 두 개의 특징 벡터를 생성하여 18차원( $3 \times 3 \times 2$ )의 벡터를 생성한다.

$$V_{sub} = [\sum dx, \sum dy] \quad (1)$$

이때 응답은 수직 방향을  $dx$ , 수평 방향을  $dy$  라고 할 때, 각각의 부분영역에 대한 응답의 합을 구한다. 위와 같은 과정을 거쳐 이미지의 크기, 회전 및 명암 변화에 강인한 특성을 갖는 특징 벡터를 생성하고 주 방향을 설정하여 정확하면서도 빠른 특징점 기술자를 생성할 수 있다.

#### 4. 실험 및 결과

본 논문에서 빠른 특징점 추출 및 정합을 이용한 효율적인 이미지 스티칭 기법에 대한 성능 평가를 위해서 다양한 이미지 데이터 세트(Image Data Set)를 사용하였다. 이미지 데이터 세트는 VisualSize에서 제공하는 이미지를 사용하였다[16]. 또한 사용된 시스템은 i5 2.3GHz 코어와 4GB RAM의 사양을 가지고 있는 Windows 7 환경에서 Visual Studio 2008을 이용하여 구현하였다.

Fig. 4~Fig. 9은 기존의 SURF 및 Clustering-SURF (C-SURF) 방법과 제안한 방법으로 이미지 스티칭한 결과 이미지를 보여주고 있다.

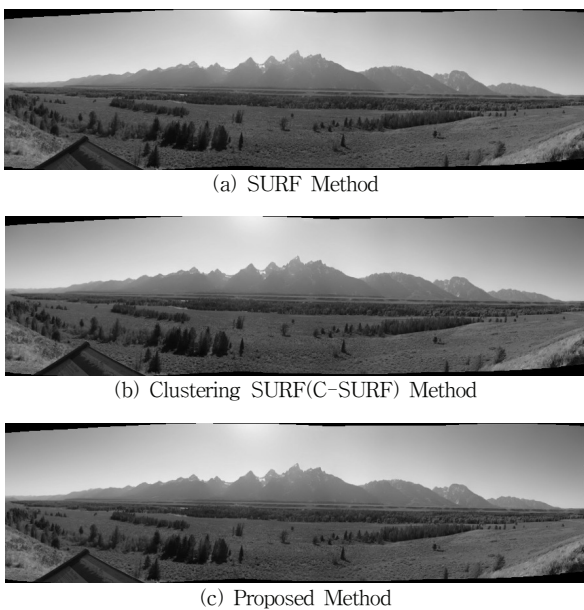


Fig. 4. A comparison between the previous methods and proposed method in No.1

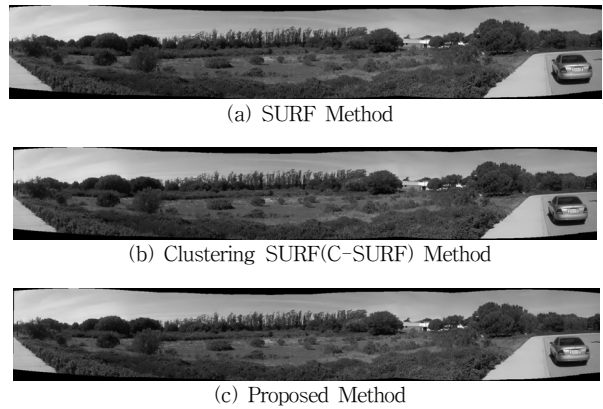
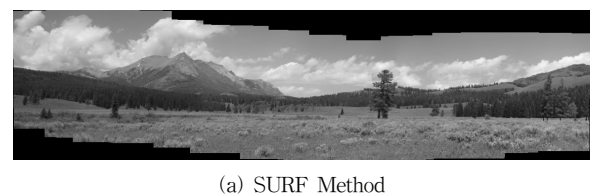


Fig. 5. A comparison between the previous methods and proposed method in No.2



Fig. 6. A comparison between the previous methods and proposed method in No.3



(a) SURF Method

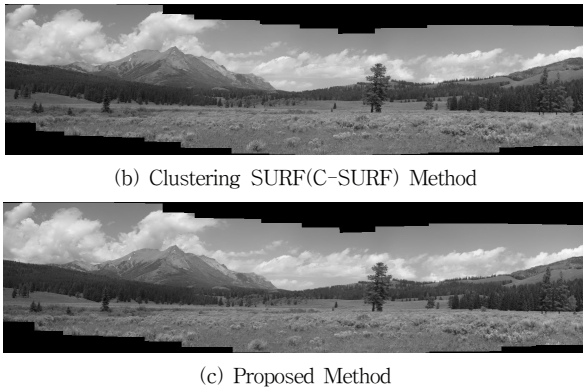


Fig. 7. A comparison between the previous methods and proposed method in No.4

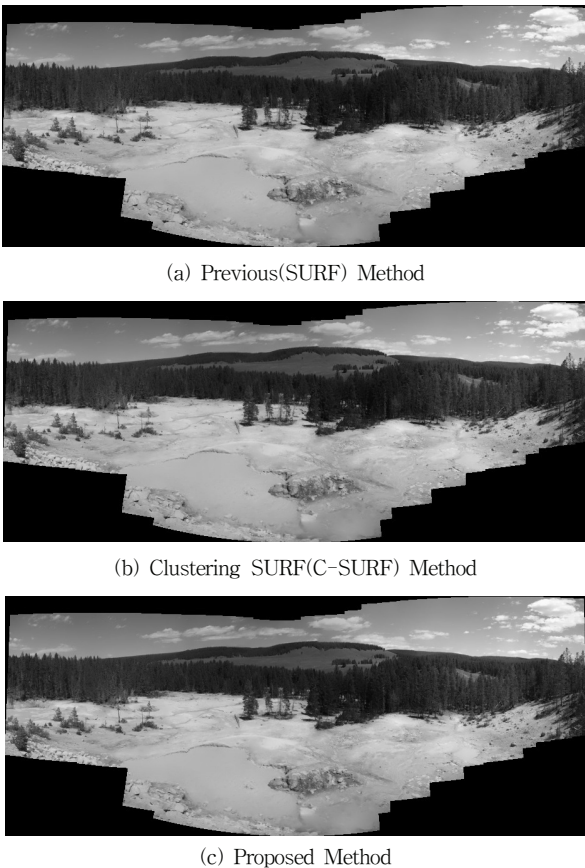


Fig. 8. A comparison between the previous methods and proposed method in No.5

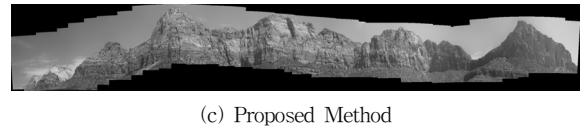
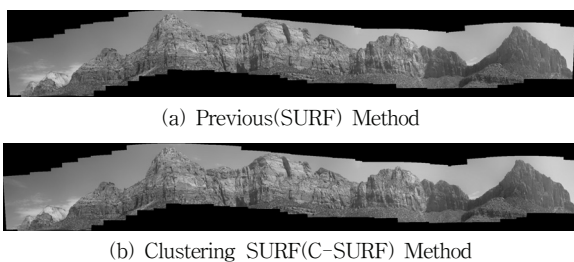


Fig. 9. A comparison between the previous methods and proposed method in No.6

이미지 스티칭 결과에 대한 정확한 수치적 차이를 보여주기 위해 기존의 방법인 SURF(a)에 대해 Clustering-SURF 방법(b)과 제안한 방법(c)의 PSNR(Peak Signal to Noise Ratio)을 측정하였다. 본 논문에서 사용하는 영상은 RGB 컬러 영상으로 각각의 R, G, B에 대하여 PSNR을 계산하고 이에 대한 평균값을 사용하였다. 수식 (2)은 PSNR을 계산하는 방법을 나타내고 있으며 수식(3)는 평균제곱오차를 나타내고 있다.

$$PSNR_{dB} = 10 \log_{10} \frac{(2^b - 1)^2}{MSE} \quad (2)$$

$$MSE = \frac{1}{N} \sum_{i=0}^{n-1} (X_i - Y_i)^2 \quad (3)$$

여기서  $b$ 는 영상의 비트량으로 8비트를 사용한다. 또한  $X_i$ 와  $Y_i$ 는 입력 영상과 대응 영상의  $i$  번째 데이터를 표현하고 있다. 실험 결과에서 알 수 있듯이 기존의 SURF방법에 대해 Clustering-SURF 방법보다는 제안한 방법의 PSNR이 향상된 것을 알 수 있다. Table 1은 기존의 알고리즘과 제안한 알고리즘과의 PSNR을 비교한 결과를 나타내고 있다.

Table 1. A comparison for PSNR between the previous methods and proposed method (dB)

	(a) vs (b)	(a) vs (c)
No.1	24.26	27.44
No.2	21.34	22.61
No.3	23.08	25.15
No.4	22.20	23.17
No.5	24.50	26.92
No.6	22.21	25.41

또한 기존의 알고리즘과 제안한 알고리즘의 속도 향상을 보여주기 위해서 특징점 추출에 따른 특징점 추출 시간, 특징점 정합 시간을 서로 비교하였다. 뿐만 아니라 정확한 이미지 스티칭의 성능을 평가하기 위해서 동일한 데이터 세트로 이미지를 정합하는데 소요되는 전체 이미지 스티칭 시간을 측정하였다.

Table 2에서 알 수 있듯이 기존의 알고리즘과 비교하여 제안한 알고리즘이 향상된 성능을 보여주고 있다. 실험 이미

Table 2. A comparison for FT, MT and ST between the previous methods and proposed method (sec)

		No.1	No.2	No.3	No.4	No.5	No.6
SURF	FT	8.0	13.6	10.5	4.5	13.3	14.6
	MT	28.4	38.8	47.7	7.2	70.8	85.0
	ST	126.9	95.8	174.7	23.7	249.3	211.1
C-SURF	FT	9.3	18.9	14.5	6.7	19.1	20.0
	MT	19.8	21.9	19.2	3.6	36.1	40.9
	ST	102.3	80.5	124.4	21.4	194.5	158.7
Proposed	FT	7.4	15.81	12.3	5.5	16.4	16.8
	MT	16.6	22.17	19.3	3.5	35.2	40.8
	ST	100.1	78.6	122.3	20.4	190.6	156.0

지의 평균 특징점 추출 시간(FT)은 SURF, C-SURF 및 제안한 알고리즘에서는 각각 10.8초, 14.8초 및 12.4초로 나타난다. 따라서 특징점 추출 시간은 SURF 보다는 15.1% 증가되지만 C-SURF 보다는 16.1% 감소되어 C-SURF 보다는 향상된 것을 알 수 있다. 또한 정합점 추출 시간(MT)은 SURF, C-SURF 및 제안한 알고리즘에서는 각각 46.3초, 23.5초 및 22.9초로 나타난다. 따라서 정합점 추출 시간은 SURF 보다는 50.5% 감소하고 C-SURF 보다는 2.8% 감소되어 전체적으로 향상된 것을 알 수 있다. 그리고 이미지 스티칭 시간(ST)은 SURF, C-SURF 및 제안한 알고리즘에서는 각각 146.9초, 113.6초 및 111.3초로 나타난다. 따라서 이미지 스티칭 시간은 SURF 보다는 24.2% 감소하고 C-SURF 보다는 2.0% 감소되어 전체적으로 향상된 것을 알 수 있다.

실험 결과에서 나타나듯이 전체적인 이미지 스티칭 시간이 SURF 보다는 크게 향상되었지만, C-SURF 보다는 약간 향상되었다. 그러나 PSNR 에서는 C-SURF 보다 향상되어 제안한 알고리즘이 기존의 방법보다 결과 이미지의 개선뿐만 아니라 속도적인 측면에서도 향상된 것을 알 수 있다.

### 5. 결론 및 향후 연구 과제

실험 결과를 보면 기존의 C-SURF 방법보다 제안한 방법이 특징점 기술자의 차원을 축소 시켜 연산량을 줄일 수 있었으며, 또한 방향 윈도우의 크기를 확장함으로써 정확한 정합점을 찾을 수 있어 제안한 알고리즘이 빠르면서도 효율적인 이미지 스티칭을 수행함을 알 수 있었다.

향후 연구 과제로는 CCTV등과 같은 감시 장비로부터 받아들이는 이미지를 스티칭하여 실시간으로 객체 추출, 추적 및 감시하는 기법에 대한 연구가 요구된다.

### 참 고 문 헌

[1] R. Szeliski, "Image alignment and stitching: A tutorial", Preliminary draft, Jan., 2005.

[2] P. F. McLauchlan, A. Jaenicke, "Image mosaicing using sequential bundle adjustment", Image Vision Computer, Vol.20, No.9-10, pp.751-759, Aug., 2002.

[3] J. P. Snyder, "Geometry of mapping satellite", PE&RS, Vol.48, No.10, pp.1593-1602, 1982.

[4] C. Y. Chen and R. Klette, "Image stitching: comparisons and new techniques", Computer Analysis of Images and Patterns, pp.615-622, 1999.

[5] E. Vincent, R. Laganieri, "An empirical study of some feature matching strategies", Vision Interface, pp.139-145, 2002.

[6] L. Juan and O. Gwun "A Comparison of SIFT, PCA-SIFT and SURF", IJIP, Vol.3, Issue 4, 2009.

[7] E. Rosten and T. Drummond, "Machine Learning for High-Speed Corner Detection", Computer Vision - ECCV 2006, Lecture Notes in Computer Science, Vol.3951, pp. 430-443, 2006.

[8] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, Vol.60, No.2, pp.91 - 110, 2004.

[9] Y. Ke and R. Sukthankar. "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors", Proc. Conf. Computer Vision and Pattern Recognition, pp.511-517, 2004.

[10] K. Mikolajczyk, C. Schmid, "A performance evaluation of local descriptors", PAMI27, pp.1615 - 1630, 2005.

[11] H. Ahn, S. Rhee, "Fast Image Stitching Based on Improved SURF Algorithm Using Meaningful Features", The KIPS Transaction:Part B, Vol.19B, No.2, pp.93-98, April, 2012.

[12] H. Bay, T. Tuytelaars. "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol.110, No.3, pp.346-359, 2008.

[13] M. Brown, D. Lowe, "Invariant features from interest point groups", In BMVC, 2002.

[14] T. Schenk, Digitalphotogrammetry, Terrascience, 1999.

[15] P. Viola, M. Jones, "Rapid object detection using a boosted cascade of simple feature". In CVPR(1), pp.511-518, 2001.

[16] Image Data Set, <http://www.visualsize.com/>

### 이 상 범

e-mail : sbrhee@dankook.ac.kr

1974년 연세대학교 전자공학과(공학사)  
 1978년 서울대학교 전자공학과(공학석사)  
 1986년 연세대학교 전자공학과(공학박사)  
 2000년 미국 SanJose대학 컴퓨터공학과  
 객원교수



2005년 한국정보처리학회 회장  
 1979년~현 재 단국대학교 응용컴퓨터공학과 교수  
 2010년~2012년 단국대학교 정보미디어대학원 원장  
 2012년~현 재 단국대학교 공과대학 학장  
 관심분야: 컴퓨터구조, 패턴인식, 영상처리, 임베디드 시스템