

# 의미 있는 특징점을 이용한 향상된 SURF 알고리즘 기반의 고속 이미지 스티칭 기법

안 호 창<sup>†</sup> · 이 상 범<sup>††</sup>

## 요 약

최근 고성능 디지털 카메라의 발전으로 영상을 쉽게 획득하고, 많은 곳에서 활용하고 있다. 그 중에서 영상을 정합하여 사용하는 이미지 스티칭 방법에 대한 많은 연구가 진행되고 있다. 이미지 스티칭은 위성이나 정찰기 등의 군사용 목적 및 의료 영상, 지도 등의 컴퓨터 비전 분야 등에서 활용할 수 있다. 본 논문에서는 영상에서 특징점을 추출하고 이를 정합하는 과정에서 의미 있는 특징점을 분류하고 이를 사용하는 향상된 SURF 알고리즘 기반의 고속 이미지 스티칭 방법을 제안한다. 여러 장의 영상에서 정합되는 부분을 찾기 위해 각각의 영상에서 특징점을 추출한다. 각각의 영상에서 추출된 특징점들 중 잡음 등과 같은 오류를 제거하여 의미 있는 특징점을 분류하고 이를 정합하여 연산 처리량을 줄임으로써 이미지 스티칭의 속도를 향상시켰다. 실험 결과 특징점 정합 속도 및 이미지 스티칭 속도가 기존의 알고리즘 보다 빠르면서도 자연스러운 영상을 생성할 수 있었다.

키워드 : 이미지 스티칭, SURF

## Fast Image Stitching Based on Improved SURF Algorithm Using Meaningful Features

Hyochang Ahn<sup>†</sup> · Sang-Burm Rhee<sup>††</sup>

## ABSTRACT

Recently, we can easily create high resolution images with digital cameras for high-performance and make use them at variety fields. Especially, the image stitching method which adjusts couple of images has been researched. Image stitching can be used for military purposes such as satellites and reconnaissance aircraft, and computer vision such as medical image and the map. In this paper, we have proposed fast image stitching based on improved SURF algorithm using meaningful features in the process of images matching after extracting features from scenery image. The features are extracted in each image to find out corresponding points. At this time, the meaningful features can be searched by removing the error, such as noise, in extracted features. And these features are used for corresponding points on image matching. The total processing time of image stitching is improved due to the reduced time in searching out corresponding points. In our results, the processing time of feature matching and image stitching is faster than previous algorithms, and also that method can make natural-looking stitched image.

Keywords : Image stitching, SURF

## 1. 서 론

최근 고성능의 디지털 카메라의 보급으로 일상에서 손쉽게 영상을 획득하고, 여러 분야에서 이를 활용하고 있다. 하지만 디지털 카메라를 이용하여 넓은 지역의 자연 경관 등을 한 장의 영상으로 담기에는 부족한 점이 많다[1]. 일반적

으로 넓은 지역의 경관을 획득하는 방법은 어안렌즈나 초광각 렌즈와 같은 넓은 화각 영역을 갖는 렌즈를 가지고 있는 특수한 카메라를 이용하는 방법과 여러 장의 영상을 정합하여 하나의 영상으로 생성하는 방법이 있다[2,3,4]. 이미지 스티칭은 제한된 화각의 카메라를 사용하여 획득한 여러 장의 영상에서 중첩된 부분을 서로 정합하여 한 장의 영상으로 나타내는 방법이다[5].

이미지 스티칭에서 가장 중요한 요소는 특징점 추출 및 서로 대응되는 정합점을 찾는 것이다[6,7]. 추출된 정합점을 이용하여 각각의 영상을 하나의 영상으로 정합시킬 수 있는

※ 이 연구는 2010년도 단국대학교 대학연구비의 지원으로 연구되었음.

† 준 회 원 : 단국대학교 전자컴퓨터공학과 공학박사

†† 정 회 원 : 단국대학교 응용컴퓨터공학과 교수

논문접수: 2012년 2월 21일

수정일: 1차 2012년 3월 26일, 2차 2012년 4월 18일

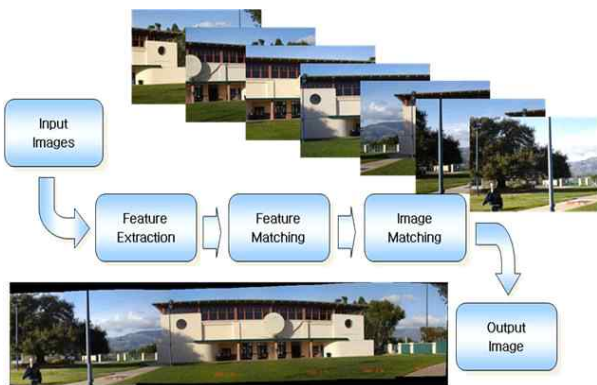
심사완료: 2012년 4월 24일

상호 관계를 도출하고 이를 이용하여 이미지 스티칭 과정을 수행하게 된다.

정확한 특징점 추출은 정합하는 과정에 연산량에 영향을 많이 주기 때문에 이미지 스티칭의 연산 속도를 좌우하게 된다. 영상에서 많은 특징점이 추출되면 풍부한 정보로 인하여 정확한 이미지 스티칭을 수행할 수 있다. 그러나 특징점에 대한 기술자 생성 및 생성된 기술자를 바탕으로 수행되는 영상 정합 과정에서 서로 비교되는 정합점이 많아지므로 연산량이 크게 증가하여 처리 속도가 저하 되는 문제점이 발생한다. 반면에 추출된 특징점의 개수가 적으면 빈약한 정보로 인하여, 특징점 기술자의 생성이 어렵고 서로 대응되는 특징점을 찾기 어려워지므로 정확한 정합을 수행할 수 없는 문제점이 발생한다. 본 논문에서는 추출된 특징점에 대해 클러스터링 필터를 적용하여 불필요한 특징점을 제거함으로써 의미 있는 특징점만을 이용한 향상된 SURF 기반의 고속 이미지 스티칭 기법을 제안한다.

## 2. 기존의 이미지 스티칭 알고리즘

이미지 스티칭은 원하는 경치 및 지역을 고해상도의 여러 장의 영상으로 분할하여 생성하고 중첩된 부분을 정합하여 한 장의 고해상도 영상으로 생성하는 방법이다[4]. 이러한 이미지 스티칭 시스템은 (그림 1)과 같은 과정으로 이루어진다.



(그림 1) 이미지 스티칭 시스템 구성

이미지 스티칭에서 특징점을 추출하기 위해서 사용되는 방법은 SIFT, PCA-SIFT 및 SURF 등이 가장 많이 사용되고 있다[8]. SIFT 방법은 Lowe에 의해 제안 되었으며, 객체의 크기, 회전, 투영 변환 및 조명 변화 등에 강인함을 갖는 지역적 특성 벡터들의 집합으로 변환하는 방법이다[9]. 영상에서 특징이 되는 특징점을 찾기 위해 영상을 피라미드 형식으로 스케일 공간을 생성하고 이를 가우시안 함수를 적용하여, 차분 영상에서 최대값 또는 최저값을 추출하여 후보 특징점으로 선정한다. 이렇게 얻어진 다양한 레벨의 스케일 영상에서 DoG(Difference of Gaussian) 함수를 계산한다. 다양한 사이즈와 스케일 변화에 대해 DoG를 찾기 때문에 SIFT의 특징인 객체의 크기, 회전, 투영 변환 및 조명 변화

등에 강인한 불변의 특징을 가질 수 있다. 반면에 SIFT 방법은 고차원적인 특징 때문에 영상의 크기가 커질수록 계산되는 데이터량이 많아지게 된다. 데이터량이 많아지면 연산량이 크게 늘어나게 되어 수행 시간이 증가하는 문제점이 발생된다. 이러한 고차원적인 문제를 해결하기 위해 Sukthankar는 PCA-SIFT 방법을 제안하였다. PCA-SIFT 방법은 고차원으로 인한 연산량 문제를 해결하기 위해 고차원 특징점을 저차원적 특징으로 줄이는 PCA기법을 이용하여 특징점 기술자 벡터의 차원을 줄이는 방법이다[10]. PCA는 전체 영상의 데이터를 분산이 큰 몇 개의 고유 벡터(Eigen Vector)에 대한 축으로 선형 투영시켜 데이터의 차원을 줄이는 방법으로, 서로 다른 클래스의 차원을 축소시켜 간단하게 표현할 수 있는 체계적이고 실용적인 방법으로 객체 인식, 얼굴 인식 분야 등에서 많이 활용되고 있다.

실시간 응용성을 갖기 위한 기존의 방법들은 계산 복잡도가 높아 연산량이 많기 때문에 특징점 추출 과정에서 성능 저하를 유발하게 된다. 이러한 단점을 극복하기 위해서 빠른 연산 속도와 우수한 특징점 추출을 가능하게 하는 FAST(Feature from Accelerated Segment Test)가 제안 되었다[11]. FAST는 영상의 모든 화소에 대하여 주변 16개 화소를 검사하고 모서리 점인지 아닌지를 판별하여 효과적으로 영상의 특징점을 추출할 수 있다. Bay에 의해 제안된 SURF 기술은 SIFT가 가지고 있는 추출 및 정합 단계에서 발생하는 연산의 복잡성을 줄여 빠르게 특징점 및 특징 기술자를 추출하고 정합을 수행할 수 있도록 하는 방법이다. SURF의 핵심은 적분 영상(Integral Image)을 사용하여 영상에 적용되는 회선(Convolution) 작업을 빠르게 수행할 수 있도록 하고, 근사화환 헤이시안 행렬식을 이용하여 빠르게 특징점 기술자를 생성함으로써 크기 및 회전 등의 왜곡에 강인하면서도 효과적인 특징점 추출 결과를 얻을 수 있다[12]. 그러나 SURF 방법도 실시간으로 이미지 스티칭을 수행하기에는 연산량이 상당히 많다. 따라서 연산량을 줄이면서도 정확한 이미지 스티칭을 수행할 수 있는 알고리즘 개발이 필요하다.

## 3. 제안하는 이미지 스티칭 알고리즘

특징점 기반의 이미지 스티칭에서 가장 중요한 부분은 정확하면서 필요한 특징점만을 추출하는 것이다. 영상에서 불필요한 특징점을 제거하고 의미 있는 특징점만을 이용하여 특징점 정합과정을 수행하는 것이 빠른 이미지 스티칭을 수행하기 위해서 필요하다. 영상에서 의미 있는 특징점들은 대부분 군집되어 있는 특성을 가지고 있다. 반대로 군집 지역에서 벗어난 영역에서 추출된 특징점들은 잡음 등으로 분류되거나 정합과정에서 불필요한 연산을 가중 시키는 의미 없는 특징점으로 간주할 수 있다. 본 논문에서는 주변 환경의 변화에 강인하면서도 비중이 높은 특징점들을 선별하여 정합점을 추출할 수 있는 적응적 클러스터링 필터를 적용한 고속 이미지 스티칭 방법을 제안하였다.

### 3.1 적분 영상

SURF방법에서는 특징점을 추출하기 위해 박스 필터를 이용한 회선 연산을 반복적으로 수행한다. 박스 필터의 회선 연산은 필터 내부에 존재하는 화소값의 합을 구하는 작업을 수행하는 것이다. 이때 적분 영상(Integral Image)을 생성하면 박스 필터의 크기와 상관없이 빠른 속도로 회선 연산을 수행 할 수 있어, 스케일 공간 생성시 처리되는 연산량을 감소시킬 수 있다[13]. 적분 영상을 사용하면 영상에서 임의의 크기의 부분 영역을 선택하더라도 4번의 연산을 수행하면 선택된 영역의 모든 화소의 합을 구할 수 있다. 먼저 입력 영상의 사각형 영역의 특정 위치에서 모든 화소값을 더하면 적분 영상을 생성할 수 있다.

$$I_{int}(X) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \quad (1)$$

수식 (1)에서  $I_{int}(X)$ 는  $X = (x, y)^T$ 의 위치에 대한 적분 영상을 나타낸다. 또한  $I(i, j)$ 는 입력 영상에서 화소값을 나타낸다. 적분 영상은 연산 영역에 종속적이지 않기 때문에 큰 필터 사이즈를 반복해서 계산하는데 매우 효과적으로 사용할 수 있는 장점을 가지고 있다.

### 3.2 특징점 추출

SURF 기반의 특징점 추출에서는 고속으로 특징점을 추출하기 위해 가우시안 필터를 사용하지 않고 헤이시안 행렬 기반의 추출기를 사용한다[14]. 헤이시안 행렬 기반의 추출기는 가우시안 2차 미분의 극값을 추출하여 특징점을 추출한다. 이때, 헤이시안 행렬 기반의 추출기에서 가우시안 2차 미분을 사용하지 않고 근사화된 박스 필터를 사용한다. 따라서 근사화된 헤이시안 행렬 기반의 추출기를 사용하여 원본 영상의 스케일은 변화시키지 않고 박스 필터의 크기만 증가시켜 다양한 스케일 변화에 강인한 특징점을 추출할 수 있다. 헤이시안 행렬은 2차 미분된 가우시안 필터와 회선 연산을 통하여 얻을 수 있으며, 수식 (2)와 같이 나타낼 수 있다[14,15].

$$H(X, \sigma) = \begin{bmatrix} L_{xx}(X, \sigma), & L_{xy}(X, \sigma) \\ L_{xy}(X, \sigma), & L_{yy}(X, \sigma) \end{bmatrix} \quad (2)$$

$$L_{xx}(X, \sigma) = I(x, y) * \frac{\partial^2}{\partial x^2} g(\sigma) \quad (3)$$

여기서  $L_{xx}(X, \sigma)$ 는 입력 영상의  $X = I(x, y)$ 인 점에서  $\sigma$ 의 스케일을 갖는  $x$ 방향(수평)의 2차 미분된 가우시안 필터와 입력 영상의 회선 연산을 의미한다. 또한  $L_{xy}(X, \sigma)$ 과  $L_{yy}(X, \sigma)$ 는 각각  $xy$ 방향(대각)과  $y$ 방향(수직)으로 2차 미분된 가우시안 필터와 입력 영상의 회선 연산을 나타낸다. 그러나 가우시안 필터는 연산량이 많아 수행 속도가 느리다는 단점을 가지고 있다. SURF에서는

SIFT가 가지고 있는 연산량 증가 문제점을 해결하고자 박스 필터를 사용하여 가우시안 2차 미분의 회선 연산을 근사화한 값을 적용한다[16]. 따라서 근사화 시킨 헤이시안 행렬은 다음 수식 (4)와 같이 나타낼 수 있다.

$$H(X, \sigma) = \begin{bmatrix} D_{xx}(X, \sigma), & D_{xy}(X, \sigma) \\ D_{xy}(X, \sigma), & D_{yy}(X, \sigma) \end{bmatrix} \quad (4)$$

$$\begin{aligned} L_{xx}(X, \sigma) &= I(x, y) \otimes \frac{\partial^2}{\partial x^2} g(\sigma) \\ &\cong I(x, y) \otimes B_f(xx) \\ &= D_{xx}(X, \sigma) \end{aligned} \quad (5)$$

여기서  $D_{xx}$ 는  $L_{xx}$ 를 근사화 시킨 것으로 수식 (5)에서 나타나듯이 입력 영상과 박스 필터를 이용한 회선 연산의 의미한다.  $B_f(xx)$ 는  $x$ 방향의 박스 필터를 의미한다.  $B_f(xx)$ 는 박스 필터를 이용함으로써 연산량을 크게 줄여 같은 크기의 필터를 처리하는 연산에서 우수한 성능을 보인다. 각 스케일에 따라 박스 필터의 크기를 변경시켜 헤이시안 행렬식의 값을 구하면 이미지 피라미드를 생성할 수 있다. 그리고 생성된 이미지 피라미드에서 헤이시안 행렬식의 값이 임계값보다 크고 인접한 영역의 이웃 픽셀보다 클 경우 특징점으로 추출한다.

### 3.3 특징점 클러스터링

고속으로 이미지 스티칭을 하기 위해서 추출된 특징점을 모두 사용하지 않고 불필요한 특징점을 제거하고 의미 있는 특징점만을 분류한다. 의미 있는 특징점을 분류하기 위해서 헤이시안 행렬식으로 추출된 특징점들을 클러스터링 필터를 통하여 군집되지 않은 특징점을 제거한다. 군집되지 않은 특징점은 영상에서 잡음 등으로 간주되므로 정합과정에서 불필요한 연산을 가중시키는 결과를 초래하게 된다. 제안한 클러스터링 필터를 통한 의미 있는 특징점들을 추출하기 위해서는 필터 내에 존재하는 특징점들의 개수 및 밀집도를 계산하여 특징점으로 사용 여부를 결정하고 이를 통하여 특징점 기술자를 생성한다. 수식 (6)과 (7)은 추출된 특징점 중에서 의미 있는 특징점들을 분류하기 위해서 사용된다.

$$s^2 = \frac{1}{N} \sum_{i=1}^N (X_i - \bar{X})^2 \quad (6)$$

$$G = \begin{cases} 1, & N > T_N \text{ and } s > T_s \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

여기서  $s$ 는 특징점 클러스터의 밀집도를 나타내고,  $N$ 은 각 클러스터에 속한 특징점들의 개수를 나타낸다.  $X_i$ 는 특징점들의 좌표  $(x_i, y_i)$ 를 의미하며,  $\bar{X}$ 는 특징점 클러스터의 중심을 의미한다.  $T_N$ 은 클러스터링 필터 내에 존재하는 특징점의 개수에 대한 임계치를 나타내고,  $T_s$ 는 클러스터

링 필터 내에 존재하는 특징점들 간의 밀집도에 대한 임계치를 의미한다.  $G$ 는 특징점 클러스터링 필터를 통한 특징점 선정 여부를 나타낸다.

본 논문에서 특징점을 분류하기 위해서 사용한 클러스터링 필터의 크기는  $21 \times 21$ 을 사용하였으며, 필터내부에 존재하는 특징점 개수는 전체 화소수의 약 1/8인 50개를 임계치로 사용하였다. 또한 필터내에 존재하는 특징점간의 밀집도에 대한 임계치는 7로 설정하여 사용하였다. 이와 같은 방법으로 특징점들에 대해 클러스터링 필터를 이용하면 군집된 특징점을 분류하여 의미 있는 특징점을 선별할 수 있으며 수식 (8)과 같이 표현된다.

$$F_{cluster}(x, y) = f_1, f_2, \dots, f_n \quad (8)$$

여기서  $F_{cluster}(x, y)$ 는 특징점 클러스터링 필터를 사용하여 의미 있는 특징점들을 분류한 집합을 나타내고 있으며,  $f_n$ 은 선정된 특징점들의 화소 위치를 나타낸다.

또한 영상에 복잡한 영상일수록 영상의 에지 성분이 많이 추출되므로 다른 영상에 비해서 특징점을 추출하였을 때는 특징점의 개수가 많아진다. 따라서 클러스터링 필터 내부의 특징점의 개수 및 밀집도가 높아지게 된다. 이러한 영상에서 클러스터 필터를 적용하더라도 제거되는 특징점의 수가 거의 없어 영상의 밀집도가 높다. 결과적으로 특징점의 개수가 임계치보다 많으면 클러스터링 필터를 통한 특징점 분류가 의미 없어질 뿐만 아니라, 오히려 특징점을 분류하기 위한 연산만을 가중시키게 된다. 반대로 단순한 영상에서는 전체 특징점의 개수가 적고 밀집도 또한 낮기 때문에 클러스터 필터를 통한 회선 연산을 수행하면 의미 있는 특징점도 제거되어 정보가 많이 없어지게 될 수 있다. 따라서 영상 복잡도에 따라 발생할 수 있는 문제점을 해결하기 위해서 특징점이 많을 때와 적을 때에 대한 임계치를 적용하여 적응적으로 클러스터 필터를 이용한다. 적응적으로 클러스터링 필터를 이용하면 특징점이 많은 경우에는 불필요한 연산을 수행하지 않기 때문에 연산의 속도를 향상 시킬 수 있고, 반대의 경우에는 영상 정보의 손실을 줄일 수 있어 정확한 이미지 스티칭을 수행할 수 있으며 수식(9)와 같이 표현된다.

$$Filter_{Adapt} = \begin{cases} 1, & T_{min} < img_f < T_{max} \\ 0, & otherwise \end{cases} \quad (9)$$

여기서  $img_f$ 는 영상에서 추출된 특징점의 개수를 의미한다.  $T_{min}$ 과  $T_{max}$ 은 실험 데이터 세트의 영상들에서 평균 특징점 개수를 측정하고 측정된 평균 특징점의 개수의 분포를 바탕으로 평균 특징점의 개수의 상위 20%와 하위 20%를 나타낸다. 이러한 방법을 통해서 추출된 의미 있는 특징점만을 가지고 특징점 정합을 위해 특징점 기술자를 생성한다.

#### 4. 실험 및 결과

본 논문에서 제안한 방법을 적용한 이미지 스티칭 방법에 대하여 실험하기 위해 사용된 시스템은 코어 i5 2.3GHz, 메모리 4GB RAM을 사용하였으며, 운영체제는 Windows 7 환경에서 Visual Studio 2008을 이용하여 구현하였다. 실험을 위해 사용된 영상 데이터 세트(Image Data Set)는 VisualSize에서 제공하는 영상을 사용하였다[17]. 영상에 따라 자연 경관 및 도심 경관, 단순한 영상 및 복잡한 영상 등으로 구성되어 있다. (그림 2)~(그림 6)은 기존의 SURF방법과 제안한 방법으로 이미지 스티칭한 영상을 보여주고 있다.



(a) Previous(SURF) Method



(b) Proposed Method

(그림 2) 기존의 알고리즘과 제안한 알고리즘의 이미지 스티칭 영상 비교(No.1)



(a) Previous(SURF) Method



(b) Proposed Method

(그림 3) 기존의 알고리즘과 제안한 알고리즘의 이미지 스티칭 영상 비교(No.2)



(a) Previous(SURF) Method



(b) Proposed Method

(그림 4) 기존의 알고리즘과 제안한 알고리즘의 이미지 스티칭 영상 비교(No.3)



(a) Previous(SURF) Method



(b) Proposed Method

(그림 5) 기존의 알고리즘과 제안한 알고리즘의 이미지 스티칭 영상 비교(No.4)



(a) Previous(SURF) Method



(b) Proposed Method

(그림 6) 기존의 알고리즘과 제안한 알고리즘의 이미지 스티칭 영상 비교(No.5)

두 영상을 비교하였을 때 정합된 영상이 거의 유사한 결과를 보여 주는 것을 알 수 있다. 정확한 차이를 수치적으로 나타내기 위해서 기존의 방법인 SURF와 제안한 방법의 이미지 스티칭 영상을 PSNR(Peak Signal to Noise Ratio)를 사용하여 비교하였다.

비교 대상은 기존의 SURF 방법으로 특징점 추출에 따른 특징점 추출 시간 및 특징점 정합 시간을 비교하였다. 또한

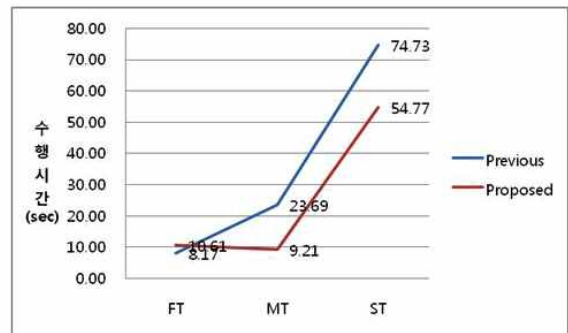
정확한 이미지 스티칭의 성능을 평가하기 위해서 동일한 영상 데이터 세트로 영상을 정합하는 총 수행 시간을 측정하였다.

<표 1> 기존의 알고리즘과 제안한 알고리즘의 PSNR 비교

	No.1	No.2	No.3	No.4	No.5
PSNR	25.90	24.53	27.00	23.81	22.30

<표 2> 기존의 알고리즘과 제안한 알고리즘의 FT, MT 및 ST 비교 (단위 : sec)

		No.1	No.2	No.3	No.4	No.5
Previous	FT	5.15	5.34	7.70	5.55	17.13
	MT	15.33	11.43	16.04	5.36	70.27
	ST	71.32	40.97	55.38	16.11	189.88
Proposed	FT	5.58	7.06	9.39	7.32	23.72
	MT	10.09	6.30	6.30	3.34	20.05
	ST	63.28	35.31	41.02	15.75	118.48



(그림 7) 기존의 알고리즘과 제안한 알고리즘의 이미지 스티칭 평균 소요시간 비교

(그림 7)에서 나타나듯이 기존의 알고리즘과 비교하여 제안한 알고리즘의 성능을 나타내고 있다. No.1 영상에서 제안한 알고리즘은 특징점 추출 시간(FT)에서 0.43초 증가하여 약 8.4% 증가된 결과를 보이지만 정합점 추출 시간(MT)에서는 5.24초 감소하여 약 34.2% 감소된 결과를 보이고 있다. 또한 이미지 스티칭 시간(ST)에서는 8.05초 감소하여 약 11.3% 감소된 결과를 보여주고 있다. 다른 영상들에서도 동일한 결과를 보여주어 평균 특징점 추출 시간은 2.44초 증가하여 약 29.9% 증가된 결과를 보여주지만 정합점 추출 시간과 이미지 스티칭 시간은 각각 14.47초, 19.96초가 감소하여 약 61.1%와 26.7% 감소된 결과를 보여주고 있다.

특징점 추출에서는 추출된 특징점을 다시 클러스터링 필터를 통과하기 때문에 다소 느리다. 그러나 불필요한 특징점을 제거하고 의미 있는 특징점만을 사용하기 때문에 기존의 알고리즘 보다는 특징점 정합 속도 및 이미지 스티칭 속도에서 향상된 성능을 보여 주고 있다.

### 5. 결론 및 향후 연구 과제

실험 결과를 보면 기존의 방법보다 제안한 방법이 특징점을 추출한 후에 특징점의 밀집도 및 개수를 비교하기 때문에 전체 특징점 추출 과정에서 보다 많은 시간을 소비하지만 의미 있는 특징점만을 사용하기 때문에 특징점 정합 속도는 크게 향상되어 전체 영상의 정합 시간을 단축하는 성능을 보여주었다. 일반적으로 특징점을 추출하는 시간 보다는 특징점을 정합하는 시간이 이미지 스티칭에서 비중이 높다. 따라서 특징점 추출에서 추가되는 시간 비용보다 특징점 정합에서 보상 받는 시간 비용이 크기 때문에 전체적인 이미지 스티칭 수행 시간은 줄어들어 제안한 방법이 효과적인 이미지 스티칭을 수행함을 알 수 있었다.

본 연구를 통하여 제안된 특징점 추출 방법을 다른 기법들과 함께 조합하면 보다 향상된 성능을 추구할 수도 있다. 향후 연구 과제로 여러 장의 영상에서 정합된 영상을 기반으로 한 객체 추출, 추적 및 감시하는 기법에 대한 연구가 요구된다.

### 참 고 문 헌

[1] Q. Zhao, "Summary of Virtual Reality", Chinese Science, Jan., 2009.

[2] S. J. Ha, et al., "Panorama mosaic optimization for mobile camera systems", IEEE Trans. Consumer Electronics, Vol.53, No.4, pp.1217-1225, Aug., 2007.

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

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

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

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

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

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

[9] B. Zitova and J. Flusser, "Image registration methods: a survey", Image and Vision Computing 21, pp.977 - 1000, 2003.

[10] C. Tang and Y. Dong, "Automatic Registration based on Improved SIFT for Medical Microscopic Sequence Images", Second International Symposium on Intelligent Information Technology Application, Vol.1, pp.580-583, 2008.

[11] 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.

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

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

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

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

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

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

#### 안 호 창



e-mail : youcu92@dankook.ac.kr  
 2003년 상지대학교 전자계산학과(이학사)  
 2006년 단국대학교 전자컴퓨터공학과  
 (공학석사)  
 2012년 단국대학교 전자컴퓨터공학과  
 (공학박사)

관심분야 : 영상처리, 컴퓨터비전, 임베디드, 모바일

#### 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~현 재 단국대학교 정보미디어대학원 원장  
 관심분야 : 컴퓨터구조, 패턴인식, 영상처리, 임베디드 시스템