

일반논문 (Regular Paper)

방송공학회논문지 제22권 제1호, 2017년 1월 (JBE Vol. 22, No. 1, January 2017)

<https://doi.org/10.5909/JBE.2017.22.1.51>

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

관성 센서 데이터를 활용한 3 DoF 이미지 스티칭 향상

김민우^{a)}, 김상균^{a)†}

Enhancement on 3 DoF Image Stitching Using Inertia Sensor Data

Minwoo Kim^{a)} and Sang-Kyun Kim^{a)†}

요 약

본 논문은 수평을 유지하여 촬영해야 한다는 기존 이미지 스티칭을 이용한 영상 정합 과정의 단점을 극복하기 위하여, 스마트폰의 가속도 센서와 자기장 센서 데이터를 사용하여 3가지 자유도(3 DoF)에 강인한 이미지 스티칭 방법을 제안한다. 이미지를 붙이는 작업인 이미지 스티칭은 크게 이미지 특징점 추출, 추출된 특징점에서 매칭에 필요한 참인 점(inlier)을 선별, 참인 점을 호모그래피(homography) 행렬로 변환, 호모그래피 행렬을 사용하여 이미지를 왜곡(warping), 왜곡된 이미지와 다른 이미지를 합하는 과정으로 이루어져 있다. 본 논문에서는 일반적으로 사용하는 SIFT, SURF 등의 알고리즘뿐만 아니라 MPEG에서 표준화한 MPEG-7 CDVS(Compact Descriptor for Visual Search) 표준의 특징점 추출 알고리즘을 사용하여 이미지의 특징점을 추출한다. 또한 각 알고리즘의 특징점 추출시간, 추출된 특징점 개수, 선별된 참인 점의 개수를 비교하고, 스티칭 정확도를 판단하여 본 연구에서 활용한 데이터에 어느 알고리즘이 효율적인지 살펴본다.

Abstract

This paper proposes a method to generate panoramic images by combining conventional feature extraction algorithms (e.g., SIFT, SURF, MPEG-7 CDVS) with sensed data from an inertia sensor to enhance the stitching results. The challenge of image stitching increases when the images are taken from two different mobile phones with no posture calibration. Using inertia sensor data obtained by the mobile phone, images with different yaw angles, pitch angles, roll angles are preprocessed and adjusted before performing stitching process. Performance of stitching (e.g., feature extraction time, inlier point numbers, stitching accuracy) between conventional feature extraction algorithms is reported along with the stitching performance with/without using the inertia sensor data.

Keyword : Image stitching, video stitching, panoramic video, ultra wide viewing video, sensor-based stitching

a) 명지대학교 컴퓨터공학과(Myongji University Computer Engineering Department)

† Corresponding Author : 김상균(Sang-Kyun Kim)

E-mail: goldmunt@gmail.com

Tel: +82-31-330-6443

ORCID: <http://orcid.org/0000-0002-2359-8709>

※ 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [B0126-15-1013, 퍼즐형 Ultra-wide viewing 공간 미디어 생성 및 소비 기술 개발]

· Manuscript received August 31, 2016; Revised November 15, 2016; Accepted November 15, 2016.

I. 서론

최근 디스플레이의 대형화와 단일 디스플레이를 여러 대 연결한 현재의 디스플레이 환경이 제공됨에 따라, 파노라마 콘텐츠를 더 폭넓게 사용하고 있다. 파노라마 콘텐츠는 거리 파노라마, 관광명소, 숙박, 음식점 등 특정 장소의 정보를 고해상도로 보여주는 포인트 파노라마 등 실생활의 여러 부분에서 사용되고 있으며^[1], 광고, 교육 등 다양한 타 산업 분야에 활용도가 높다.

세계 센서 시장은 2014년 795억 달러에서 연평균 7.9% 성장하여 2019년 1,161억 달러 규모로 TV시장과 비슷한 규모로 성장할 전망이다^[2]. 최근 사물인터넷(Internet of Things)의 발전으로 센서 데이터의 중요성 또한 높아졌다^[3]. 또한, 고속 영상 스티칭을 위한 방법^[4]과 효과적인 영상 특징점 추출 방법^{[5][6]}이 연구되었다.

본 논문은 수평을 유지하여 촬영해야 한다는 기존 영상 정합 제작의 단점을 극복하기 위하여, 관성 센서 데이터를 사용한 3가지 자유도(3 DoF)에 강인한 이미지 스티칭 방법을 제안한다. 또한 일반적으로 사용하는 SIFT, SURF 등의 알고리즘 뿐만 아니라 MPEG에서 표준화한 MPEG-7 CDVS(Compact Descriptor for Visual Search) 표준의 특징점 추출 알고리즘을 사용하여 이미지의 특징점을 추출하여 스티칭을 진행하고, 각 알고리즘의 특징점 추출시간, 추출된 특징점 개수, 선별된 참인 점의 개수를 비교한다.

II. 관련 연구 및 연구 목표

1. 관련 연구

D. G. Lowe는 이미지로부터 스케일(scale)과 회전(rotation)에 강인한 특징들을 추출하는 SIFT(Scale-Invariant Feature Transform) 알고리즘을 제안하였다^[7]. SIFT 알고리즘은 스케일과 회전, 노이즈 영상에 강인하지만 속도가 느리다는 단점이 있다.

H. Bay 외 2명은 여러 개의 이미지로부터 스케일, 조명, 시점 등의 환경변화에 강인한 특징들을 찾는 SURF(Speeded Up Robust Features) 알고리즘을 제안하였다^{[8][9]}. SURF

알고리즘은 SIFT와 견줄만한 성능을 유지하면서 계산 속도를 향상시킨 알고리즘이지만, 그레이공간의 정보만 이용함에 따라 컬러공간에 주어진 많은 유용한 특징을 활용하지 못한다.

P. M. Panchal 외 2명은 기존의 SIFT, SURF 알고리즘의 특징점 개수와 추출속도를 비교하였고, L. Juan과 O. Gwun은 SIFT, PCA-SIFT, SURF 알고리즘의 특징점 개수, 추출 속도 및 스케일, 회전 등을 비교하였다^{[10][11]}.

MPEG에서도 영상의 특징점을 추출 할 수 있는 MPEG-7 CDVS 표준을 개발하였다^{[12][13]}. 이는 두 영상의 전체를 비교하는 것이 아닌, 일부분만 비교를 하여 영상 검색의 성능과 속도를 획기적으로 개선하였다^[14].

이미지 스티칭 시, 추출된 여러 특징점들 중에서 매칭에 적합한 특징점을 선택해야 한다. 매칭에 적합한 특징점을 고르기 위해서는 이상점(outlier)을 제거하는 방법이 필요하다. M. A. Fischler와 R. C. Bolles가 제안한 RANSAC(RANdom SAMple Consensus) 알고리즘은 무작위로 샘플 데이터들을 선별한 후, 최대로 컨센서스(Consensus)가 형성된 데이터를 선택하는 방법이다^[15].

E. Dubrofsky가 제안한 호모그래피(homography) 추정은 이미지 스티칭 시, 2D 변환에 사용된다^[16]. 한 이미지에 존재하는 평면이 다른 이미지로 투영 변환(projective transformation)되는 호모그래피는 이미지 스티칭, 3차원 복원 및 카메라 캘리브레이션 등에 이용되는 중요한 파라미터이다^[17].

2. 연구 목표

본 논문은 관성 센서 데이터를 사용한 3가지 자유도에 강인한 파노라마 이미지를 생성하는 방법을 제안한다. 이미지 스티칭은 크게 이미지 특징점 추출, 추출된 특징점에서 매칭에 필요한 참인 점(inlier)을 선별, 참인 점을 호모그래피(homography) 행렬로 변환, 호모그래피 행렬을 사용하여 이미지를 왜곡(warping), 왜곡된 이미지와 다른 이미지를 합하는 과정으로 이루어져 있다^[18].

3가지 자유도는 사진을 촬영할 때, 수평으로 틀어진 정도(panning), 수직으로 틀어진 정도(tilting), 좌우로 회전한 정도(rolling)를 의미한다. 본 논문에서는 3가지 자유도에 강인한 이미지 스티칭 방법을 위하여 스마트폰의 관성 센서

데이터를 활용한다. 추출된 관성 센서 데이터로 방위각(azimuth), 스마트폰의 경사도(pitch), 좌우 회전각도(roll)을 구하여, 방위각은 수평으로 틀어진 정도의 보정, 스마트폰의 경사도는 수직으로 틀어진 정도의 보정, 좌우 회전각도는 좌우로 회전한 정도를 보정한다. 보정된 이미지를 사용하여 이미지 스티칭을 수행한 결과이미지와 원본 이미지로 이미지 스티칭을 수행한 결과이미지를 비교하여 제안한 방법의 적절성을 평가한다.

III. 관성 센서 데이터를 활용한 이미지 전처리

1. 이미지 및 센서 데이터 수집

관성 센서 데이터를 활용한 이미지 전처리를 수행하기 위하여, 안드로이드 스마트폰의 가속도 센서와 자기장 센서를 사용한다. 가속도 센서는 스마트폰의 x, y, z축 각 방향으로 가해진 가속도를 측정한다. [그림 1]은 안드로이드 스마트폰의 x, y, z축이다.

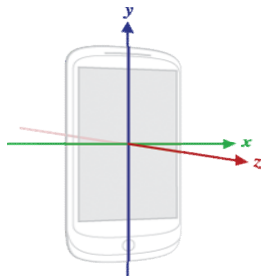


그림 1. 안드로이드 스마트폰의 x, y, z축
Fig. 1. x, y, z axis of Android smartphone

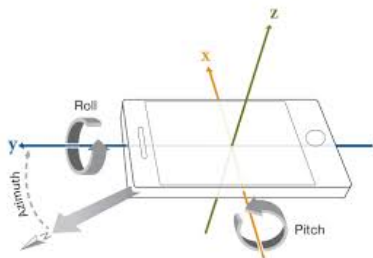


그림 2. 3축 회전각
Fig. 2. Rotation angle of 3-axis

안드로이드의 표준 API를 사용하여, 가속도 센서 데이터와 자기장 센서 데이터로 회전 행렬을 얻는다. 회전 행렬을 사용하여 3개의 축에 해당하는 회전각을 얻을 수 있다. x축 회전각(pitch)은 스마트폰의 경사도를 나타내고, y축 회전각(roll)은 좌우 회전각도, z축 회전각(azimuth)은 방위각을 나타낸다. [그림 2]은 각 축별 회전각에 대한 그림이다.

2. 센서 데이터를 활용한 이미지 전처리

어플리케이션을 사용하여 구해진 x, y, z축의 회전각을 이용하여 원본 이미지를 전처리하기 위해서 스마트폰의 x, y, z축과 이미지의 x, y, z축을 일치시킨다. [그림 3]은 스마트폰과 이미지의 x, y, z축이다.

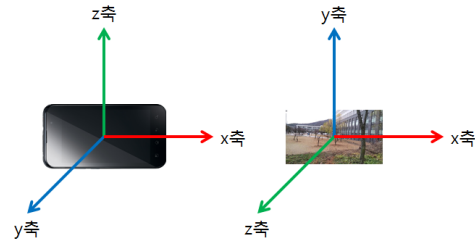


그림 3. 스마트폰과 이미지의 3축
Fig. 3. 3-axis of smartphone and image

[그림 3]의 스마트폰과 이미지는 x축이 같고 y축과 z축은 반대이다. 따라서 x축 회전각을 제외한 y축, z축 회전각을 반대로 적용시킨다. 전처리에 사용되는 회전각은 다음과 같이 구하였다.

$$\begin{aligned}\Delta pitch &= image2.pitch - image1.pitch \\ \Delta roll &= image2.roll - image1.roll \\ \Delta azimuth &= image2.azimuth - image1.azimuth\end{aligned}$$

이미지 스티칭은 사람이 직접 순서대로 이미지를 입력하거나 제한을 가하는 행동이 필요하다^[19]. 본 연구에서는 이미지 입력 시, 두 장의 이미지를 직접 분류하여 왼쪽에 배치되는 이미지(image1)를 먼저 입력하고 오른쪽에 배치되는 이미지(image2)를 나중에 입력한다. 각 이미지를 회전하기 위하여 센서 데이터 차이의 평균을 구하였다.

$$avg_{pitch} = \Delta pitch / NUMBER_OF_IMAGES$$

$$avg_{roll} = \Delta roll / NUMBER_OF_IMAGES$$

$$avg_{azimuth} = \Delta azimuth / NUMBER_OF_IMAGES$$

본 연구에서는 이미지의 3차원 회전을 위하여 3차원 회전 행렬을 사용하였다. 3차원 회전을 위하여 2차원 이미지를 3차원으로 투영하는 행렬을 만든다^[20].

$$P_{2d \rightarrow 3d} = \begin{bmatrix} 1 & 0 & -image.width/2 \\ 0 & 1 & -image.height/2 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3차원 이미지를 회전하기 위하여 각 축에 대한 회전행렬을 구한다.

$$R_x(avg_{pitch}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(avg_{pitch}) & -\sin(avg_{pitch}) & 0 \\ 0 & \sin(avg_{pitch}) & \cos(avg_{pitch}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_y(avg_{azimuth}) = \begin{bmatrix} \cos(avg_{azimuth}) & 0 & \sin(avg_{azimuth}) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(avg_{azimuth}) & 0 & \cos(avg_{azimuth}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_z(avg_{roll}) = \begin{bmatrix} \cos(avg_{roll}) & -\sin(avg_{roll}) & 0 & 0 \\ \sin(avg_{roll}) & \cos(avg_{roll}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

세 행렬을 곱하여 회전행렬을 통합한다.

$$R_{xyz} = R_x(avg_{pitch}) * R_y(avg_{azimuth}) * R_z(avg_{roll})$$

마지막으로 3차원 이미지를 다시 2차원으로 투영하는 행

렬을 구한다.

$$P_{3d \rightarrow 2d} = \begin{bmatrix} 1 & 0 & image.width/2 & 0 \\ 0 & 1 & image.height/2 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

3차원 투영행렬과 회전행렬, 2차원 투영행렬을 곱하여 변환행렬을 구한다.

$$M_{trans} = P_{3d \rightarrow 2d} * (R_{xyz} * P_{2d \rightarrow 3d})$$

구해진 변환행렬을 사용하여 각 이미지를 와핑(warping)한다. 와핑은 원래 이미지의 점들의 위치를 바꾸어 영상을 왜곡시키는 방법이다. [그림 4]는 회전 전의 이미지와 센서 데이터이다. 각 센서 데이터의 차이를 구하면 다음과 같다.

x축을 기준으로 각 이미지를 약 7.505°, y축을 기준으로 약 5.615°, z축을 기준으로 약 5.43° 회전하면 [그림 5]와 같은 결과가 나오며 두 이미지가 거의 비슷한 선상에 위치하게 된다.



그림 5. 3차원 회전 후 이미지
Fig. 5. Images after 3d rotation

Ⅳ. 이미지 스티칭

[그림 6]은 비슷한 두 이미지를 붙여 파노라마 이미지를 생



그림 4. 3차원 회전 전 이미지와 센서 데이터
Fig. 4. Images and sensed data before 3d rotation

$$\begin{aligned} \Delta pitch &= 0.26191015(\text{radian}) = \text{약 } 15.01^\circ \\ \Delta roll &= 0.1895717788(\text{radian}) = \text{약 } 10.86^\circ \\ \Delta azimuth &= 0.1959486(\text{radian}) = \text{약 } 11.23^\circ \end{aligned}$$

성하는 작업인 이미지 스티칭 과정을 보여주는 흐름도이다.

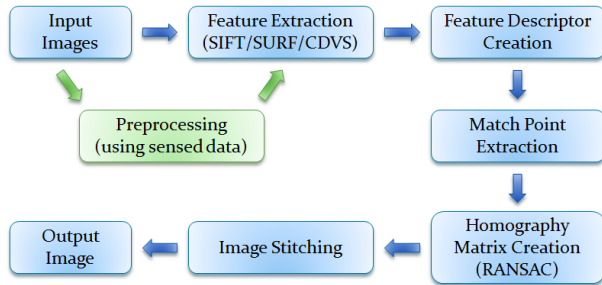


그림 6. 이미지 스티칭 흐름도
Fig. 6. Image stitching flow chart

본 논문에서 제안하는 관성 센서 데이터를 활용한 전처리하는 이미지 입력과 특징점 추출 과정 사이에 진행한다.

1. 이미지 특징점 추출

이미지에서 물체를 인식하거나 추적하고 매칭할 때, 일반적으로 특징점을 추출하여 이미지 매칭을 수행한다.



그림 7. 쉽게 식별이 가능한 점 A와 쉽게 식별이 가능하지 않은 점 B
Fig. 7. Easily identified point A and not easily identified point B

[그림 7]에서 왼쪽 이미지의 A, B와 대응되는 지점을 오른쪽 이미지에서 찾을 때, A는 쉽게 찾을 수 있지만 B는 그렇지 않다. A와 같이 식별이 용이한 지점을 특징점으로 잡아 매칭을 수행하는 것이 유리하다. 본 연구에서는 A와 같은 조건을 가진 점을 추출하는 SIFT, SURF, MPEG-7 CDVS 알고리즘을 사용하여 특징점을 추출한다.

2. 이미지 스티칭

두 이미지를 붙이기 위해서는 특징점 추출 알고리즘을 사용하여 이미지의 특징점을 추출해야 한다. 추출된 특징

점은 [그림 8]과 같이 매칭에 필요 없는 이상점(outlier)들을 포함한다. 이상점은 데이터의 분포에서 현저하게 벗어나 있는 값이다. 이러한 이상점들을 제거하기 위해 M. A. Fischler와 R. C. Bolles가 제안한 RANSAC알고리즘을 사용한다^[15].

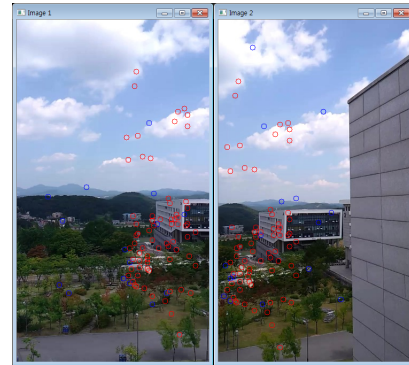


그림 8. 추출된 CDVS 특징점(참인 점: 빨강, 이상점: 파랑)
Fig. 8. Extracted CDVS interest point(inlier: red, outlier: blue)

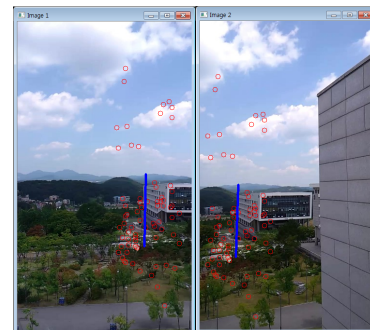


그림 9. 추출된 CDVS 특징점과 예측된 직선
Fig. 9. Extracted CDVS interest point(removed outlier) and line prediction

[그림 9]는 RANSAC 알고리즘을 사용하여, CDVS 특징점의 참인 점을 추출하고, 참인 점으로부터 예측한 직선을 나타낸다. RANSAC 알고리즘을 통해 구한 각 이미지의 참인 특징점들로 호모그래피(Homography) 행렬을 구한 후, 이를 적용해 이미지를 와핑(warping) 시킨다. 호모그래피 행렬을 사용하여 와핑한 이미지에 다른 이미지를 더하여 이미지 스티칭을 하였다. [그림 10]은 이미지 스티칭 결과이다. 빨간색 원 부분을 보면 스티칭이 매끄럽게 되지 않은 것을 볼 수 있다. 이는 호모그래피가 정확하게 추정되지 않아 발생한 결

과이다. 본 논문에서 제안하는 방법을 사용하여 특징점 추출 전에 이미지를 3D 회전하는 전처리 과정을 거친 후, 이미지 스티칭 과정을 진행한 결과는 [그림 11]과 같다.

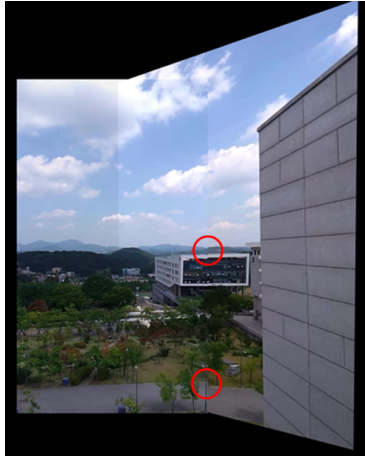


그림 10. 스티칭 결과 이미지
Fig. 10. Stitching result image



그림 11. 전처리 후 스티칭 결과 이미지
Fig. 11. Stitching result image after preprocessing

V. 실험 결과

1. 실험 환경

관성 센서 데이터를 활용한 이미지 스티칭 실험을 위해

[표 1]과 같은 실험 환경을 구성하였다.

표 1. 관성 센서 데이터를 활용한 이미지 스티칭 실험 환경

Table 1. Experimental environment of image stitching using sensed data

Experimental environment of image stitching	
CPU	Intel Core i7-4770 3.4GHz
RAM	8GB
OS	Windows 7 Enterprise K 64-bit
IDE	Visual Studio Professional 2013
Library	OpenCV 2.4.11
	MPEG-7 CDVS evaluation framework

2. 관성 센서 데이터 수집방법

실험 데이터는 다음과 같은 규칙으로 제작되었다. 수평으로 0°, 수직으로 0°, 좌우회전 0°인 기준 이미지를 왼쪽 스마트폰에서 촬영하였다. 또한, 오른쪽 스마트폰은 수평으로 0~12°, 수직으로 0~12°, 좌우회전 0~12°로 4도씩 회전하여 촬영하였다.

3. 이미지 특징점 추출 알고리즘 성능 비교

특징점의 개수는 일정량 이상이 되면 데이터 처리시간이 증가하게 된다. 반면에 특징점 추출속도는 빠를수록 이미지 스티칭에 걸리는 시간이 줄어들게 된다. 본 논문에서는 640×360의 이미지 1만장을 각 알고리즘을 사용하여 추출하였다.

표 2. SIFT, SURF, CDVS 특징점 추출 알고리즘 비교

Table 2. Interest point extraction algorithms comparison of SIFT, SURF, CDVS

	SIFT	SURF	CDVS
Avg. extraction time(sec)	1.726	0.347	1.111
Avg. number of interest points(unit)	1776.11	1553.95	300
Avg. number of inliers(unit)	155.66	251.68	39.2
The ratio of inliers per extracted interest points(%)	8.76	16.20	13.06

[표 2]는 SIFT, SURF, CDVS 특징점 추출 알고리즘의 비교 결과이다. SIFT 알고리즘은 세 알고리즘 중 가장 많은 특징점을 추출하였고, 특징점당 참인 점의 비율이 8.76%로

가장 낮았다. SURF 알고리즘은 단시간에 많은 특징점을 추출하였으며 특징점당 참인 점의 비율이 16.2%로 가장 높았다. CDVS 알고리즘은 프로그램 상에서 제한한 대로 300개의 특징만 추출하였으며, 특징점당 참인 점의 비율은 13.06%로 중간에 위치하였다.

4. 이미지 스티칭 성능 비교

이미지 스티칭의 성능 비교 시 중요한 요소 중 하나는 스티칭에 걸린 시간이다. [표 3]은 SIFT, SURF, CDVS 알고리즘을 사용하여 전체 이미지 스티칭을 수행했을 때 걸린 시간이다. SIFT 알고리즘은 특징점 추출시간과 같이, 스티칭 시간도 4.53초로 가장 오래 걸렸다. SURF 알고리즘은 스티칭에 걸리는 시간이 SIFT의 특징점 추출시간보다 빠른 1.116초로 가장 빨랐다.

표 3. SIFT, SURF, CDVS 이미지 스티칭 시간 비교

Table 3. Image stitching time comparison of SIFT, SURF, CDVS

	SIFT	SURF	CDVS
Avg. stitching time(sec)	4.530	1.116	2.477

이미지 스티칭의 성능은 스티칭에 걸린 시간 외에도 정확도로 평가할 수 있다. 하지만 스티칭 정확도는 마땅한 정확도 평가방법이 존재하지 않는다. 따라서 스티칭 결과 화면을 직접 분석하였다. 알고리즘별 스티칭 정확도를 평가하기 위하여 결과 이미지를 봤을 때, 완벽히 일치하면 “상”, 이미지가 약간 어긋났을 때 생기는 잔상효과가 생기면 “중”, 이미지가 어긋나면 “하”, 호모그래피 행렬이 비정상적으로 생성되거나 스티칭에 실패하면 “실패”로 평가하였다. <부록 1>은 이미지 스티칭 정확도 평가표이다. 실험의 결과를 종합하였을 때, 특징점 추출 속도와 정확도 면에서 SURF 알고리즘을 선택하는 것이 빠르고 정확하게 스티칭을 진행할 수 있었다.

본 논문에서는 3가지 자유도에 강인한 이미지 스티칭을 위하여 관성 센서 데이터를 활용한 전처리를 하였다. [그림 12]는 전처리를 하지 않고 SIFT 알고리즘으로 스티칭을 한 결과이고 [그림 13]은 전처리 후 SIFT 알고리즘으로 스티칭을 한 결과이다.



그림 12. 전처리를 하지 않은 SIFT 스티칭 결과

Fig. 12. Stitching result using SIFT without preprocessing



그림 13. 전처리 후 SIFT 스티칭 결과

Fig. 13. Stitching result using SIFT with preprocessing

전처리 후 스티칭 결과는 <부록 2>와 같다. 전처리 대상은 <부록 1>의 결과에서 스티칭 평가 “중” 이하인 실험데이터로 정하였다. 실험을 통해 관성 센서 데이터를 활용한 전처리를 이용하면 스티칭 정확도를 높일 수 있다는 것을 보았다.

VI. 결론 및 향후 연구

이미지의 특징점 추출 알고리즘 성능평가의 중요한 요소는 추출 및 스티칭에 걸리는 시간, 추출된 특징점당 참인 점의 비율, 매칭의 정확도이다. 주어진 데이터를 기반으로 실험한 결과, 추출 및 스티칭에 걸리는 시간은 SURF 알고리즘이 제일 적게 걸렸고, 추출된 특징점당 참인 점의 비율도 SURF 알고리즘이 제일 높았으며 매칭 정확도는 또한

SURF 알고리즘이 가장 높았다. 실험결과 SURF 알고리즘이 가장 효율이 좋음을 보였다. 이는 실험에 사용한 데이터의 스케일 변화가 적어, 두 상대 영상 간의 스케일 변화를 고려하여 특징점을 찾아내는 SIFT나 CDVS보다 SURF의 속도 및 정확도가 더 높았던 것으로 판단된다.

본 논문은 기존의 이미지 스티칭 방법에서 이미지의 특징점을 추출하기 전 미리 관성 센서 데이터를 활용하여 각 이미지를 3차원 회전하는 방법으로 스티칭 정확도를 향상시키려는 시도를 하였다. <부록 1>은 전처리 과정을 사용하지 않은 스티칭 정확도이며, 스티칭 정확도 향상을 위하여 스티칭 정확도가 “중” 이하인 데이터에 대하여 전처리 과정을 적용한 스티칭을 수행하였다. 육안으로 정확도를 판단하여 비교한 결과, 제안한 방법이 추가되는 시간에 비해 정확도를 향상시킬 수 있어서 효과적인 이미지 스티칭을 수행함을 보였다.

향 후 비디오 스티칭 시 제안된 관성 센서 데이터 영상 스티칭 방법을 이용하여, 영상 특징점 추출을 최소화하여 스티칭 속도를 높이면서도 비디오 스티칭 정확도를 높이는 방법에 대하여 연구를 진행할 예정이다.

참 고 문 헌 (References)

- [1] Panorama VR, from <http://www.uok3d.com/bussiness/panorama.php>
- [2] K. M. Park, W. H. Seok, and K. H. Lee, “Sensor Industry and Major Promising Sensor Market and Technology Trend,” Electronics and Telecommunications Research Institute, 2015, 02.
- [3] S. H. Lee, “State and the Issues of IOT,” Institute for Information & communications Technology Promotion, 2014, 04.
- [4] Park, Siyoung; Kim, Jongho; Yoo, Jisang, “Fast Stitching Algorithm by using Feature Tracking,” Journal of Broadcast Engineering, Vol. 20, no 5 pp. 728-737, 2015.
- [5] Ahn, Yang-Keun, Hong, Ji-Man, “Nonlinear Optimization Method for Multiple Image Registration,” Journal of Broadcast Engineering, Vol. 17, no. 4, pp. 634-639, 2012.
- [6] Kwon, Oh Seol, “Image Matching Based on Robust Feature Extraction for Remote Sensing Haze Images,” Journal of Broadcast Engineering, Vol. 21, no. 2, pp. 272-275, 2016.
- [7] D. G. Lowe, “Distinctive Image Features from Scale-invariant Keypoints,” International Journal of Computer Vision, 60, 2, 91-110, 05, 2004.
- [8] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, “Speeded-Up Robust Features (SURF),” Similarity Matching in Computer Vision and Multimedia, 110, 3, 346-359, 06, 2008.
- [9] H. Bay, T. Tuytelaars, and L. V. Gool, “SURF: Speeded Up Robust Features,” 9th European Conference on Computer Vision, 404-417, 05, 2006.
- [10] P. M. Panchal, S. R. Panchal, and S. K. Shah, “A Comparison of SIFT and SURF,” International Journal of Innovative Research in Computer and Communication Engineering, 1, 2, 323-327, 04, 2013.
- [11] L. Juan and O. Gwun, “A Comparison of SIFT, PCA-SIFT and SURF,” International Journal of Image Processing (IJIP), 3, 4, 143-152, 2009.
- [12] CDVS1, “Call for Proposals for Compact Descriptors for Visual Search,” N12201, Turin, Italy, ISO/IEC JTC1/SC29/WG11, 2011.
- [13] Ling-Yu Duan, Feng Gao, Jie Chen, Jie Lin, and Tiejun Huang, “Compact Descriptors for Mobile Visual Search and MPEG CDVS Standardization,” IEEE International Symposium on Circuits and Systems, 885-888, 05, 2013.
- [14] H. K. Kim, “MPEG-7 CDVS Standardization Media Searching under Mobile Environment,” Telecommunications Technology Association, 144, 35-38, 2012.
- [15] 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, 24, 6, 381-395, 06, 1981.
- [16] E. Dubrofsky, “Homography Estimation,” University of British Columbia, 2009.
- [17] Y. H. Park and O. S. Kwon, “Multiple Homographies Estimation using a Guided Sequential RANSAC,” The Korea Contents Association, 10, 7, 10-22, 2010, 07.
- [18] R. Szeliski, “Image Alignment and Stitching: A Tutorial,” Foundations and Trends® in Computer Graphics and Vision, 2, 1, 1-104, 2006.
- [19] M. Brown and D. G. Lowe, “Automatic Panoramic Image Stitching using Invariant Features,” International Journal of Computer Vision, 74, 1, 59-73, 2007, 12.
- [20] J. Heikkilä and O. Silvén, “A Four-step Camera Calibration Procedure with Implicit Image Correction,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1106-1112, 06, 1997

부 록

<부록 1> 이미지 스티칭 정확도 평가(상: 3, 중: 2, 하: 1, 실패: 0)

Index	Description	SIFT	SURF	CDVS
1	Reference image	-	-	-
2	Horizontally 4° tilted image	3	3	3
3	Horizontally 8° tilted image	3	3	2
4	Horizontally 12° tilted image	3	3	3
5	Vertically 4° tilted image	3	3	3
6	Horizontally 4°, vertically 4° tilted image	3	3	3
7	Horizontally 8°, vertically 4° tilted image	3	2	2
8	Horizontally 12°, vertically 4° tilted image	3	2	2
9	Vertically 8° tilted image	3	3	2
10	Horizontally 4°, vertically 8° tilted image	2	3	1
11	Horizontally 8°, vertically 8° tilted image	2	3	2
12	Horizontally 12°, vertically 8° tilted image	3	3	3
13	Vertically 12° tilted image	3	3	1
14	Horizontally 4°, vertically 12° tilted image	1	3	1
15	Horizontally 8°, vertically 12° tilted image	3	1	0
16	Horizontally 12°, vertically 12° tilted image	0	1	1
17	Rightwards 4° tilted image	3	3	3
18	Horizontally 4°, rightwards 4° tilted image	3	3	1
19	Horizontally 8°, rightwards 4° tilted image	3	3	1
20	Horizontally 12°, rightwards 4° tilted image	1	2	1
21	Vertically 4°, rightwards 4° tilted image	3	3	3
22	Horizontally 4°, vertically 4°, rightwards 4° tilted image	3	3	2
23	Horizontally 8°, vertically 4°, rightwards 4° tilted image	3	2	3
24	Horizontally 12°, vertically 4°, rightwards 4° tilted image	3	2	1
25	Vertically 8°, rightwards 4° tilted image	3	3	3
26	Horizontally 4°, vertically 8°, rightwards 4° tilted image	3	3	3
27	Horizontally 8°, vertically 8°, rightwards 4° tilted image	2	3	1
28	Horizontally 12°, vertically 8°, rightwards 4° tilted image	3	3	1
29	Vertically 12°, rightwards 4° tilted image	2	3	1
30	Horizontally 4°, vertically 12°, rightwards 4° tilted image	2	2	1
31	Horizontally 8°, vertically 12°, rightwards 4° tilted image	2	2	2
32	Horizontally 12°, vertically 12°, rightwards 4° tilted image	0	3	3
33	Rightwards 8° tilted image	0	3	3
34	Horizontally 4°, rightwards 8° tilted image	3	2	3
35	Horizontally 8°, rightwards 8° tilted image	3	3	1
36	Horizontally 12°, rightwards 8° tilted image	2	3	1
37	Vertically 4°, rightwards 8° tilted image	3	3	3
38	Horizontally 4°, vertically 4°, rightwards 8° tilted image	3	3	2
39	Horizontally 8°, vertically 4°, rightwards 8° tilted image	3	3	2
40	Horizontally 12°, vertically 4°, rightwards 8° tilted image	3	3	2
41	Vertically 8°, rightwards 8° tilted image	0	3	2
42	Horizontally 4°, vertically 8°, rightwards 8° tilted image	2	3	1
43	Horizontally 8°, vertically 8°, rightwards 8° tilted image	2	2	3

44	Horizontally 12°, vertically 8°, rightwards 8° tilted image	2	3	3
45	Vertically 12°, rightwards 8° tilted image	2	3	1
46	Horizontally 4°, vertically 12°, rightwards 8° tilted image	1	3	1
47	Horizontally 8°, vertically 12°, rightwards 8° tilted image	2	3	2
48	Horizontally 12°, vertically 12°, rightwards 8° tilted image	1	1	1
49	Rightwards 12° tilted image	3	3	2
50	Horizontally 4°, rightwards 12° tilted image	3	2	3
51	Horizontally 8°, rightwards 12° tilted image	3	3	3
52	Horizontally 12°, rightwards 12° tilted image	3	2	3
53	Vertically 4°, rightwards 12° tilted image	3	3	3
54	Horizontally 4°, vertically 4°, rightwards 12° tilted image	3	3	3
55	Horizontally 8°, vertically 4°, rightwards 12° tilted image	2	3	1
56	Horizontally 12°, vertically 4°, rightwards 12° tilted image	3	3	3
57	Vertically 8°, rightwards 12° tilted image	3	3	2
58	Horizontally 4°, vertically 8°, rightwards 12° tilted image	2	3	2
59	Horizontally 8°, vertically 8°, rightwards 12° tilted image	3	1	3
60	Horizontally 12°, vertically 8°, rightwards 12° tilted image	1	3	1
61	Vertically 12°, rightwards 12° tilted image	2	3	1
62	Horizontally 4°, vertically 12°, rightwards 12° tilted image	2	3	1
63	Horizontally 8°, vertically 12°, rightwards 12° tilted image	0	1	3
64	Horizontally 12°, vertically 12°, rightwards 12° tilted image	2	2	1
Average		2.33	2.65	1.98

<부록 2> 전처리 후 스티칭 결과

Index	Description	SIFT	SURF	CDVS
10	Horizontally 4°, vertically 8° tilted image	2->3	3	1->3
13	Vertically 12° tilted image	3	3	1->1
14	Horizontally 4°, vertically 12° tilted image	1->1	3	1->1
15	Horizontally 8°, vertically 12° tilted image	3	1->2	0->1
16	Horizontally 12°, vertically 12° tilted image	0->3	1->0	1->0
18	Horizontally 4°, rightwards 4° tilted image	3	3	1->2
19	Horizontally 8°, rightwards 4° tilted image	3	3	1->2
20	Horizontally 12°, rightwards 4° tilted image	1->3	2->2	1->3
24	Horizontally 12°, vertically 4°, rightwards 4° tilted image	3	2->3	1->2
27	Horizontally 8°, vertically 8°, rightwards 4° tilted image	2->2	3	1->2
28	Horizontally 12°, vertically 8°, rightwards 4° tilted image	3	3	1->3
29	Vertically 12°, rightwards 4° tilted image	2	3	1->2
30	Horizontally 4°, vertically 12°, rightwards 4° tilted image	2->2	3	1->2
31	Horizontally 8°, vertically 12°, rightwards 4° tilted image	2->1	2->3	2->3
32	Horizontally 12°, vertically 12°, rightwards 4° tilted image	0->2	3	3
33	Rightwards 8° tilted image	0->3	3	3
35	Horizontally 8°, rightwards 8° tilted image	3	3	1->3
36	Horizontally 12°, rightwards 8° tilted image	2->3	3	1->2
41	Vertically 8°, rightwards 8° tilted image	0->3	3	2->2
42	Horizontally 4°, vertically 8°, rightwards 8° tilted image	2->3	3	1->3
45	Vertically 12°, rightwards 8° tilted image	2->2	3	1->2

46	Horizontally 4°, vertically 12°, rightwards 8° tilted image	1->2	3	1->2
48	Horizontally 12°, vertically 12°, rightwards 8° tilted image	1->3	1->3	1->2
55	Horizontally 8°, vertically 4°, rightwards 12° tilted image	2->3	3	1->2
59	Horizontally 8°, vertically 8°, rightwards 12° tilted image	3	1->3	3
60	Horizontally 12°, vertically 8°, rightwards 12° tilted image	1->3	3	1->2
61	Vertically 12°, rightwards 12° tilted image	2->2	3	1->3
62	Horizontally 4°, vertically 12°, rightwards 12° tilted image	2->2	3	1->2
63	Horizontally 8°, vertically 12°, rightwards 12° tilted image	0->2	1->3	3
64	Horizontally 12°, vertically 12°, rightwards 12° tilted image	2->3	2->3	1->2
Average		1.77 -> 2.57	2.53 -> 2.83	1.30 -> 2.20

저 자 소 개



김민우

- 2014년 ~ 2016년 : 명지대학교 컴퓨터공학과 BS, MS
- 2016년 11월 ~ 현재 : MintPot 기술이사
- ORCID : <http://orcid.org/0000-0002-9405-6069>
- 주관심분야 : image processing, pattern recognition, 4D media, sensors and actuators, VR, and Internet of Things



김상균

- 1997년 : 아이오와 대학 (U of Iowa) 전산과학 BS(1991), MS((1995), PhD
- 1997년 3월 ~ 2007년 2월 : 삼성종합기술원 멀티미디어랩
- 2007년 3월 ~ 현재 : 명지대학교 공과대학 컴퓨터공학과 교수
- 2016년 3월 ~ 현재 : 명지대학교 IT융합대학 융합소프트웨어학부 교수
- MPEG 국제표준화 (MPEG-7, MPEG-A, MPEG-V, IoMT & W) 내 다수의 의장/에디터 수임
- ORCID : <http://orcid.org/0000-0002-2359-8709>
- 주관심분야 : digital content (image, video, and music) analysis and management, fast image search and indexing, color adaptation, 4D media, sensors and actuators, VR, Internet of Things, and multimedia standardization