

An Improved RANSAC Algorithm Based on Correspondence Point Information for Calculating Correct Conversion of Image Stitching

Hyunchul Lee[†] · Kangseok Kim^{**}

ABSTRACT

Recently, the use of image stitching technology has been increasing as the number of contents based on virtual reality increases. Image Stitching is a method for matching multiple images to produce a high resolution image and a wide field of view image. The image stitching is used in various fields beyond the limitation of images generated from one camera. Image Stitching detects feature points and corresponding points to match multiple images, and calculates the homography among images using the RANSAC algorithm. Generally, corresponding points are needed for calculating conversion relation. However, the corresponding points include various types of noise that can be caused by false assumptions or errors about the conversion relationship. This noise is an obstacle to accurately predict the conversion relation. Therefore, RANSAC algorithm is used to construct an accurate conversion relationship from the outliers that interfere with the prediction of the model parameters because matching methods can usually occur incorrect correspondence points. In this paper, we propose an algorithm that extracts more accurate inliers and computes accurate transformation relations by using correspondence point relation information used in RANSAC algorithm. The correspondence point relation information uses distance ratio between corresponding points used in image matching. This paper aims to reduce the processing time while maintaining the same performance as RANSAC.

Keywords : RANSAC, Random Sampling, Homography, Image Stitching, Performance

이미지 Stitching의 정확한 변환관계 계산을 위한 대응점 관계정보 기반의 개선된 RANSAC 알고리즘

이 현 철[†] · 김 강 석^{**}

요 약

최근 가상현실 기반의 콘텐츠들이 늘어나면서 이미지 Stitching 기술의 사용이 증가하고 있다. 이미지 Stitching이란 고해상도 이미지 및 넓은 시야(Wide Field of View)의 이미지를 생성하기 위해 다중의 영상을 정합하는 방법이다. 이런 이미지 Stitching은 하나의 카메라로부터 생성되는 영상의 한계를 넘어 다양한 분야에서 활용되고 있다. 이미지 Stitching은 다중의 영상을 정합하기 위해 특징 점 및 대응점을 검출하고 RANSAC 알고리즘을 이용하여 영상간의 변환관계(Homography)를 계산한다. 일반적으로 변환관계 계산을 위해 대응점들이 필요하다. 그러나 대응점들에는 변환관계에 대한 잘못된 가정이나 오류로 인해 발생할 수 있는 다양한 유형의 노이즈(Noise)가 포함되어 있다. 이러한 노이즈는 변환관계를 정확히 예측하는 방해 요인이 된다. 이처럼 일반적으로 사용되는 대응점 매칭(Matching) 방법들은 잘못된 대응점들을 매칭할 수 있는 경우가 발생하기 때문에 모델 파라미터의 예측을 방해하는 대응점(Outlier)로부터 정확한 변환관계를 구축하기 위해 RANSAC 알고리즘을 사용한다. 본 논문에서는 RANSAC 알고리즘에 사용되는 대응점 관계 정보를 이용하여 좀 더 정확한 대응점(Inlier)을 추출하고 정확한 변환관계를 계산하는 알고리즘을 제안한다. 대응점 관계 정보는 이미지 매칭에 사용되는 대응점 간의 거리 비율을 사용하며, 본 논문은 기존 RANSAC 알고리즘과 같은 성능을 유지하면서 처리 시간을 단축시키는데 있다.

키워드 : RANSAC, 랜덤 샘플링, Homography, 이미지 Stitching, 성능

※ 이 논문은 2015년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(No. NRF-2015R1D1A1A01060236).

† 준 회 원 : QURAM 연구소 선임연구원

** 정 회 원 : 아주대학교 사이버보안학과 부교수

Manuscript Received : July 24, 2017

First Revision : September 12, 2017

Accepted : September 25, 2017

* Corresponding Author : Kangseok Kim(kangskim@ajou.ac.kr)

1. 서 론

최근 가상현실 기반의 콘텐츠들이 늘어나면서 360도 카메라로 촬영한 영상 등이 많은 사람들에게 관심을 받고 있다. 이런 360도 카메라를 사용하여 다중의 영상을 얻을 경우 촬영된 이미지를 연결해 주는 기술이 필요하다. 이렇게 촬영

된 각각의 영상을 하나로 이어주는 기술을 이미지 Stitching 이라 부르며, 고해상도 이미지 및 넓은 시야각의 이미지와 같은 파노라마 영상을 생성할 때 사용된다[1]. 파노라마 영상이란 다양한 방향에서 영상을 촬영한 후 모든 방향을 한 시점에서 바라볼 수 있도록 단일 영상들을 합성한 영상이다. 이미지 Stitching 기술은 하나의 카메라로부터 생성되는 영상의 한계점들을 해결 할 수 있으며 컴퓨터 비전, 컴퓨터 그래픽스 등과 같은 여러 분야에서 널리 연구되고 있다[2, 3].

가상현실 기반의 콘텐츠들은 대부분 실시간으로 동작하는 경우가 많기 때문에 실시간 캘리브레이션(calibration) 및 실시간 이미지 Stitching과 같은 실시간으로 처리되는 빠른 기법들이 필요하게 된다. 본 논문은 이미지 Stitching에서 사용되는 RANSAC(RANdom SAMple Consensus)의 실시간 처리 능력을 높이기 위해 RANSAC의 속도를 개선하고자 한다.

일반적으로 두 이미지 사이의 변환관계는 Homography로 표현되며, 그 관계를 계산하기 위해 대응되는 특징 점들을 사용한다. 이런 특징 점들은 부정확한 위치 또는 잘못된 대응 관계를 포함하는 경우가 있는데, 이런 데이터는 두 이미지 사이에 실제로 존재하는 변환관계와 다른 경향을 보인다. 이렇게 노이즈나 오류로 인하여 실제 모델과 다른 특성을 보이는 데이터를 Outlier라고 부르고, 반면에 허용 임계치 내에서 모델과 유사한 특성을 보이는 데이터를 Inlier라고 부른다.

만들어진 대응점 중에는 불일치하는 대응점들이 존재할 수 있고, 이런 대응점의 비율이 높은 상태에서 추정된 변환관계는 부정확할 확률이 매우 높다. 결국 Inlier에 비하여 Outlier의 비율이 높은 경우에도 강력한 파라미터 추정(Robust Parameter Estimation)이 가능한 방법으로 RANSAC 알고리즘이 제안되었다[4].

RANSAC 알고리즘은 대응 관계의 Outlier를 식별하여 모델 추정 대상에서 제외하고 Inlier 그룹을 찾는 지역적인 파라미터 추정을 수행한다. 이 방법은 단순하면서도 높은 유연성을 가지고 있어서 다양한 모델 추정에 효과적으로 사용되고 있다. 그러나 RANSAC 알고리즘은 랜덤 샘플링 과정에서 선형을 이루거나 특정영역에 밀집되는 문제 및 Outlier가 포함되는 문제로 인해 불필요한 계산 과정을 수행하여 효율성이 저하되는 문제점이 존재한다.

따라서 본 논문에서는 RANSAC 알고리즘의 랜덤 샘플링과정에서 발생하는 문제를 해결하여 RANSAC 알고리즘의 효율성을 향상시키는, 개선된 RANSAC 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 투영(Projection)된 두 이미지의 변환관계를 구하는 방법과 두 이미지의 변환관계를 계산하는 과정에서 사용되는 RANSAC 알고리즘에 대해 설명한다. 3장에서는 RANSAC 알고리즘의 랜덤 샘플링 문제를 해결하는 방안을 제시한다. 4장에서는 실험을 통하여 제안한 방식과 기존의 RANSAC 알고리즘의 성능을 비교하고, 5장에서 결론을 맺는다.

2. 관련 연구

2.1 Homography Estimation

3차원 공간에 존재하는 임의의 물체는 시점을 달리하는 카메라에 의해 2D 이미지로 투영되어 다중 영상을 구성한다. 투영된 두 평면 사이에는 일정한 변환관계가 성립하는데 이 변환관계를 Homography라고 한다.

Homography는 두 평면의 기하 관계로부터 이미지 모자이크(image mosaicing), 3차원 복원(3D reconstruction) 및 카메라 캘리브레이션등에 이용되는 중요한 파라미터이다[5].

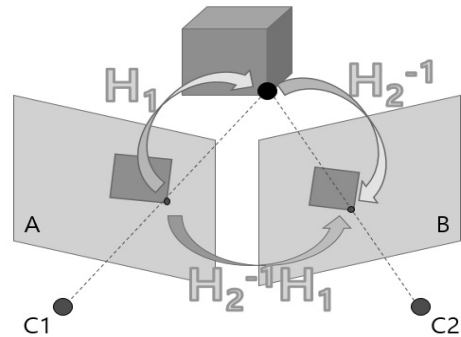


Fig. 1. Homography Transformation Relationship

Fig. 1에서 서로 다른 카메라 위치 C1과 C2에서 이미지 A와 이미지 B로 투영되었다면 이미지 A와 이미지 B의 변환관계를 Homography로 표현할 수 있다.

Homography는 Homogeneous 좌표계에서 정의되며 그 일반식은 Equation (1)과 같다.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \sim \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1)$$

Homography는 자유도가 8이며 Homography를 결정하기 위해서는 최소 4개의 대응점을 필요로 한다. Homography의 자유도가 9가 아니라 8인 이유는 Homogeneous 좌표이므로 Homography의 scale을 결정할 수 없기 때문이다.

4개의 대응되는 점이 주어질 경우 일반식을 만족하는 Homography를 구하는 과정은 Equation (6) 같다. 우선 한 개의 대응점을 구하는 방법은 Equations (2)-(5)이다.

$$\begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} \sim \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad (2)$$

$$x'_i = \frac{u}{w} = \frac{h_{00}x_i + h_{01}y_i + h_{02}}{h_{20}x_i + h_{21}y_i + h_{22}}$$

$$y'_i = \frac{v}{w} = \frac{h_{10}x_i + h_{11}y_i + h_{12}}{h_{20}x_i + h_{21}y_i + h_{22}} \quad (3)$$

$$\begin{aligned} h_{10}x_i + h_{11}y_i + h_{12} - y'_i x_i h_{20} - y'_i y_i h_{21} - y'_i h_{22} &= 0 \\ h_{00}x_i + h_{01}y_i + h_{02} - x'_i x_i h_{20} - x'_i y_i h_{21} - x'_i h_{22} &= 0 \end{aligned} \quad (4)$$

$$\begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x'_i x_i & -x'_i y_i & -x'_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -y'_i x_i & -y'_i y_i & -y'_i \\ & & & & & & \vdots & & \\ x_n & y_n & 1 & 0 & 0 & 0 & -x'_n x_n & -x'_n y_n & -x'_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -y'_n x_n & -y'_n y_n & -y'_n \end{bmatrix} \begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (5)$$

4개의 대응점에 대해서 관계식을 알게 되면 Equation (6)과 같이 Homography Matrix를 구할 수 있다.

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1 x_1 & -x'_1 y_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1 x_1 & -y'_1 y_1 & -y'_1 \\ & & & & & & \vdots & & \\ x_n & y_n & 1 & 0 & 0 & 0 & -x'_n x_n & -x'_n y_n & -x'_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -y'_n x_n & -y'_n y_n & -y'_n \end{bmatrix} \begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (6)$$

결국 Homography Transform Matrix를 계산하기 위해서는 4개의 대응점이 필요한데 일반적으로 대응점이 4개 이상인 경우가 대부분이므로 위 Equation (6)의 정확한 해를 구하기는 어렵다. Equation (6)은 $Ah = 0$ 으로 표현할 수 있으며 h 가 0일 경우 식이 항상 성립하므로 $\|h\| = 1$ 제약조건을 주어 $\|Ah\|$ 를 최소화하는 해를 찾아야 한다. h 는 SVD(Singular Value Decomposition)를 사용하여 가장 작은 singular value에 해당하는 singular vector를 근사 해로서 구하게 된다[6].

Fig. 2와 같이 Homography 추정 방법은 크게 세 가지 단계로 나누어 설명할 수 있다.

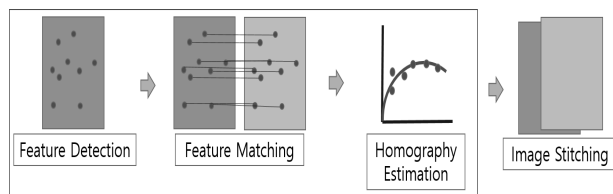


Fig. 2. Homography Estimation Flow

첫 번째 단계는 Feature Detection 단계다. Fig. 3은 이미지 안에 존재하는 특징 점들을 찾는 Feature Detection 과정을 나타낸다. 특징 점은 물체를 추적하거나 인식할 때, 영상과 영상을 매칭할 때 사용한다. 사용되는 알고리즘은 Harris corner[7], FAST[8], SIFT[9] 등이 있으며 본 논문에서는 ORB(Oriented FAST and Rotated BRIEF)를 사용하여 Feature Detection을 진행하였다[10]. 또한 이미지의 외곽 Feature에 중점을 주도록 하였으며 Feature가 선택된 영역

에 중복해서 찾지 않도록 하였다(non-maximal suppression). 영상에서 좋은 특징 점이 되기 위한 조건은 물체의 형태나 크기, 위치가 변해도 쉽게 식별이 가능하며 카메라의 시점, 조명이 변해도 영상에서 해당 지점을 쉽게 찾아낼 수 있어야 한다.



Fig. 3. Feature Detection



Fig. 4. Feature Matching

두 번째 단계는 Feature Matching 단계다. Fig. 4는 두 개의 영상간의 대응하는 점들을 찾는 Feature Matching 과정을 보인다. 이 과정 중에 불일치하는 대응점이 생겨날 수 있으며 불일치하는 점을 걸러내기 위해 RANSAC 알고리즘이 사용된다.

세 번째 단계는 Homography Estimation 단계이다. 모델 추정 단계는 Feature Matching 단계에서 매칭 시킨 대응점을 이용하여 모든 대응점이 만족하는 모델을 생성하는 과정이다. 투영된 영상을 사용하기 때문에 Homography 모델을 사용하여 추정한다. 보통 모델을 추정할 때 방정식의 개수가 미지수의 개수보다 많기 때문에 모든 대응점을 만족하는 모델 파라미터는 구하기 힘들다. 때문에 최대한 적은 에러(Residual)를 포함하는 모델 파라미터를 구하기 위해 최소 자승법과 같은 파라미터 추정 기법을 사용하게 된다.

2.2 RANSAC (RANdom SAmple Consensus)

어떤 현상을 설명하는 이론 모델을 수립하고자 하면 현상으로 부터 관측 데이터를 수집해야 한다. 그러나 관측 데이터에는 모델에 대한 잘못된 가정이나 측정 장비의 오차 등에서 생기는 여러 가지 형태의 노이즈가 있을 수 있다. 이러한 노이즈는 모델을 정확히 예측하는 것을 방해한다. 이

처럼 모델 파라미터의 예측을 방해하는 Outlier 데이터가 들어 있을 경우 관측 데이터로부터 정확한 모델을 구하기 위해 RANSAC 알고리즘을 사용한다. 이미지 관계를 구하는 과정에서 관측데이터는 대응점이 되고 그 대응점으로부터 정확한 Inlier를 찾아내는 것이 필요하다.

RANSAC 알고리즘은 최소 자승법 같은 기술과는 차이가 있다. 최소 자승법은 해를 얻기 위해 가능한 많은 데이터를 초기에 사용한다. 그리고 Outlier에 의한 해를 보정하기 위해 반복적으로 임계 수준을 평가하여 개별 데이터를 제거하고 다시 추정하는 절차를 별도로 추가한다. 그런데, RANSAC은 적합한 초기 데이터 집합을 가능한 작게 사용한다. 그리고 일관성 있는 데이터 집합을 중심으로 Inlier 들을 식별한다. 만일 충분히 일관성을 가진 데이터들을 확보한다면, 최종 추정 단계에서 최소 자승법 같은 기술을 적용한다.

RANSAC은 가설과 검증에 기반한 반복 알고리즘이며, Fig. 5의 가설 설정 단계와 Fig. 6의 가설 검증 단계들을 통해서 실행되는 알고리즘이다[4].

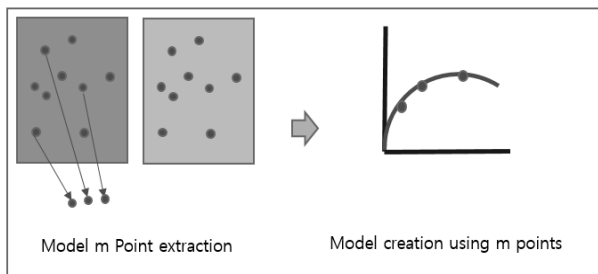


Fig. 5. Hypothesis Setting Step

첫 번째는 가설 설정 단계이다. 입력 데이터 집합으로 부터 모델 추정에 요구되는 최소 샘플 집합을 랜덤하게 선택한다. 랜덤으로 선택된 데이터를 이용하여 모델을 생성한다.

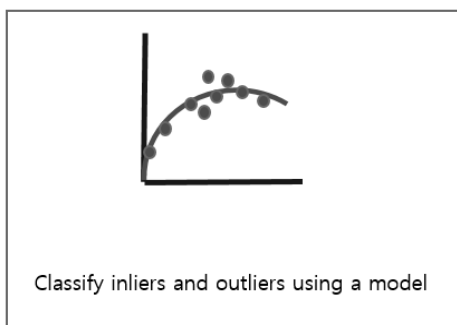


Fig. 6. Hypothesis Verification Step

두 번째는 검증 단계이다. 첫 번째 단계에서 추정된 모델이 전체 데이터 집합의 각 원소들에 대하여 어느 정도 일관성이 있는지를 체크한다. 사전에 설정된 오차 범위 안의 원소들을 Inlier로 선택하여 새로운 집합을 구성한다.

마지막으로 생성된 집합의 원소 개수가 임계 값 보다 크면 이를 이용해서 새로운 모델을 추정하고, 이를 해로서 선

정한다. 그렇지 않으면, 첫 번째 단계부터 다시 반복한다. 만일 사전에 결정된 반복 시도횟수를 초과한다면, 지금까지 발견된 가장 큰 집합에 대한 변환 행렬을 해로서 선정하거나, 실패로 종료한다.

다음 Fig. 7은 매칭된 대응점들의 매칭 관계를 보여주며 Fig. 8은 그 대응점들에 RANSAC 알고리즘을 적용한 결과이다. Fig. 8의 결과를 보면 잘못 대응된 대응점들이 사라지는 것을 확인할 수 있다. 이때 대응점의 개수는 10개이며 그 때 샘플링 할 수 있는 경우의 수는 210 가지가 된다.

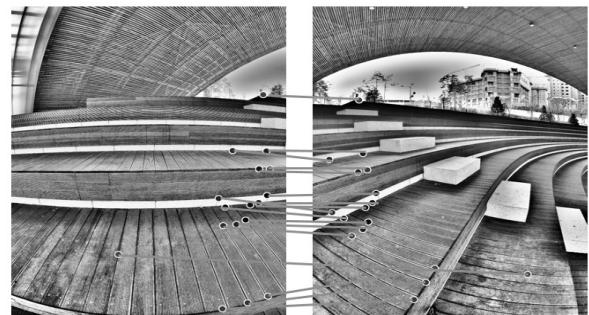


Fig. 7. Matching Relationship of Corresponding Points



Fig. 8. Matching Relation of Corresponding Points after Applying RANSAC Algorithm

결국 모든 경우의 수로 가설을 만들어 테스트를 진행하기 위해선 210번 반복해야 된다. 210번을 모두 반복할 경우 정확한 Inlier를 찾아낼 수 있지만 많은 반복 횟수를 진행하는 건 비효율적이다. 그렇다고 너무 적은 반복 횟수로 진행하면 정확한 Inlier들을 찾아낼 수 없기 때문에 반복 횟수를 설정하는 것도 RANSAC 알고리즘의 중요한 이슈가 된다. RANSAC 알고리즘에서는 대응점의 개수가 증가할수록 경우의 수가 기하급수적으로 늘어나기 때문에 랜덤 샘플링 과정을 적용하여 적은 반복 횟수로 큰 효과를 얻을 수 있는 구조로 만들어져 있다. 하지만 랜덤이라는 과정 때문에 만족할 만한 결과가 나오지 않는 경우가 생길 수 있다. 본 논문에서는 랜덤 샘플링 과정을 신뢰할 수 없어 반복 횟수 (1~500)를 늘려 진행하였다. 현재 랜덤 샘플링과정의 불확실성 때문에 많은 반복이 시도되며 RANSAC 알고리즘의 반복 시도횟수를 최소화하기 위한 연구 또한 활발하게 진행되어 왔다[11-14].

일반적으로 RANSAC에서 사용되는 반복 횟수를 구하는 공식은 Equation (7)과 같다[4].

$$K = \log(1-P) / \log(1-W^n) \quad (7)$$

P : M개 중에서 N개의 대응점을 선택했을 때 Inlier만 선택할 확률.

1-P : 해당 알고리즘이 모두가 Inlier가 아닌 Outlier N개의 대응점을 선택할 확률.

Wⁿ : M개중에서 N번 뽑은 대응점 모두 Inlier일 확률.

1-Wⁿ : N개의 대응점 중에 최소한 1개의 대응점이 Inlier가 아닐 확률.

2.3 기존의 RANSAC 기반 알고리즘

RANSAC 알고리즘은 가설과 검증에 기반 한 반복 알고리즘으로 최소한의 데이터 셋으로 모델을 생성하며, 반복적인 작업을 통해 해를 예측하고 찾는 방법이다. 반복횟수가 많아지면 RANSAC의 정확성은 올라가지만 그만큼 처리 시간도 증가하기 때문에 반복횟수는 RANSAC 알고리즘에서 중요한 변수가 되며 최소한의 반복횟수로 정확성을 높이기 위해 RANSAC 관련된 연구들이 진행되어 왔다.

기존의 연구들은 가설 검증 단계에서 부적합한 가설을 효과적으로 제거하는 방법들[11-13]을 제안하였지만, 본 논문에서는 가설 설정 단계에서 부적합한 대응점이 선택되는 문제를 효과적으로 제거하는 방법을 제안한다. 본 논문과 유사한 기존 연구들도 존재하며, PROSAC 알고리즘[14]은 랜덤성의 단점을 보완한 방법으로 대응점을 매칭하는데 사용된 데이터를 이용한다. 유사성 측정이 임의의 추측보다 일치의 정확성을 보장(예측)한다는 가정하에 PROSAC 알고리즘은 많은 계산량을 절감하였다. 실험에서는 최대 100배 이상의 처리 속도 개선을 보여준다. 제안하는 알고리즘은 대응점 매칭에 사용된 데이터가 아닌 대응점만의 관계정보를 이용하여 처리 속도를 개선할 것이며, PROSAC 알고리즘 기법과 제안하는 알고리즘 기법을 동시에 쓴다면 더 좋은 성능을 낼 수 있을 것이라고 생각한다.

그 외에 다양한 RANSAC 알고리즘들은 비교 분석하고 접근법을 제공하는 동시에 광범위한 Inlier 비율에 대해 정확한 추정을 통해 RANSAC의 정확도와 속도를 개선한 연구 방법[15]이 존재하며 최근 이슈가 되는 Deep Learning 기법을 활용하여 RANSAC의 성능을 향상시키는 연구도 활발하게 진행되고 있다[16].

본 논문에서 제안하는 알고리즘은 기존의 RANSAC 기반의 알고리즘들과 같은 랜덤 샘플링 과정을 개선하였으나, 기존의 알고리즘들과 비교하기는 어려움이 존재한다. 그 이유는 본 논문에서 제안하는 알고리즘이나 기존에 제안된 일부 알고리즘은 샘플링 과정에서 랜덤성을 지니고 있기 때문에 정확한 수치를 계산하고 비교하기가 어렵다. 때문에 본 논문에서는 기존 알고리즘 개선이 주 목표가 아닌 현재 검출된 대응점의 특징을 이용한 반복 횟수를 줄이기 위한 방안을 소개한다. 제안하는 알고리즘은 기존에 제안된 알고리즘과 함께 사용되면 더 많은 효과를 볼 수 있을 것이다.

3. 대응점들 간의 관계정보를 이용한 개선된 RANSAC 샘플링 알고리즘

검출된 대응점들 중에는 대부분 노이즈와 같은 Outlier가 포함되어 있다. 그래서 Outlier가 제거된 대응점 집합으로 모델을 생성해야 한다. 이러한 모델을 찾기 위해서 일반적으로 일치하지 않는 대응점들을 제거해야 된다. 하지만 RANSAC 알고리즘은 모델을 결정하는데 랜덤으로 데이터 집합을 만들기 때문에 Outlier가 포함되어 있을 확률이 높다. RANSAC 알고리즘은 최소한의 데이터 셋으로 모델을 생성하며 반복적인 작업을 통하여 해를 예측하고 찾는 방법이다. 때문에 반복이 많아지면 좀 더 정확한 모델을 생성할 확률이 높아진다. 그러나 반복이 많아지면 처리 시간도 증가하기 때문에 너무 많은 반복 횟수는 RANSAC 성능 저하의 원인이 될 수 있다.

RANSAC 알고리즘은 랜덤으로 집합을 만드는 랜덤 샘플링 과정에서 4개의 대응점을 찾는다. Fig. 9A에 보여진 것처럼 그 4개의 대응점이 모두 Inlier 경우 올바른 모델을 생성하여 Outlier를 제거 할 수 있을 것이다. 그러나 RANSAC 알고리즘의 경우 모든 대응점을 후보군으로 두기 때문에 랜덤 샘플링 과정에서 Outlier가 포함될 수 있다.

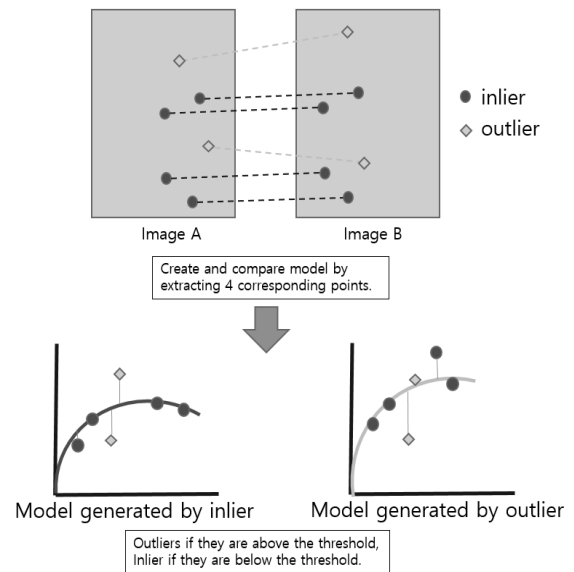


Fig. 9. RANSAC Algorithm Problem

Outlier가 포함된 대응점 집합으로 모델을 생성할 경우 올바른 모델을 만들 수 없으며, Fig. 9B에 보여진 것처럼 Outlier로 선택되어야 할 대응점이 Inlier로 선택될 확률이 높아질 것이다. 이 경우 제대로 된 모델을 만들 수 없으며, 잘못된 결과를 도출하게 된다.

RANSAC 알고리즘 성능에 영향을 주는 중요한 요소는 반복 횟수와 Inlier와 Outlier를 분류하는 임계값이다. 제안하는 알고리즘은 요소 중 반복 횟수를 줄이기 위해 랜덤 샘플링 과정에서 발생하는 선형 문제와 특정 영역에 밀집되는

문제, Outlier 데이터를 포함하는 문제를 검증 전에 파악하여 반복 횟수를 줄이고자 한다. 데이터 셋을 선택할 때 문제가 발생할 가능성을 배제할 경우 불필요한 모델 추정 및 평가를 수행하지 않으므로 많은 시간을 절약할 수 있다.

제안하는 알고리즘에서 사용되는 데이터 셋의 문제 여부 판단 방법은 대응점간의 관계정보를 이용하는데 그 관계 정보는 대응점간의 거리 비율을 비교하여 데이터 셋의 신뢰도를 측정하는 것이다. 그 신뢰도를 이용하여 사전에 문제가 될 데이터 셋을 알아낼 수 있다.

먼저 대응점간의 관계정보에서 거리정보란 각각의 영상 안에 존재하는 대응점들 간의 거리(Pixel)이며, 비율이란 어떤 수량(비교하는 양)의 다른 수량(기준량)에 대한 비의 값이다. 결국 거리 비율이란 Fig. 10에 보여진 것처럼 각각의 영상에 존재하는 거리 정보들에 대해 비를 구한 값이 된다. 정리해서 말하면 두 장의 이미지가 있을 경우 이미지 A의 대응점간의 거리와 이미지 B의 대응점간의 거리 비를 계산한 것이 거리 비율이다.

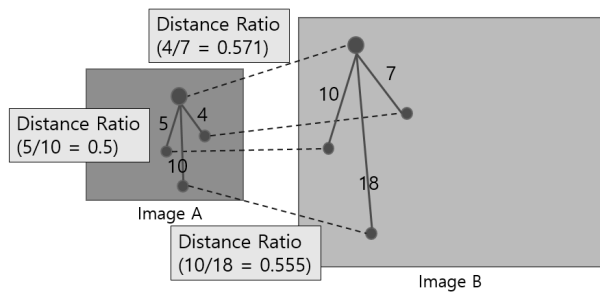


Fig. 10. Distance Ratio

신뢰도란 거리 비율의 평균값을 이용하여 거리 차의 제곱의 합으로 나타낸 점수(Score)이다. 만약 4개의 대응점을 선택할 경우 첫 번째 대응점을 선택할 때 3개의 거리 값이 존재하며 Equation (8)과 같이 각각의 거리 값을 제공한 후 차를 계산한다. 이때 기준이 되는 Image의 거리에는 평균 비율을 곱하여 서로의 크기를 맞추어 준다. 비율을 이용해서 신뢰도로 측정하게 되면 현재 대응점이 Inlier 대응점인지 판단할 수 있게 된다. Equation (8)에 사용되는 n은 대응점의 개수이며, d는 각 대응점간의 거리를 나타낸다. ratio는 Fig.10에서 각 거리 비율의 평균 비율이다.

$$Score = \sum_{k=0}^n d_k^2 - \sum_{k=0}^n (d_k \times ratio)^2 \quad (8)$$

이 과정을 4번 진행하게 되면 선택된 대응점은 모두 Inlier가 될 것이다. 이와 같이 제안된 알고리즘으로 계산할 경우 이미지 크기의 영향을 받지 않으며 각각의 대응점들은 회전과 이동 변환에서도 정확한 신뢰도를 얻을 수 있다. 또한 제안하는 방법은 파노라마 같은 영상을 생성할 때 사용할 수 있는 기법으로 동일한 카메라를 이용하여 촬영한 영상으

로 가정하기 때문에 투영된 영상들은 동일한 왜곡 지수와 동일한 Focal Length를 가지고 있으며 각각의 영상에서 거리계산을 하기 때문에 Focal Length 및 크기(Scale), 회전(Rotation), 이동(Translation) 변환에 대해 고려할 필요가 없다. 하지만 카메라와 물체의 거리가 서로 다를 수 있기 때문에 원근의 정도/방향, 이미지의 왜곡 등이 거리 비율을 계산하는데 문제가 될 수 있다. 특정 원근감으로 인해 생기는 거리 비율 차이는 임계 값 범위 안에서 사용될지 제거될지 정해지며, 잘못된 거리 비율 값은 제거될 것이다. 때문에 RANSAC 샘플링 과정에서 왜곡이 적은 대응점들만 선택하게 되고 일치하는 대응점이지만 왜곡이 심한 대응점은 선택이 되지 않게 된다. 제안하는 샘플링 기법은 특별하게 일치하는 대응점이 샘플링 데이터 셋에 선택되지 못한다는 점인데 이런 문제를 해결하기 위해 좀 더 개선의 여지가 필요하다. 하지만 심하게 왜곡되는 이미지는 드물며 어느 정도 왜곡정보는 임계치 범위 안에 포함되기 때문에 많은 문제를 야기하진 않을 것이다.

제안하는 알고리즘은 예를 들어 이미지 A와 이미지 B에 일치하는 4개의 대응점이 존재한다고 가정했을 때 이미지 A에서 존재하는 대응점들은 이미지 B에서도 비슷한 형태로 존재할 것이다. 하지만 그 위치 좌표는 서로 같지는 않을 것이며, 이미지 A에 존재하는 4개의 대응점의 거리는 이미지 B에 존재하는 4개의 대응점의 거리와 비율에 맞게 비슷할 것이다.

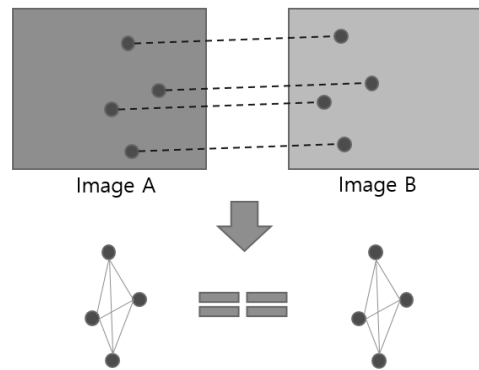


Fig. 11. Correct Distance Ratio

이미지의 첫 번째 대응점과 두 번째 대응점의 거리 비율은 유사할 것이다. 때문에 4개의 대응점이 모두 일치한다면 각각의 이미지 안에 존재하는 대응점의 거리 비율은 서로 유사할 것이다. 반대로 말하면 Fig. 11에 보여진 것처럼 거리 비율이 모두 같다면 4개의 대응점은 모두 일치한다는 것을 알 수 있다.

Fig. 12와 같이 만약 4개의 대응점 중 불일치하는 대응점을 포함하고 있다면 각각의 이미지 안에 존재하는 대응점들 간의 거리 비율은 다를 것이다.

불일치하는 대응점으로 인해 대응점과 연결된 점들의 거리 비율은 서로 달라지며, 달라진 값을 누적하여 신뢰도를 평가할 수 있게 된다.

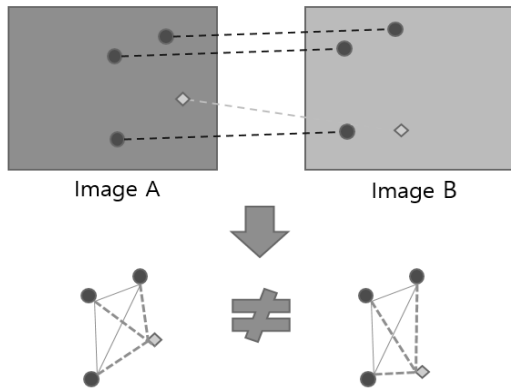


Fig. 12. Incorrect Distance Ratio

결국 모델로 선택된 4개의 대응점이 모두 일치하는 대응점이라면 서로의 위치 거리는 다를 수 있지만 거리 비율은 같을 것이다. 때문에 일치하는 4개의 대응점이 전체 대응점들 사이에 존재한다면 90% Outlier를 걸러낼 수 있을 것이다.

제안된 샘플링 기법이 Outlier를 걸러내지 못하는 10%는 일치하지 않는 대응점을 포함하는 4개의 대응점이 같은 거리 비율로 존재할 경우와 원근 값에 따른 거리 비율 계산의 오차가 신뢰도 범위 안에 포함되지 못할 경우이다.

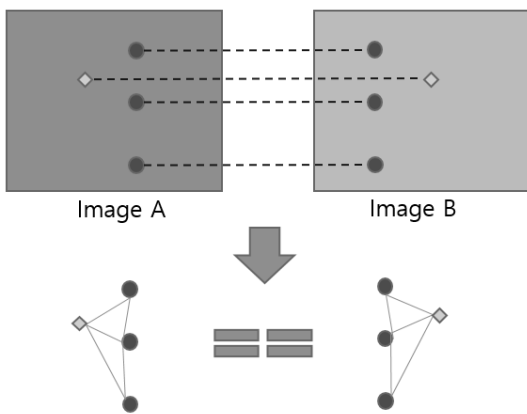


Fig. 13. Problems of Distance Ratio

선택된 대응점이 최악의 경우 같은 거리 비율로 뿔할 수 있으며, 이 경우 제안하는 샘플링 기법은 Inlier를 찾아내지 못한다. 하지만 Fig. 13에 보여진 것처럼 그러한 경우는 극히 드물게 나타나며, 발생한다고 가정해도 RANSAC 알고리즘의 검증 단계에서 걸러진다. 또한 잘못 매칭된 대응점들은 알고리즘상 비슷한 위치에 존재하기 어렵다.

결국 Fig. 14에 보여진 것처럼 개선된 RANSAC 알고리즘을 이용할 경우 대응점간의 거리 비율을 보고 데이터 셋을 선택하기 때문에 두 이미지 간의 특별한 경우를 제외하고 일치하는 대응점들을 선택하여 검증할 것이다. 결국 일치하는 데이터 셋만 모델로 생성하여 Consensus를 확인할 수 있기 때문에 속도 면에서 비용절감이 발생하게 된다.

대응점은 투영된 서로 다른 이미지 안에서 특징 점으로

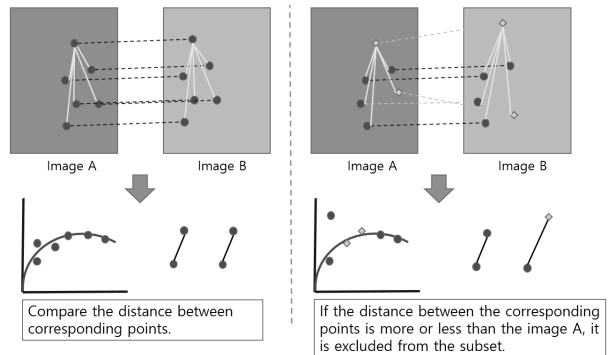


Fig. 14. Improved RANSAC Algorithm

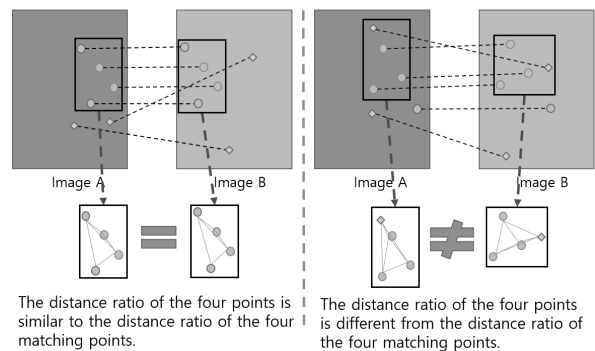


Fig. 15. How to Determine the Ratio of Distance Between Corresponding Points

존재한다. Fig. 15A에 보여진 것처럼 정확한 대응점이라면 각각의 투영된 이미지 안에서 같은 거리 비율로 존재할 것이다. 부정확한 대응점이라면 Fig. 15B와 같이 대응하는 점들 간의 거리 비율이 다르게 존재할 것이다. 거리 비율과 평균 비율의 차이가 임계치 이상이 되면 선택된 대응점들 사이에는 Outlier가 포함되어 있다고 판단할 수 있으며, 임계치 이하가 되면 선택된 대응점들은 모두 Inlier로 구성되어 있을 것이다.

임계값을 정할 때 고려할 사항은 두 가지이다. 첫 번째는 입력 영상의 해상도이다. 해상도가 클 경우 거리 정보의 크기 또한 커지기 때문에 작은 해상도의 이미지 보다 큰 이미지의 해상도에서 더 큰 임계값을 주어야 한다. 두 번째는 대응점 간의 오차 값이다. 대응점을 계산할 때 잘 알려진 알고리즘을 사용한다 해도 정확히 일치하는 대응점을 찾기란 쉽지 않다. 그 이유는 이미지의 노이즈나 밝기 변화, Pixel 값의 오차로 인해 알맞게 일치시킨 대응점이라도 약간의 오차가 생길 수 있다. 때문에 이런 오차를 고려하여 임계값을 좀 더 여유롭게 주어야 된다.

결론적으로 개선된 RANSAC의 샘플링 방법은 대응점간의 관계가 유사하지 않은 점들을 데이터 셋으로 선택하지 않는다. 대응점들 간의 거리 비율의 차이가 샘플 데이터들의 신뢰도가 되며, 그 신뢰도가 높은 데이터 셋들만 검증 과정을 거치면 된다. 결국 정확한 데이터 셋만 모델로 생성하여 검증하기 때문에 속도 면에서 비용절감이 발생한다.

4. 실험 및 결과

본 연구는 RANSAC 알고리즘의 샘플링 과정을 개선하였으며 Algorithm 1이 개선된 RANSAC 랜덤 샘플링 방식을 설명한다. 개선된 버전과 기존 RANSAC 버전을 구분하기 위하여 개선된 버전의 이름을 CONSAC(CONnect random SAample Consensus)이라 부르도록 하겠다.

제안한 CONSAC의 성능을 평가하기 위해서 앞서 기술한 Homography를 구하는 과정을 수행하였으며, 그 과정에서 RANSAC과 CONSAC의 성능을 비교하였다. 성능 분석을 수행한 실험 환경은 다음 Table 1과 같다.

Algorithm 1 proposed RANSAC Random Sample Method

- Input:** 2 images and n corresponding points
- I 4 corresponding points are selected randomly.
 - II Obtain the distance difference among the four corresponding points in a single image.
 - III Calculate the ratio of each distance and then calculate the average distance ratio.
 - IV The reliability of the corresponding points is calculated using the average distance ratio.
 - V Find the exact correspondence based on reliability.
 - VI Repeat steps I through IV until four correct corresponding points are found.
 - VII The four corresponding points are used to verify n corresponding points.
 - VIII Repeat steps I through VI.
 - IX Select the model with the largest number of inliers.
- Output:** inlier corresponding point

Table 1. Experimental Environment

Elements	Values
CPU Specifications	Intel(R) Core(TM) i7-3770K CPU 3.50GHz
Memory Size	12 GB
OS Specifications	Windows 10 Pro x64
Tool	Visual Studio 2013

비교 기준은 크게 세 가지로 나누었으며 수행 반복 횟수와 알고리즘의 수행 시간, 알고리즘을 통해 얻은 모델의 오차율을 사용하였다. 여기서 말하는 오차율은 대응점간의 차이를 나타내며 Homography 모델을 이용하여 대응점의 오차율을 구하였다.

비교하기 위한 이미지 셋은 실내와 실외 셋으로 구성하였으며, 이미지의 크기는 1440x1080x24b이다. 이미지들은 밝은 날 촬영되었으며, 같은 카메라를 사용하였다. 촬영 시 카메라의 중심은 최대한 변화가 없게 회전하면서 촬영하였다. 테스트는 총 20개의 셋으로 진행하였으며 실내 10개 셋과 실외 10개 셋으로, 20개 셋의 평균값을 사용하였다.

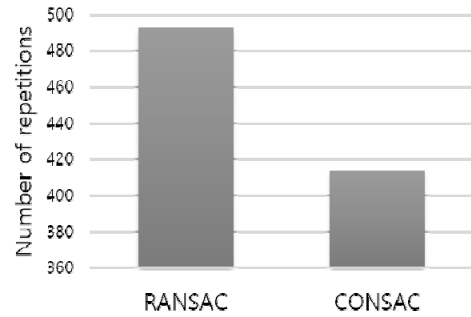


Fig. 16. Number of Repetitions (RANSAC vs. CONSAC)

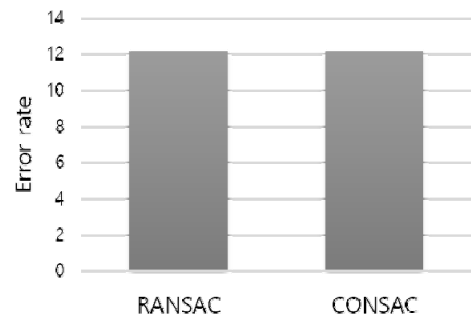


Fig. 17. Error Rate (RANSAC vs. CONSAC)

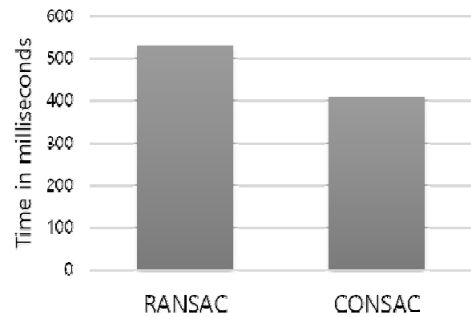


Fig. 18. Processing Time (RANSAC vs. CONSAC)

실험 결과 Fig. 16은 기존 RANSAC을 사용한 반복 횟수보다 CONSAC을 사용한 반복 횟수가 평균적으로 약 20% 정도 적은 횟수로 수행된 결과를 나타내며, Fig. 17에 보여진 것처럼 오차율은 동일하게 나타나는 것을 확인 할 수 있었다. 또한 Fig. 18의 그래프에 보여진 것처럼 반복 횟수가 평균적으로 약 20% 정도 줄면서 약 25%의 처리 시간이 단축되는 것을 확인할 수 있었다.

여러 이미지 데이터의 줄어드는 반복 횟수의 비율을 확인한 결과 Outlier의 비율이 높은 경우 줄어드는 반복 횟수의 비율이 커지며, Inlier의 비율이 높은 경우 RANSAC과 CONSAC의 성능은 비슷하게 유지되는 것을 확인 할 수 있었다. 그러므로 최악(Worst Case)의 경우는 Fig. 19에 보여진 것처럼 RANSAC과 CONSAC의 반복 횟수가 동일하게 나타내며, 최상(Best Case)의 경우는 한 번의 수행으로 RANSAC을 11번 수행한 결과와 동일한 성능을 나타낸다.

본 논문에서 제안한 CONSAC 알고리즘은 RANSAC의 샘플링 과정을 개선하여 Fig. 20에 보여진 것처럼 동일한 오차율을 제공하면서, Fig. 21에 나타난 것처럼 빠른 수행시간을 제공한다는 것을 확인할 수 있었다.

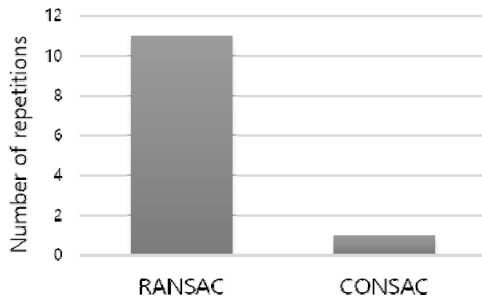


Fig. 19. Number of Repetitions (Best Case) (RANSAC vs. CONSAC)

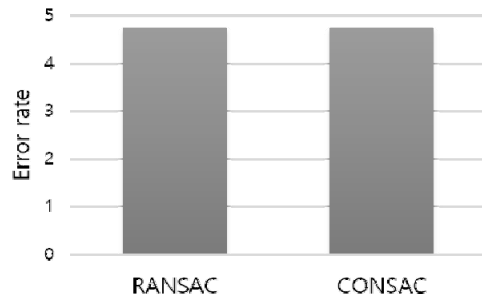


Fig. 20. Error Rate (Best Case) (RANSAC vs. CONSAC)

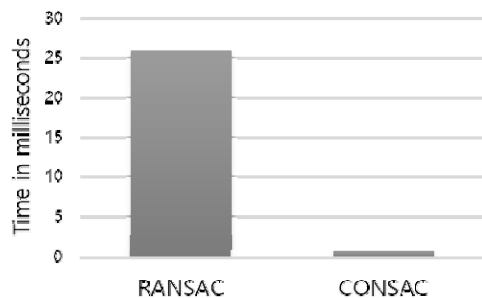


Fig. 21. Processing Time (Best Case) (RANSAC vs. CONSAC)

5. 결론 및 향후 연구계획

5.1 결론

본 논문에서는 두 이미지 사이의 모델(Homography)을 추정하기 위해 RANSAC 알고리즘의 랜덤 샘플링 과정을 개선하였다. 기존의 랜덤 샘플링은 데이터가 선형을 이루거나 특정 영역에 밀집되는 문제와 샘플링 과정을 통해 선택된 데이터에 Outlier 데이터들을 포함하고 있을 경우 문제가 되어 왔다. 그런 잘못된 샘플링 과정을 통해서 불필요하게

모델을 생성하고 검증하기 위한 처리 시간을 낭비하였다.

본 연구에서는 개선된 샘플링 기법을 통해 RANSAC 알고리즘의 처리 시간문제를 해결 하였다. 개선된 샘플링 기법은 대응점들의 관계정보를 이용하여 문제가 되는 샘플링 과정들을 제거함으로써 RANSAC의 처리 시간을 단축시킬 수 있었다. 실험 결과 제안한 방법으로 속도 및 반복 횟수를 측정된 결과 평균적으로 약 20%의 향상을 얻을 수 있었으며, 데이터의 양이 많고 Outlier의 비율이 높은 경우 더 큰 기대 효과를 볼 수 있었다.

5.2 향후 연구 계획

향후 연구 과제로 최근 이슈가 되고 있는 기계학습 기술을 적용하여 RANSAC의 성능을 향상 시킬 방법을 찾고자 한다. 기계학습 기법은 이미지의 특징 점을 추출하거나 특징 점들을 매칭 할 때 또는 모델을 최적화하기 위하여 효과적으로 사용될 수 있을 것이다. RANSAC의 성능을 향상시키기 위해 모델을 생성할 때 최적화 알고리즘을 사용하게 되는데 현재 최적화 알고리즘들은 모두 Local Minimum을 찾고 있다. 따라서 Global Minimum을 효과적으로 찾는 방법 또한 연구되어야 할 과제라고 생각된다.

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] Daehyun Kim and Jongsoo Choi, "View Interpolation Algorithm for Continuously Changing Viewpoints in the Multi-panorama Based Navigation," *IEIE Journal (SP)*, Vol.40, No.6, pp.141-148, 2003.
- [3] Sehwan Kim, Kiyong Kim, and Woontack Woo, "Multiple Camera Calibration for Panoramic 3D Virtual Environment," *IEIE Journal (CI)*, Vol.41, No.2, pp.137-148, 2004.
- [4] 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, 1981.
- [5] E. Dubrofsky, "Homography Estimation," Master Thesis, University of British Columbia, Canada, 2009.
- [6] Richard Hartley and Andrew Zisserman, "Multiple View Geometry in Computer Vision, 2nd Edition," Cambridge University Press, 2003.
- [7] Chris Harris and Mike Stephens, "A combined corner and edge detector," *Proceedings of Fourth Alvey Vision Conference*, University of Manchester, England, 1988.
- [8] Edward Rosten and Tom Drummond, "Machine learning for high-speed corner detection," *European Conference on Computer Vision (ECCV 2006)*, pp.430-443, Graz, Austria, May 7-13, 2006.

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

[10] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski, "ORB: an efficient alternative to SIFT or SURF," *IEEE International Conference on Computer Vision (ICCV 2011)*, pp.2564-2571, Barcelona, Spain, Nov. 6-13, 2011.

[11] J. Matas and O. Chum, "Randomized RANSAC with Td,d test," *Image and Vision Computing*, Vol.22, No.10, pp.837-842, 2004.

[12] D. Capel, "An effective bail-out test for RANSAC consensus scoring," *Proceedings of the 16th British Machine Vision Conference*, pp.629-638, Oxford Brookes University, England, Sept. 5-8, 2005.

[13] J. Matas and O. Chum, "Randomized RANSAC with sequential probability ratio test," *Proceedings of the 10th International Conference on Computer Vision*, pp.1727-1732, Beijing, China, Oct. 15-21, 2005.

[14] O. Chum and J. Matas, "Matching with PROSAC: progressive sample consensus," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Vol.1, pp.220-226, Jun. 20-25, San Diego, CA, USA, 2005.

[15] Rahul Raguram, Jan-Michael Frahm, and Marc Pollefeys, "A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus," *ECCV 2008, Part II, LNCS 5303*, pp.500-513, 2008.

[16] Dustin Morley and Hassan Foroosh, "Improving RANSAC-Based Segmentation Through CNN Encapsulation," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.6338-6347, Jul. 21-26, Hawaii Convention Center, HI, USA, 2017.



이 현 철

<http://orcid.org/0000-0003-2089-2010>

e-mail : deletenim@naver.com

2009년 나사렛대학교 전산정보학(학사)

2012년 아주대학교 지식정보공학과(석사)

2012년~현 재 QURAM 연구소

선임연구원

관심분야 : Image Processing, Computer Vision



김 강 석

<http://orcid.org/0000-0001-8950-7577>

e-mail : kangskim@ajou.ac.kr

2007년 인디애나대학교 컴퓨터공학(박사)

2010년~2016년 아주대학교

지식정보공학과 연구교수

2016년~현 재 아주대학교

사이버보안학과 부교수

관심분야 : 클라우드 컴퓨팅, 유비쿼터스 컴퓨팅, 모바일 보안, IoT 협업시스템