

논문 2014-51-4-16

SURF와 RANSAC 알고리즘을 이용한 대응점 필터링 적용 파노라마 이미지 처리

(Matching Points Filtering Applied Panorama Image Processing Using
SURF and RANSAC Algorithm)

김 정 호*, 김 대 원**

(Jeongho Kim and Daewon Kim[©])

요 약

다중의 영상을 이용하여 하나의 파노라마 영상을 제작하는 기법은 컴퓨터 비전, 컴퓨터 그래픽스 등과 같은 여러 분야에서 널리 연구되고 있다. 파노라마 영상은 하나의 카메라에서 얻을 수 있는 영상의 한계, 즉 예를 들어 화각, 화질, 정보량 등의 한계를 극복할 수 있는 좋은 방법으로서 가상현실, 로봇비전 등과 같이 광각의 영상이 요구되는 다양한 분야에서 응용될 수 있다. 파노라마 영상은 단일 영상과 비교하여 보다 큰 몰입감을 제공한다는 점에서 큰 의미를 갖는다. 현재 다양한 파노라마 영상 제작 기법들이 존재하지만, 대부분의 기법들이 공통적으로 파노라마 영상을 구성할 때 각 영상에 존재하는 특징점 및 대응점을 검출하는 방식을 사용하고 있다. 또한, 대응점을 이용한 RANSAC(RANdom SAMple Consensus) 알고리즘을 사용, Homography Matrix를 구하여 영상을 변환하는 방법을 사용한다. 본 논문에서 사용한 SURF(Speeded Up Robust Features) 알고리즘은 영상의 특징점을 검출할 때 영상의 흑백정보와 지역 공간 정보를 활용하는데, 영상의 크기 변화와 시점 검출에 강하며 SIFT(Scale Invariant Features Transform) 알고리즘에 비해 속도가 빠르다는 장점이 있어서 널리 사용되고 있다. SURF 알고리즘은 대응점 검출 시 잘못된 대응점을 검출하는 경우가 생긴다는 단점이 존재하는데 이는 RANSAC 알고리즘의 수행속도를 늦추며, 그로인해 CPU 사용 점유율을 높이기도 한다. 대응점 검출 오류는 파노라마 영상의 정확성 및 선명성을 떨어뜨리는 핵심 요인이 된다. 본 논문에서는 이러한 대응점 검출의 오류를 최소화하기 위하여 대응점 좌표 주변 3×3 영역의 RGB값을 사용하여 잘못된 대응점들을 제거하는 중간 필터링 과정을 수행하고, 문제해결을 시도하는 동시에 파노라마 이미지구성 처리 속도 및 CPU 사용 점유율 등의 성능 향상 결과와 추출된 대응점 감소율, 정확도 등과 관련한 분석 및 평가 결과를 제시하였다.

Abstract

Techniques for making a single panoramic image using multiple pictures are widely studied in many areas such as computer vision, computer graphics, etc. The panorama image can be applied to various fields like virtual reality, robot vision areas which require wide-angled shots as an useful way to overcome the limitations such as picture-angle, resolutions, and internal informations of an image taken from a single camera. It is so much meaningful in a point that a panoramic image usually provides better immersion feeling than a plain image. Although there are many ways to build a panoramic image, most of them are using the way of extracting feature points and matching points of each images for making a single panoramic image. In addition, those methods use the RANSAC(RANdom SAMple Consensus) algorithm with matching points and the Homography matrix to transform the image. The SURF(Speeded Up Robust Features) algorithm which is used in this paper to extract featuring points uses an image's black and white informations and local spatial informations. The SURF is widely being used since it is very much robust at detecting image's size, view-point changes, and additionally, faster than the SIFT(Scale Invariant Features Transform) algorithm. The SURF has a shortcoming of making an error which results in decreasing the RANSAC algorithm's performance speed when extracting image's feature points. As a result, this may increase the CPU usage occupation rate. The error of detecting matching points may role as a critical reason for disqualifying panoramic image's accuracy and lucidity. In this paper, in order to minimize errors of extracting matching points, we used 3×3 region's RGB pixel values around the matching points' coordinates to perform intermediate filtering process for removing wrong matching points. We have also presented analysis and evaluation results relating to enhanced working speed for producing a panorama image, CPU usage rate, extracted matching points' decreasing rate and accuracy.

Keywords: 파노라마, SURF, RANSAC, 대응점, 필터링

* 학생회원, 단국대학교 대학원 컴퓨터학과

(Department of Computer Science, Graduate School of Dankook University)

** 정회원, 단국대학교 멀티미디어공학과

(Department of Multimedia Engineering, College of Engineering, Dankook University)

© Corresponding Author (E-mail: drdwkim@dku.edu)

접수일자: 2014년1월24일, 수정일자: 2014년2월4일, 수정완료: 2014년3월31일

I. 서 론

파노라마 영상이란 사용자 주변의 다양한 방향에 대하여 영상을 촬영한 후 주위의 모든 방향을 한 시점에서 바라볼 수 있도록 하기 위해 단일 영상들을 결합하여 합성한 영상을 의미한다. 이러한 다중의 영상을 이용하여 파노라마 영상을 제작하는 기법은 컴퓨터 비전, 컴퓨터 그래픽스 등 여러 분야에서 널리 연구되고 있다.^[1~2] 파노라마 영상기법은 하나의 카메라에서 얻을 수 있는 영상의 한계, 즉 예를 들어 화각, 화질, 정보량 등의 한계를 극복할 수 있는 좋은 방법으로서 가상현실, 로봇비전 등과 같은 광각의 영상이 요구되는 다양한 분야에서 응용이 가능하며, 이를 위해 영상의 특징점을 검출하여 대응점 매칭 부분을 찾고^[3~4], 종합적으로 영상을 병합하는 분야에 대한 연구가 진행되고 있다.^[5] 파노라마 영상은 단일 영상과 비교하여 보다 큰 몰입감을 제공한다는 점에서 큰 의미를 갖는데^[6], 이를 제작하는 데에는 다양한 프로그램 환경과 알고리즘들이 사용된다.^[7] 통상적으로 이러한 프로그램들에서 사용되는 공통적인 파노라마 영상 생성 과정을 살펴보면 먼저, 입력된 둘 이상의 영상을 사용하여 각 영상들의 특징점을 구하고, 그 특징점들을 사용하여 각 영상이 서로 대응되는 대응점을 찾는다. 그 후, 대응점들을 이용하여 영상들을 서로 붙이는 과정을 거치게 되는 것이 일반적인 과정이라고 볼 수 있다.^[3~7] 그러나 이와 같은 파노라마 영상을 생성하는 공통적인 과정에서 여러 가지 제한사항들이 나타나게 되는데, 첫 번째 제한사항은 영상을 입력할 때의 순서이다. 예를 들어, 기존의 방법 중엔 두 장의 영상을 사용하여 파노라마 영상을 만들기 위해 두 영상 중 상대적으로 오른쪽 혹은 아래에 위치하게 될 영상을 먼저 입력해야 하는 불편한 경우가 있다. 두 번째 제한사항으로는 영상의 특징점을 사용하여 대응점을 검출할 시 적절하지 않은 대응점들 또한 검출되는 오류가 생길 수 있는데, 이런 이유로 RANSAC (Random Sample Consensus) 알고리즘을 사용하여 호모그래피 행렬을 구하는데 까지의 수행시간이 오래 걸릴 뿐만 아니라 영상을 하나로 결합시킬 때에도 정확성이 떨어지게 된다. 본 논문에서는 대응점 필터링을 통해 SURF(Speeded Up Robust Features)와 RANSAC 알고리즘을 이용하여 파노라마 영상을 생성할 시 나타나는 위에서 언급한 여러 가지 제한 사항을 해결할 수

있는 방안 및 성능 향상 분석 결과를 제시한다. II장에서 파노라마 영상과 SURF 및 RANSAC 알고리즘을 소개하고, III장에서는 본 연구에서 제안하는 문제 해결 방안을 설명하며, IV장에서는 본 논문에서 제안한 개량형 알고리즘의 성능을 테스트하기 위해 샘플 영상을 이용하여 실시한 실험에 대해 논한다. 끝으로 V장에서는 실험 결과를 총 정리하여 제한조건 해결에 관한 내용과 대응점 감소율, 수행속도, CPU 사용 점유율, 정확도 등에서 향상된 결과를 제시하고 이를 분석, 평가한다. 그리고 결론에서는 이를 이용한 향후 연구 방향에 대해 논한다.

II. 파노라마 이미지 처리 기법

1. 파노라마 영상

파노라마 영상은 하나의 카메라에서 얻을 수 있는 영상의 한계를 극복할 수 있는 좋은 방법으로서 광각의 영상이 요구되는 다양한 분야에서 응용되고 있다. 파노라마 영상을 제작하는 데에는 다양한 프로그램 환경과 알고리즘이 사용된다. 그림 1은 다양한 알고리즘을 사용하는 일반적인 파노라마 영상 생성 과정을 나타낸 것이다.^[8] 파노라마 영상을 만들기 위해서는 첫 번째로 파노라마 영상을 구성할 둘 이상의 영상을 획득한다. 그 후 각 영상의 특징점들을 검출한 뒤 이들을 이용하여 대응점을 뽑아내게 되는데, 이 과정에서 해리스 코너 검출기, 헤이시안 검출기, 고속 헤이시안 검출기 등을 사용한다. 그러나 앞서 말한 알고리즘들은 크기와 시점의 변화에 강인하지 못하다는 단점이 있다.^[9~12] 이러한 문제에 대해 강인한 알고리즘들 중 대표적으로

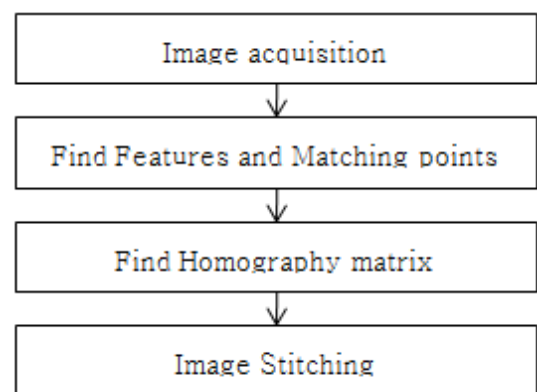


그림 1. 파노라마 영상 생성 순서도

Fig. 1. Flowchart of generation of a panoramic image.

는 SIFT(Scale Invariant Features Transform)^{[13][14]}와 SURF^[15] 기법 등이 있으며 본 연구에서는 SURF 알고리즘을 사용하여 영상의 특징점 및 대응점을 검출하였다. 그 후, 영상 변환을 위하여 검출한 대응점을 사용하고 RANSAC 알고리즘을 수행하여 Homography Matrix를^[16] 구한 후, 이 행렬을 이용하여 두 영상을 병합하는 과정을 거친다^[17~18]. 이러한 기존의 일반적인 파노라마 영상 생성 과정에는 몇 가지 제약 사항들이 존재한다. 이는 파노라마 영상을 획득하는 과정에서 해당 응용 프로그램에 맞게 영상의 입력순서를 지켜줘야 한다는 것이다. 또한 특징점을 사용하여 대응점을 검출할 때 일부 부정확한 대응점을 계산 해내는 경우가 발생한다.

2. SURF(Speeded Up Robust Features) 알고리즘

SURF 알고리즘은 앞서 언급한바와 같이 영상의 특징점^[19] 검출을 위해 사용되는 대표적인 알고리즘이며, 크게 특징점을 검출하는 것과 표현하는 것으로 나뉜다.

$$H_{\approx} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (1)$$

이미지의 특징점 검출 과정에서는 식 (1)에 나타난 근사 헤시안 행렬 (Approximated Hessian Matrix)의 행렬식(Determinant)을 이용하여 이미지의 가장자리 여부를 판단한다.

식 (1)에서, 근사 헤시안 행렬을 구성하는 D 는 가우시안 커널(Gaussian Kernel)의 2차 편도 함수(Second Order Partial Derivative)의 근사식을 의미하며, 각각의 근사식은 그림 2와 같이 필터의 형태로 영상에 회선(Convolution)된다. 그림 3에 나타난 바와 같이 필터들은 스케일이 일정하게 증가되기 때문에 다양한 스케일의 코너가 검출될 수 있다.

이어서 검출된 특징점 위치 주변의 화소 값을 이용하

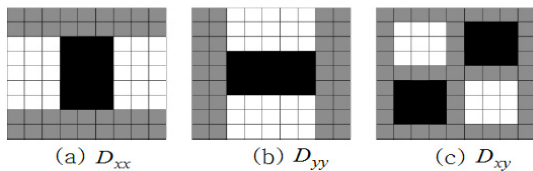


그림 2. 근사 헤시안 행렬의 필터
Fig. 2. Filters of approximated Hessian matrix.

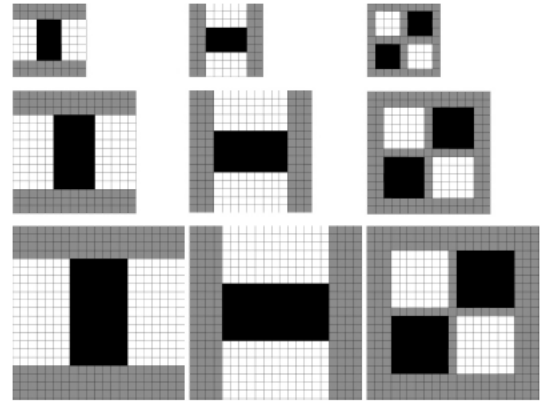


그림 3. 필터들의 스케일 변화에 따른 헤시안 행렬 필터
(D_{xx} , D_{yy} , D_{xy})
Fig. 3. Hessian matrix filters with different scales.
(D_{xx} , D_{yy} , D_{xy})

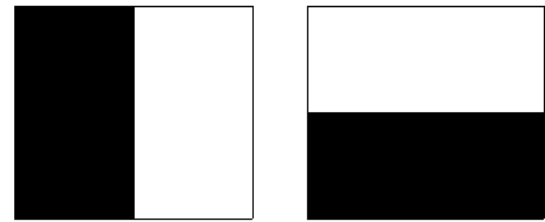


그림 4. 하르 웨이블릿 필터
Fig. 4. Haar wavelet filters.

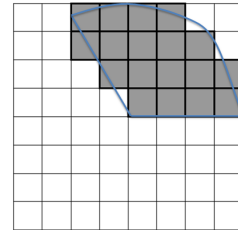


그림 5. 기울기와 슬라이딩 윈도우를 이용한 특징점의 방향 계산
Fig. 5. Calculation of the orientation of featuring points using the gradient and sliding window.

여 특징점의 방향을 계산하게 되는데, 이 때 그림 4에 나타난 하르 웨이블릿 필터(Haar Wavelet Filter)로 회선한 결과인 기울기의 크기와 방향을 이용한다.

계산된 기울기들은 그림 5에 보이는 유색 사각형에 해당되며 특징점의 위치를 중심으로 하는 부채꼴 모양의 슬라이딩 윈도우(Sliding Window)가 360° 회전하면서 윈도우 내부의 크기들을 모두 더한 후, 다수의 벡터를 생성한다. 이 벡터들 중에 가장 크기가 큰 벡터의 방향이 특징점의 방향이 된다. 이렇게 방향이 계산된 특

징점은 최종적으로 벡터 형태의 서술자로 구성되고 특징점 자체를 표현하게 된다^[20]. 본 연구에서는 SURF 알고리즘을 기반으로 하여 다중의 영상에서 특징점을 검출한 뒤, 이들을 이용하여 두 영상 사이의 적절한 대응점을 검출하였다.

3. RANSAC(RANDOM SAmple Consensus) 알고리즘

Fischler^[21]에 의해 제안된 RANSAC 알고리즘은 측정 노이즈가 심한 원본 데이터로부터 적절한 모델 파라미터를 예측하는 방법이며, 이 데이터를 통해 Homography Matrix를 생성한다^[22~23]. 이러한 이론에 의거하여 RANSAC 알고리즘은 전체의 대응점 후보들 중에서 Homography Matrix를 결정하는데 반드시 필요한 최소한의 대응점을 랜덤하게 샘플링하면서 반복적인 과정을 통해 최적의 대응점을 구한다. RANSAC 알고리즘의 수행 순서는 다음과 같다.

- ① : 전체 대응점 후보들로부터 N개의 샘플 대응점 획득
- ② : 획득한 샘플 대응점을 참 값으로 가정하고 Homography Matrix를 예측
- ③ : 예측된 Homography Matrix가 옳은지 판단
- ④ : 참 값이 아닐 경우 ①~③번 과정 반복

위의 과정 ①에서 N은 파노라마 영상을 구성하기 위해 Homography Matrix를 만드는데 필요한 4개의 대응점 개수다. 즉, 파노라마 이미지를 구성하는 데에는 최소한 4개의 대응점이 필요한데 잘못된 대응점들이 많거나, 그 수 자체가 많을 경우 앞의 ①~④번 과정을 반복하는 횟수가 늘어나는 문제점이 발생하는데 이는 다음장에 설명된 대응점 필터링을 통해 해결 하도록 한다.

III. 대응점 추출 및 필터링

본 연구에서는 다양한 방향으로 설치된 카메라로부터 획득한 다중의 영상을 파노라마 이미지로 재구성할 때 발생하는 여러 가지 제한 사항을 해결할 수 있는 방안을 제시함을 목표로 하고 있으며, 일반적인 파노라마 영상 생성 과정에 본 연구를 통해 제안하는 방법을 추가한 과정이 그림 6에 전체 6단계로 나타나 보이고 있

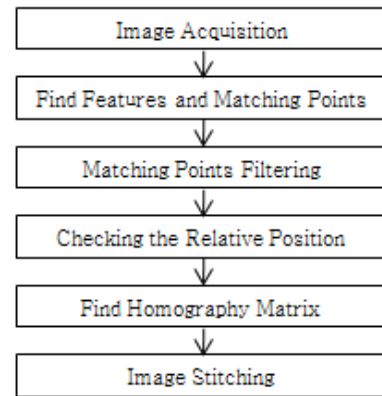


그림 6. 대응점 필터링을 이용한 파노라마 영상 생성 순서도

Fig. 6. Flowchart of generation of a panoramic image using the matching points filtering.

다. 먼저 카메라 및 기타 영상 입력 기기로부터 화상을 획득(Image Acquisition)한 후에 SURF 알고리즘을 이용하여 특징점 및 대응점을 검출(Find Features and Matching Points) 한다.

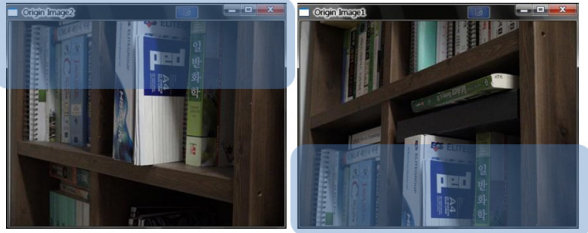
그 후, 검출된 대응점들을 필터링(Matching Points Filtering)하여 올바르지 않은 대응점들을 제거한다. 이렇게 필터링을 거친 대응점들을 사용하여 영상간의 상대적 위치를 확인하고 Homography Matrix를 구한 후 영상을 변환하고 병합(Stitching)하는 과정을 거친다.

1. 영상 입력순서 제한조건 해결

카메라로부터 획득한 영상으로 파노라마 영상을 생성하려면 해당 응용프로그램에 하나의 파노라마 영상을 구성할 두 개 이상의 이미지를 입력해야 한다. 일반적인 방법으로는 영상을 병합하는 과정에서 응용 프로그램의 입력 순서에 맞춰 영상을 입력해 주어야 한다. 하지만 모든 경우에 파노라마 영상을 구성할 응용 프로그램들이 영상 간의 상대적 위치를 능동적으로 알 수는 없으므로 이러한 제약사항은 카메라를 통해 획득한 영상을 사용하여 파노라마 영상을 생성할 때 사용자가 직접 상대적 위치를 구분지어 입력해야 한다는 단점을 내포하고 있다. 이러한 이유로 영상 입력순서에 상관없이 유연한 파노라마 영상 생성 프로그램이 필요하게 되는데, 본 연구에서는 필터링 된 대응점들의 좌표 데이터를 사용하여 응용프로그램 스스로 영상 간의 상대적인 위치를 판단할 수 있는 기능을 지원하도록 하였다. 두 영상이 하나의 파노라마 이미지로 합쳐지려면 중첩되는 부분이



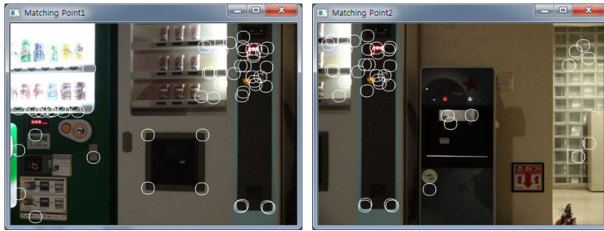
(a) 가로 파노라마 영상 구성



(b) 세로 파노라마 영상 구성

그림 7. 두 영상 간 중첩되는 영역

Fig. 7. Areas of overlapping between the two images.



(a) 왼쪽 영상

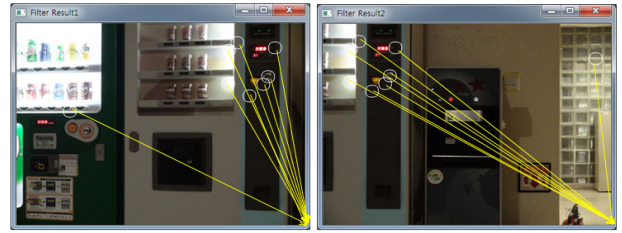
(b) 오른쪽 영상

그림 8. 필터링 과정을 거치지 않은 대응점 추출 영상

Fig. 8. Matching points images without the filtering process.

필요하고 대응점의 위치 또한 중첩된 영역 안에 있어야 한다. 두 영상 간 중첩된 영역에 대한 설명은 그림 7에 나타나 있다. 이 그림에서 (a)는 두 영상이 파노라마 이미지를 구성하였을 때 가로의 형태로 결과 영상을 구성하게 되는 예이며, (b)는 세로로 파노라마 영상을 구성하게 되는 예이다. 여기에 나타난 바와 같이 두 영상이 파노라마 이미지를 구성하는 데에는 반드시 중첩된 영역이 존재하며, 필터링 과정을 거쳐 얻어진 대부분의 대응점들의 좌표가 그 중첩영역 안에 존재하게 된다. 그림 8은 대응점 필터링이 필요한 이유를 보이기 위한 것인데 이것은 SURF 알고리즘을 사용하여 검출된 특징점으로부터 얻어진 대응점들의 위치 좌표를 나타낸 영상이다.

이 그림을 보면 실제 중첩된 영역 이외의 영역에서도 대응점이 존재하며 그 수 또한 상당히 많다는 것을 알 수 있다. 본 연구를 통해 제안하는 방법으로는 빠른 연산을 위하여 모든 대응점을 사용하는 대신 선택된 소수



(a) 왼쪽 영상

(b) 오른쪽 영상

그림 9. 대응점과 오른쪽 하단 모서리 사이의 연결선

Fig. 9. Lines between the matching points and bottom right corner.

의 대응점을 임의로 사용하게 되는데 그 이유는 대응점 필터링 과정이 없이는 정확한 연산이 힘들기 때문이다.

영상 입력 순서 제한 문제 해결 방안을 위해 위와 같은 방법으로 얻어진 대응점의 좌표 데이터는 그림 9에 나타난 것처럼 사용된다. 먼저 각 대응점의 위치 좌표를 구하고 이들과 영상의 오른쪽 하단 모서리 좌표를 사용하여 두 좌표 사이의 거리를 유클리드의 공식으로 구한다. 그림 9의 (a)는 파노라마 영상이 완성되면 상대적으로 왼쪽에 위치할 영상이며, (b)는 오른쪽에 위치할 영상이다. 그림 9의 (a), (b)에 나와 있는 선들은 필터링이 이루어진 후의 대응점 좌표에서부터 그림의 오른쪽 하단 모서리까지의 거리를 나타낸 것이다.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2)$$

$$Img_1, I_{1n}(x_n, y_n), M_1(x, y) \\ D_{1n} = \sqrt{(x - x_n)^2 + (y - y_n)^2} \quad (3)$$

$$Img_2, I_{2n}(x_n, y_n), M_2(x, y) \\ D_{2n} = \sqrt{(x - x_n)^2 + (y - y_n)^2} \quad (4) \\ (n = \text{대응점의 개수})$$

$$Avg1 = \frac{D_{11} + D_{12} + \dots + D_{1(n-1)} + D_{1n}}{n} \quad (5)$$

$$Avg2 = \frac{D_{21} + D_{22} + \dots + D_{2(n-1)} + D_{2n}}{n} \quad (6)$$

입력된 각 영상의 이러한 선들의 모든 길이를 합하여 영상의 상대적 위치를 확인하고 결정한다. 거리를 구하는 공식은 식 (2)부터 (6)에 나타나 있다. 식 (2)는 유클리드 공식을 통해 2차원 상의 두 점 사이의 거리를 구하

는 공식이며, d 는 두 점 (x_1, y_1) 과 (x_2, y_2) 의 거리를 구한 값을 뜻한다. (3)과 (4)는 영상에서 하나의 대응점 좌표에서 오른쪽 하단 모서리 부분까지의 거리를 구하는 식을 나타낸 것이다. 식 (3)의 $I_{1n}(x_n, y_n)$ 은 첫 번째 입력된 영상의 대응점 중 한 점에 대한 좌표이며, 식 (4)의 $I_{2n}(x_n, y_n)$ 은 두 번째로 입력된 영상의 대응점 좌표 중 하나를 나타낸 것이다. x_n 과 y_n 에서 n 은 대응점의 개수를 뜻하며, 연산과정에서 대응점의 개수만큼 1부터 n 까지 증가하게 된다. $M_1(x, y)$ 은 첫 번째 입력영상의 오른쪽 아래 모서리 좌표 값을 타나낸 것이고, $M_2(x, y)$ 는 두 번째로 입력된 영상의 오른쪽 모서리 아래 좌표를 나타낸다. 영상의 픽셀 크기 값, 예를 들어 첫 번째로 입력된 영상의 크기가 420×315 의 크기를 갖는다면 $M_1(x, y)$ 는 $M(420, 315)$ 가 되고 이 값은 고유 값이 된다. 식 (3), (4)에서 D_{1n} 과 D_{2n} 은 첫 번째, 두 번째 입력 영상의 임의의 대응점과 오른쪽 하단 모서리 간의 거리를 나타낸 것이다. 식 (5)와 (6)에서 $Avg1$ 과 $Avg2$ 는 첫 번째, 두 번째의 각 영상에서 구한 임의의 대응점들과 오른쪽 하단 모서리 간의 거리의 평균을 뜻한다. $Avg1$ 과 $Avg2$ 의 크기를 비교하여 영상의 상대적 위치를 정하게 되는데, $Avg1$ 이 $Avg2$ 보다 클 경우 첫 번째로 입력된 영상의 대응점들과 오른쪽 모서리 아래 부분간의 거리가 두 번째로 입력된 영상의 거리보다 더 길다는 결론이 나오게 된다. 이는 첫 번째 입력 영상의 겹치는 영역이 영상의 왼쪽 혹은 윗부분에 밀집해 있다는 것을 나타내며, 상대적으로 완성될 파노라마 이미지의 오른쪽 또는 아랫부분에 위치할 영상임을 뜻한다. 같은 맥락에서 보면 완성된 파노라마 이미지의 아랫부분에 위치하게 될 이미지도 이 경우에 해당된다. 반대로 $Avg1$ 이 $Avg2$ 보다 작다면, 첫 번째 영상은 완성된 파노라마 이미지의 왼쪽 혹은 윗부분에 위치하게 된다. 이러한 방법을 통해 영상들의 상대적 위치를 확인한 후 병합 과정에서 영상의 위치를 조정하게 되면 입력 영상 순서 제한 문제를 해결 할 수 있다.

2. 대응점 필터링

그림 10은 대응점 필터링의 필요성을 설명하기 위한 것이다. 이 그림을 보면 서로 적절치 않은 대응점들끼리 연결되어 있는 경우를 볼 수 있는데 이는 파노라마 영상을 구성하는데 치명적인 오류로 작용할 수 있다. 이러한

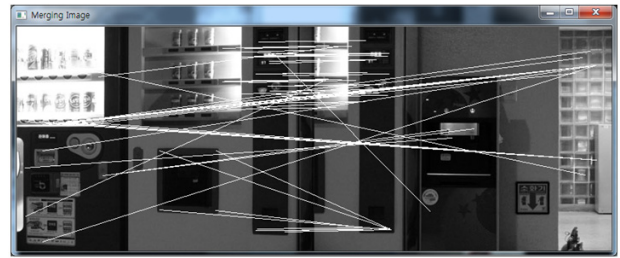


그림 10. 추출된 대응점끼리 선으로 연결된 화면

Fig. 10. Screen with lines connected between the extracted matching points.

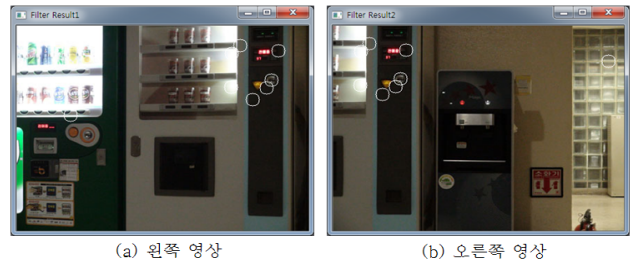


그림 11. 대응점 필터링 이후의 영상

Fig. 11. Images after the filtering of matching points.

$(x-1, y-1)$	$(x, y-1)$	$(x+1, y-1)$
$(x-1, y)$	(x, y) 특정좌표	$(x+1, y)$
$(x-1, y+1)$	$(x, y+1)$	$(x+1, y+1)$

그림 12. 특정 지점 좌표와 주변 3×3 마스크 영역 좌표

Fig. 12. Coordinates of a specific position and its 3×3 surrounded area.

문제점을 해결하기 위하여 SURF 알고리즘을 사용하여 검출한 특징점과 이를 이용하여 검출한 최초 대응점들의 좌표 데이터를 사용한다.

또한, 모든 대응점의 좌표 데이터는 각 영상마다 따로 저장되어 활용이 가능한데 원본영상에 대응점 좌표들을 대입하고 주변 픽셀 3×3 마스크 영역의 RGB값을 사용하여 정확한 대응점인지의 여부를 판단하게 된다.

필터링 과정을 수행하기 위해 3×3 마스크를 이용하여 대응점 좌표 주변의 RGB 값을 구하고 그 값의 연관성을 판단한다. 각 영상의 좌표, 예를 들어 한 지점의 특정 좌표가 (x, y) 라 하면 3×3 크기의 영역은 그림 12와 같은 좌표 영역으로 표현할 수 있다. 그리고 (x, y) 의

RGB 픽셀 값은 각각 $(x, y)_R$, $(x, y)_G$, $(x, y)_B$ 라 할 때 그 값을 구하는 공식은 수식 (7), (8), (9)에 나와 있는 것과 같이 표현된다.

$$(x, y)_R = (x-1, x-1)_r + (x, y-1)_r + (x+1, y-1)_r + \dots + (x, y+1)_r + (x+1, y+1)_r \quad (7)$$

$$(x, y)_G = (x-1, x-1)_g + (x, y-1)_g + (x+1, y-1)_g + \dots + (x, y+1)_g + (x+1, y+1)_g \quad (8)$$

$$(x, y)_B = (x-1, x-1)_b + (x, y-1)_b + (x+1, y-1)_b + \dots + (x, y+1)_b + (x+1, y+1)_b \quad (9)$$

수식 (7), (8), (9)를 사용하여 첫 번째 영상 내 하나의 특정 지점의 R, G, B 픽셀 값을 각각 구한 후 더한 값을 RGB_1 이라 하고 두 번째 입력된 영상에서 얻어진 결과 값을 RGB_2 라 한다. 이어서, RGB_1 과 RGB_2 의 차이를 구하여 얻어진 값을 $Result$ 라 지정하고 조명의 변화와 화각의 차이를 고려하여 적절한 오차 범위를 지정한다. 이는 영상 획득 시 촬영 시점 및 시간의 차이가 존재하므로 조명과 색상 부분에서 차이가 생기기 때문인데 이런 차이는 마스크를 사용하여 대응점 필터링을 하였을 때 옳게 계산된 대응점까지 제거하는 문제를 야기할 수 있다. 따라서 용인할 수 있는 오차 범위를 $\pm \alpha$ 라 정한 후, $Result$ 가 오차 범위 안에 있는지 판단하여 올바른 대응점 여부를 판단한다. 올바른 대응점 여부를 판단하는 과정에 대한 설명은 그림 13에 순서도를 이용하여 나타냈다. 여기서, $Result$ 값이 오차 범위 안에 있으면 적절한 대응점이라 판단되어 파노라마 영상을 생성하는데 사용되고, 그렇지 않은 경우에는 저장된 좌표 데이터에서 삭제되어 RANSAC 알고리즘 수행 과정에서 사용되지 않는다. 이와 같은 과정을 모든 대응점에 반복 수행하면 최종적으로 적절한 대응점들만이 남게 된다. 3×3 크기의 마스크 필터를 사용하여 RGB 연산을 한 것은 직관적인 계산을 위한 것이었으며 이로 인해 연산속도의 향상 측면에서 좋은 효과를 기대할 수 있다. 필터의 사이즈가 다양해지고 커지면 커질수록 전체 연산 속도는 오히려 더 느려질 수 있다. 또한 흑백 영상에 기반한 SURF 알고리즘을 수행한 후 해당좌표에 매칭되는 대응점의 좌표를 다시 컬러 영상에 적용시킴으로써 조명 및 색상으로 인한 문제를 최소화 할 수 있다.

이리하여 정확한 대응점을 검출하게 되면 파노라마

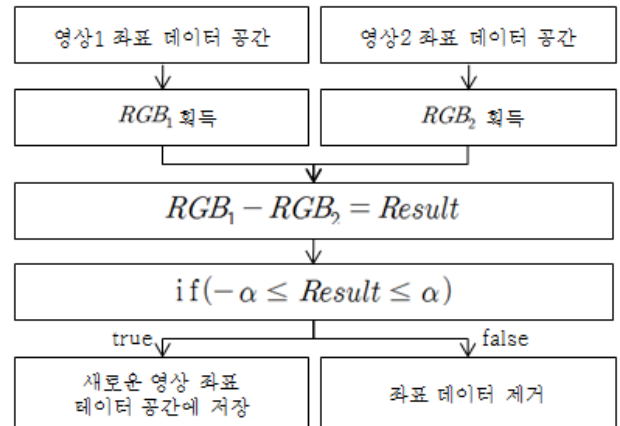


그림 13. 필터링을 통한 올바른 대응점 여부 판단 순서도

Fig. 13. Flowchart for determining correct matching points through the filtering.

영상 생성 시 발생할 수 있는 틀린 대응점으로 인한 오류를 최소화 할 수 있고 파노라마 영상의 구성 상 정확성과 생성 속도 측면에서 성능 향상을 기대할 수 있다.

IV. 영상처리 실험

본 연구를 통해 진행한 실험 환경은 표 1과 같다.

SONY HXR-MC50N 캠코더 두 대를 병렬과 직렬의 형태로 배치한 후, 각각의 기기로부터 파노라마 영상을 구성할 두 개의 영상을 획득하였으며, SONY NEX-5n 카메라를 이용해서는 MSE (Mean Square Error) 측정 등의 실험에 쓰일 완성된 파노라마 영상을 획득하였다. Open CV 2.3.1 버전은 SURF와 RANSAC 알고리즘을 이용한 파노라마 영상 생성 프로그램을 만드는데 사용되었다.

그림 14는 실험을 위해 Visual Studio 2010 작업환경

표 1. 실험 장비 및 환경

Table 1. Experimental devices and environments.

Camcorder	SONY HXR-MC50N 캠코더 / SONY NEX-5n
Program	Open CV 2.3.1
Tool	Visual Studio 2010
CPU	Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz
OS	Windows7 Enterprise K 32bit

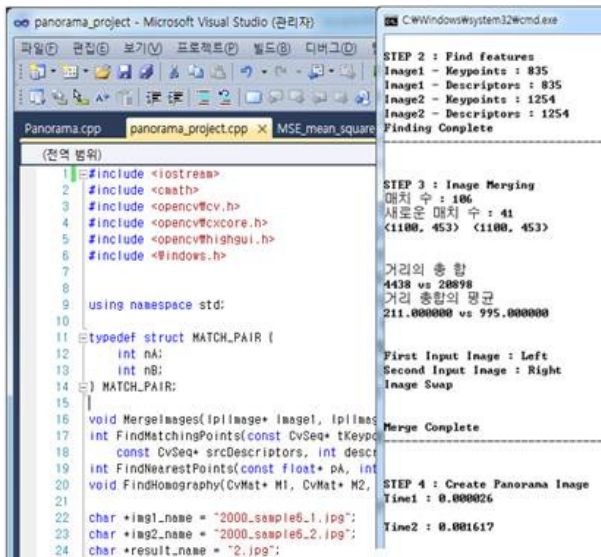


그림 14. 파노라마 이미지 생성 프로그램(좌)과 실행화면(우)

Fig. 14. Generation Program of a panorama image(Left) and the execution screen(Right)

에서 제작한 파노라마 생성 프로그램(좌)과 실행화면(우)을 보이고 있다. 실행화면(우)을 통해 2차 대응점 필터링에 필요한 자료와 순서제한 문제 해결에 필요한 대응점 좌표 데이터 등을 확인할 수 있다.

1. 영상 입력순서 제한조건 해결실험

그림 15와 16의 (a)와 (b)는 파노라마 영상을 구성할



그림 15. 영상 입력순서 제한문제 해결 실험 결과 1
Fig. 15. Experimental result 1 for solving condition of image input order.

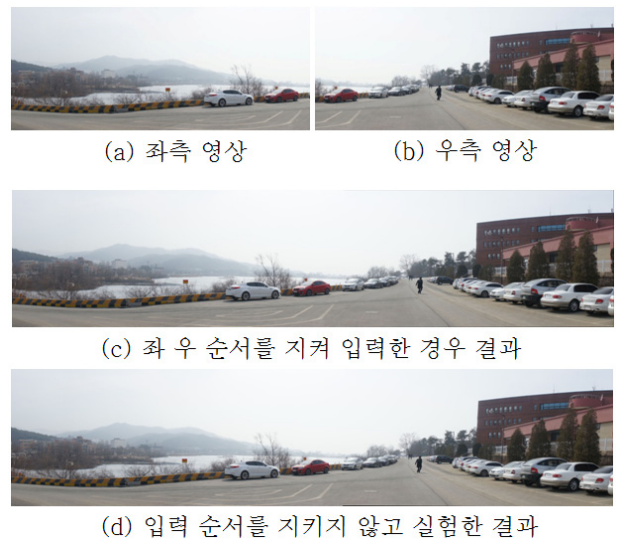


그림 16. 영상 입력순서 제한문제 해결 실험 결과 2
Fig. 16. Experimental result 2 for solving condition of image input order.

때 각각 왼쪽과 오른쪽에 위치하게 될 영상이다. 이 논문의 III장 1절에 소개된 내용대로 진행한 실험 결과가 그 다음에 이어서 보여 지고 있다. 그림 15와 16의 (c)와 (d)는 각각 영상 입력 순서를 지켜서 입력한 경우와 그렇지 않은 경우에 대한 결과를 보이고 있다. 앞서 말했듯이 기존의 방법들은 파노라마 영상 생성 시 상대적 위치를 미리 확인하여 입력해야 하는 제한사항이 있었다. 하지만, 본 연구를 통해 소개된 기법을 이용하여 이러한 제한문제 해결에 도움이 되고자 하였다. 파노라마 이미지 생성 후 그 성능 분석을 위하여 MSE 값을 측정하였고 이에 대하여 다음 장에서 토론하기로 한다.

2. 대응점 필터링

그림 17과 18의 영상들은 오류 대응점 필터링을 거치지 않고, SURF 알고리즘을 통해 획득한 대응점들의 위치와 이들을 연결한 그림을 나타내고 있다. 오류를 최소화하여 파노라마 영상을 생성하려면 그림 19에 표시된 영역에 좌우 두 영상이 겹치는 이른바 중복 영역이 나타나야하며 여기에 대부분의 대응점이 존재해야 한다. 그림 17과 18의 (a), (b)에는 영상 중복 영역 이외에도 오류 대응점들이 존재하는 것을 알 수 있다. 따라서 이러한 오류 대응점 제거를 위한 필터링 수행이 필요하며 그 결과가 그림 20과 그림 21에 나와 있다.

앞서 III장에서 설명한 내용을 기반으로 하여 오류 대응점 필터링을 수행하였으며, 이 그림들을 보면 대부분

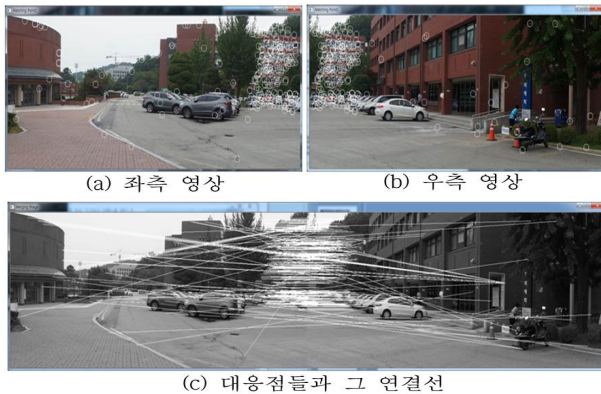


그림 17. 오류 대응점 필터링 과정을 거치기 전 대응점과 연결선 결과 1

Fig. 17. Result 1 of matching points and connected lines before error filtering.



그림 18. 오류 대응점 필터링 과정을 거치기 전 대응점과 연결선 결과 2

Fig. 18. Result 2 of matching points and connected lines before error filtering.

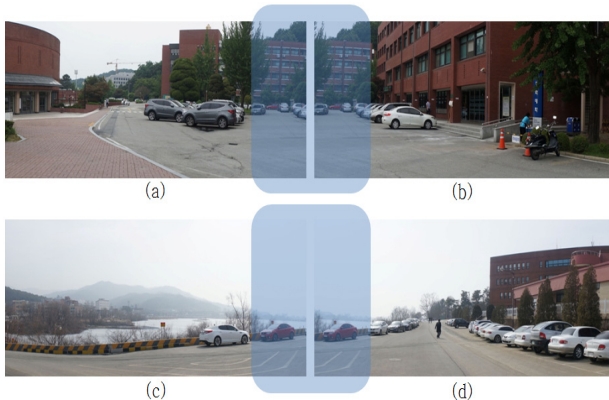


그림 19. 파노라마 영상 구성을 위한 두 영상의 중복영역

Fig. 19. Overlapped area of two images for panorama image processing: (a) Fig. 17(a), (b) Fig. 17(b), (c) Fig. 18(a), (d) Fig. 18(b).



그림 20. 오류 대응점 필터링 수행 후 결과 1

Fig. 20. Result 1 after the matching points filtering:

(a) Fig. 17(a), (b) Fig. 17(b), (c) Panorama image



그림 21. 오류 대응점 필터링 수행 후 결과 2

Fig. 21. Result 2 after the matching points filtering :

(a) Fig. 18(a), (b) Fig. 18(b), (c) Panorama image

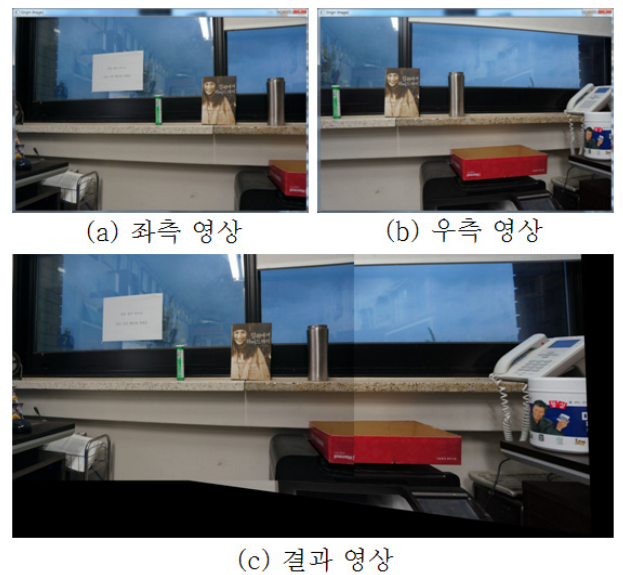


그림 22. 영상 복잡도에 따른 대응점 필터링 후 결과 1

Fig. 22. Result 1 after the matching points filtering with respect to the image's complexity.

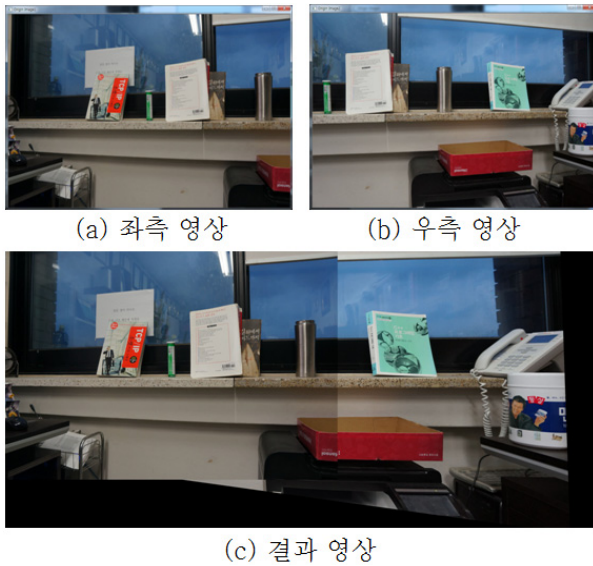


그림 23. 영상 복잡도에 따른 대응점 필터링 후 결과 2
Fig. 23. Result 2 after the matching points filtering with respect to the image's complexity.

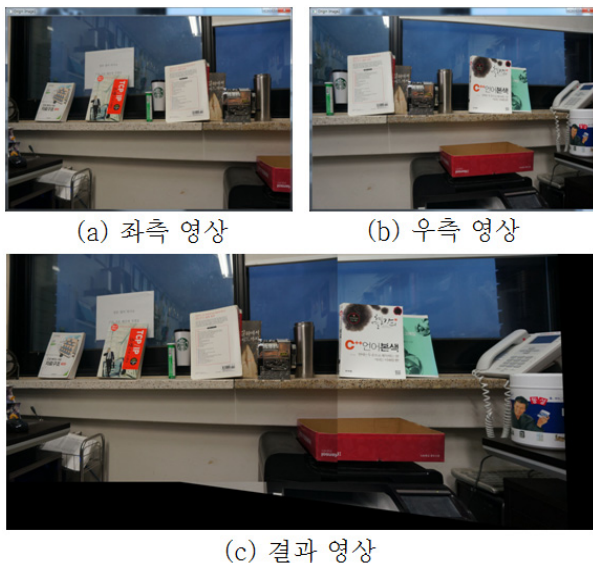


그림 24. 영상 복잡도에 따른 대응점 필터링 후 결과 3
Fig. 24. Result 3 after the matching points filtering with respect to the image's complexity.

의 대응점들이 중복 영역 안에 존재하고, 서로 다른 부분을 가리키는 대응점들이 거의 없는 것을 볼 수 있다. 이렇게 하여 오류 대응점 필터링을 거친 후 생성된 파노라마 영상은 원본 영상과의 MSE 값 비교를 통해 영상의 완성도를 분석할 수 있다. 이를 위한 성능 분석과 정확도 판단 등은 다음 장에서 토론하기로 한다. 그림 22, 23, 24는 실험에 사용된 다른 형태의 이미지와 그 결과를 보이고 있다.

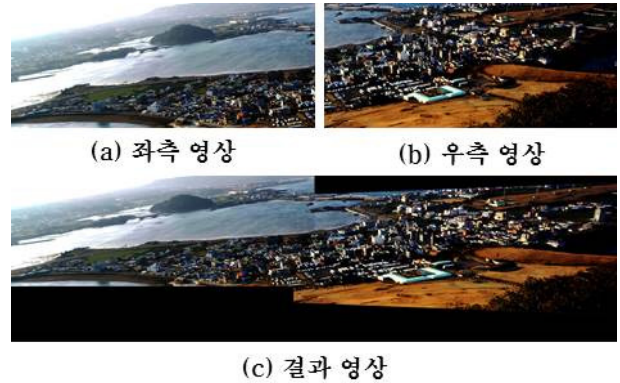


그림 25. 비수평인 이미지에 적용한 필터링 후 결과
Fig. 25. Result after the matching points filtering of an image that is out of the horizontal with the ground.

그림 22는 세 종류의 영상 중 가장 간단한 형태의 예를 보이고 있고 그림 23은 중간 정도의 복잡도를 띄고 있으며 그림 24는 가장 복잡한 정도의 영상을 이용한 파노라마 이미지 처리의 결과를 보이고 있다.

본 연구를 통해 제안한 방법의 강인성을 시험하고자 그림 22, 23, 24의 경우 좌측과 우측을 구성하는 이미지를 획득할 때 카메라의 촬영 각도를 의도적으로 서로 맞지 않도록 틀어지게 하였다.

각 그림 22, 23, 24의 (a)와 (b)를 보면 책이나 스텐레스 물통이 세워져 있는 창문 선반의 수평 각도가 약간 비틀어져 있어 파노라마 이미지 처리에 다소 어려움이 예상되도록 하였는데 이를 통하여 논문에 제시된 방법의 기능 및 효율성을 테스트 해 볼 수 있도록 하였다.

각 그림 22, 23, 24의 (c)를 보면, 수평을 이루는 창틀의 선반과 사진 윗부분의 커튼 끝자락 선의 이어진 부분을 통해 파노라마 영상처리 결과를 볼 수 있는데 직선으로 이어져야 할 부분이 다소 왜곡되어 연결되어 있는 것을 볼 수 있다. 각도를 이용한 분석은 다음 장인 결과분석 및 토론 절에서 다루기로 한다. 마지막으로 그림 25는 땅과 수평을 유지하지 않은 상태에서 찍은 사진에 대한 필터링 이후 파노라마 영상 생성 결과를 보이고 있다. 높은 곳에서 풍경을 찍은 사진들을 이용했는데 땅과 수평이 맞지 않고 좌우 각도가 틀어져 있어서 결과 영상이 만들어 진 후 위와 아랫부분에 까만색의 빈 공간이 생성되어 있음을 알 수 있다. 이를 통해 대응점 필터링 이후의 기법을 이용한 파노라마 이미지 생성 알고리즘이 비수평인 입력 이미지의 경우에도 적절히 동작함을

표 2. 그림 20부터 그림 25까지에 제시된 영상의 대응점 개수 변화량

Table 2. Changing rate of the image's matching points presented in Fig. 20 through Fig. 25.

대응점 개수 변화				
영상	크 기	필터링 전	필터링 후	감소율(%)
Fig. 20	1100×453	304	10	96.71
Fig. 21	1100×453	106	41	61.32
Fig. 22	1000×664	372	17	95.43
Fig. 23	1000×664	326	111	65.95
Fig. 24	1000×664	398	25	93.71
Fig. 25	1100×450	41	12	70.7

알 수 있다.

표 2는 그림 20부터 25의 결과에 제시된 영상으로부터 추출된, 필터링 전과 후의 대응점 변화량을 표로 나타낸 것이다. 그림 20의 경우 대응점의 개수가 필터링 전후를 통하여 304개에서 10개로 줄어들어 96.7%의 감소율을 보였고, 그림 21은 61.3%, 그림 22와 23, 24, 25는 각각 95.4%, 65.9%, 93.7%, 70.7%로 평균 80.6%의 대응점 감소율을 보였다. 파노라마 영상 생성 과정에서 대응점의 개수는 해당 알고리즘의 수행 속도, CPU 점유율 그리고 결과영상의 정확도 등에 많은 영향을 끼치게 된다. 다음 장에서 실험 결과에 대한 분석과 토론을 진행한다.

V. 결과 분석 및 토론

실험 결과의 분석을 다루는 이번 장에서는 파노라마 영상 구성을 위한 영상 입력순서 제한조건 문제 해결에 관한 내용을 제시하고 다양한 영상의 크기별 대응점 필터링 전후의 개수 감소율, 수행시간, CPU 점유율, 그리고 결과 영상에 포함된 부분 영상의 각도를 이용한 정확도 등을 측정한다.

1. 영상 입력순서 제한조건 해결실험 결과

그림 26은 그림 15의 (a)와 (b) 두 영상을 이용하여 좌우의 위치 판단을 진행한 실험 결과를 보이고 있다.

이는 첫 번째로 입력된 영상에 존재하는 필터링 후의 각 대응점들과 해당 이미지의 우측 하단 모서리와의 거

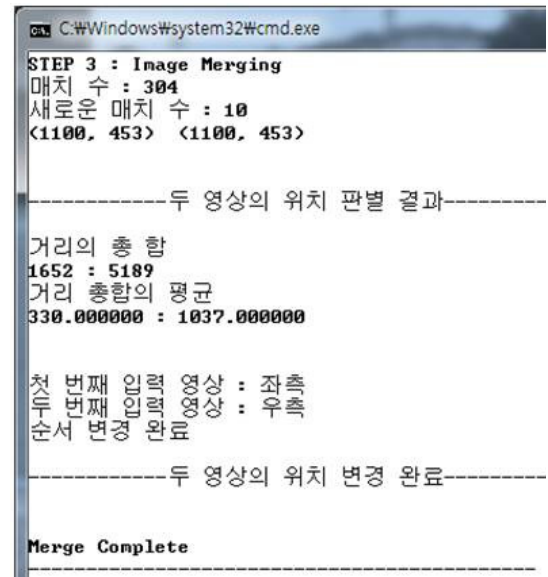


그림 26. 영상 위치 판단 실행 결과화면

Fig. 26. Display of execution result of image's position decision.

리 평균이 330이라는 것이며 두 번째로 입력된 영상의 평균거리는 1037이므로 첫 번째로 입력된 영상이 좌측에 위치하게 될 영상이라 판단하는 실험 내용이다. 따라서 이러한 결과를 이용하여 파노라마 이미지를 구성할 때 좌우측 배치 또는 상하 배치를 정확히 할 수 있다.

2. 대응점 추출 및 필터링 결과

가. 대응점 감소율 분석 및 평가

그림 20부터 25에 제시된 모든 영상을 사용한 대응점 필터링의 결과는 표 3과 같으며, 그림 20, 21은 각각 영상의 가로 크기가 1100, 2750, 5500 픽셀인 결과를 보이고 있고 그림 22, 23, 24는 각각 1000, 2500, 4000 픽셀인 경우의 크기, 그림 25는 1100, 2750, 4400 픽셀인 크기 별로 나누어 실험한 결과를 보이고 있다. 특징점은 각 영상으로부터 SURF 알고리즘을 사용하여 검출한 것이며 그 개수를 표현한 것이다. 개수가 많을수록 영상의 특징이 많다는 것을 의미하는데 이것은 영상의 복잡한 정도와 관련이 있다. 즉, 특징점이 많으면 많을수록 영상이 그만큼 복잡하다는 것을 의미한다. 그림 20과 21의 영상은 평균 76.1%의 대응점 감소율을 보였고, 그림 22부터 24까지의 영상들은 평균 90.8%, 그림 25는 평균 84.8%의 대응점 감소율을 보여 오류 대응점 필터링의 수행결과가 매우 양호함을 알 수 있다.

표 3. 대응점 변화율

Table 3. Changing rate of the matching points.

대응점 변화율							
영 상			특징점 개 수		대응점 개 수		감소율 (%)
			(좌)	(우)	필터링 전	필터링 후	
그림 20	크 기	1100x453	1284	1468	304	10	96.7
		2750x1133	6521	6450	1382	30	97.8
		5500x1813	14631	13538	3471	3197	7.8
그림 21	크 기	1100x453	835	1254	106	41	61.3
		2750x1133	4177	5636	397	11	97.2
		5500x1813	11077	13741	4704	170	96.3
그림 22	크 기	1000x664	1050	1165	372	17	95.4
		2500x1661	3656	4045	1124	6	99.4
		4000x2658	6728	7642	1994	51	99.9
그림 23	크 기	1000x664	1161	1256	326	111	65.9
		2500x1661	4203	4918	1039	328	68.4
		4000x2658	8781	9960	1956	28	98.5
그림 24	크 기	1000x664	1449	1518	398	25	93.7
		2500x1661	5198	5949	1232	37	96.9
		4000x2658	10524	11924	2378	6	99.7
그림 25	크 기	1100x450	643	1006	41	12	70.7
		2750x1126	2931	4850	167	16	90.4
		4400x1800	5592	9446	334	22	93.4

표 4. RANSAC 수행속도 비교

Table 4. Comparison of RANSAC performance speed.

Filtering 및 RANSAC 수행 속도 비교 (단위 : 10^{-2} sec)						
영 상	크 기	Filtering	RANSAC	전체	RANSAC Only	감소율 (%)
그림 20	1100x453	0.006	0.139	0.146	0.264	44.7
	2750x1133	0.034	0.137	0.172	0.661	73.9
	5500x1813	0.161	0.840	1.002	1.228	18.3
그림 21	1100x453	0.002	0.143	0.145	0.280	48.1
	2750x1133	0.009	0.136	0.146	0.434	66.2
	5500x1813	0.176	0.192	0.368	1.334	72.3
그림 22	1000x664	0.008	0.270	0.278	0.714	60.1
	2500x1661	0.028	0.117	0.145	1.722	91.5
	4000x2658	0.051	0.785	0.837	3.207	73.9
그림 23	1000x664	0.007	1.131	1.139	1.619	29.6
	2500x1661	0.027	1.354	1.381	2.019	31.5
	4000x2658	0.084	1.364	1.448	3.104	53.3
그림 24	1000x664	0.009	0.329	0.338	1.012	66.5
	2500x1661	0.031	0.960	0.991	2.076	52.2
	4000x2658	0.129	0.263	0.393	4.04	90.2
그림 25	1100x450	0.098	0.368	0.369	6.488	94.3
	2750x1126	0.478	0.387	0.392	1.519	74.1
	4400x1800	1.088	0.357	0.368	1.082	65.9

나. RANSAC 알고리즘 수행속도 분석 및 평가

표 4는 오류 대응점 필터링 단계를 거친 경우와 그렇지 않은 두 가지 경우에 대하여 대응점들을 사용해서 RANSAC 알고리즘을 수행한 후 그 속도를 측정, 비교한 결과를 보이고 있다. 이 실험에 사용된 각 영상들의 수행속도 결과를 보면 오류 대응점 필터링을 통해 얻어진 대응점들을 이용하여 RANSAC 알고리즘을 수행한 경우가 더 짧은 시간을 소요한 것을 알 수 있다. 본 연구를 통하여 제시된 오류 대응점 필터링에 걸리는 시간 또한 전체 RANSAC 알고리즘 수행시간에 포함되어 측정되었다. 그림 20, 21의 경우 수행속도가 평균 53.9% 감소한 것으로 나왔고, 그림 22부터 24까지의 경우 평균 60.9%, 그림 25는 평균 78.1%의 감소율을 보였다. 이는 오류 대응점 필터링을 포함한 RANSAC 알고리즘 전체 수행속도의 감소라는 관점에서 분석해 본다면 매우 고무적인 결과라 할 수 있다.

다. CPU 점유율 감소 분석 및 평가

표 5는 오류 대응점 필터링 과정을 포함한 RANSAC 알고리즘을 수행했을 때와 단지 RANSAC 알고리즘 하나만을 수행했을 때의 컴퓨터 중앙처리장치, 즉 CPU의 수행 점유율을 보이고 있다. 이 결과를 보면 CPU 점유율 역시 필터링 과정을 포함한 RANSAC 알고리즘 수행시 상당량 감소한 것으로 나타나 있고 전체 평균 52.5%의 감소율을 보이고 있다.

이는 본 연구를 통하여 제시된 기법으로 파노라마 영상 처리를 수행했을 시 CPU의 데이터 처리 부담 또한 많이 줄어들음을 알 수 있다.

표 5. RANSAC 수행 시 CPU 점유율

Table 5. CPU occupation rate of RANSAC algorithm.

CPU 점유율 (%)					
영 상			필터링 후 RANSAC	RANSAC Only	감소율 (%)
그 림 20	크 기	1100x453	0.086	0.126	31.7
		2750x1133	0.014	0.032	56.2
		5500x1813	0.004	0.01	60.0
그 림 21	크 기	1100x453	0.1	0.116	13.7
		2750x1133	0.02	0.026	23.0
		5500x1813	0.004	0.018	77.7
그 림 22	크 기	1000x664	0.162	0.198	18.1
		2500x1661	0.01	0.076	86.8
		4000x2658	0.022	0.074	70.2
그 림 23	크 기	1000x664	0.128	0.226	43.3
		2500x1661	0.044	0.072	38.8
		4000x2658	0.016	0.038	57.8
그 림 24	크 기	1000x664	0.08	0.176	54.5
		2500x1661	0.034	0.068	50
		4000x2658	0.012	0.062	80.6
그 림 25	크 기	1100x450	0.07	0.17	58.8
		2750x1126	0.044	0.075	41
		4400x1800	0.01	0.06	83.3

라. 정확도 분석 및 평가

표 6은 원본 파노라마 영상과 본 논문에서 제시한 방법을 사용하여 재구성한 파노라마 영상간의 차이를 알아보기 위해 평균 제곱 오차 값인 MSE(Mean Square Error)를 사용하여 정확도를 측정하는 것이다. MSE 값을 측정하기 위해, 오류 대응점 필터링을 포함한 파노라마 영상처리를 수행하여 영상 하나를 먼저 생성했고, 그 다음에 필터링을 포함하지 않은 기존의 방법을 사용하여 파노라마 영상을 생성한 후, 각각의 생성된 이미지와 원본 파노라마 영상 간의 차이를 구하였다. 두 경우의 MSE 값을 비교, 분석해보면 그 값의 차이가 그리 크지 않아 생성된 파노라마 이미지의 완성도가 예리 측정의 관점에서 매우 비슷한 정도임을 알 수 있다. 그림 27은 원본 파노라마 영상이 없는 그림 22, 23, 24의 경우 실험의 정확도를 측정하기 위해 좌우 두 영상을 이용하여 RANSAC 알고리즘을 수행하고 Homography Matrix값을 계산한 후 최종으로 파노라마 영상을 구성한 것의 일부분을 보이고 있다.

표 6. 그림 20, 21의 MSE 측정 및 비교

Table 6. Comparison of MSE for Fig. 20 and Fig. 21.

MSE				
영 상			필터링 포함	RANSAC only
그 림 20	크 기	1100x453	12.283612	11.312818
		2750x1133	0.927365	0.843366
		5500x1813	0.682813	0.827647
그 림 21	크 기	1100x453	8.386376	6.459141
		2750x1133	1.382654	0.559321
		5500x1813	0.435663	0.537658



그림 27. 정확도 측정 및 비교

Fig. 27. Measurement and comparison of accuracy :

(a) With the filtering, (b) Without the filtering

좌우 영상의 빨간색 선이 파노라마 영상을 구성하면서 경계선에서 각도의 차이가 없이 거의 0° 인 모습을 보여야 하는데 여기서 그렇지 못한 것은 계산된 대응점과 Homography Matrix가 정확하지 못하기 때문이라고 분석할 수 있다. 즉, 그림 27에서 보이는 윗부분의 하얀 커튼 연결 부분이 뒤틀리지 않고 일직선으로 연결되어 보일수록 각도의 차이는 0° 에 가까울 것이고 이렇게 될수록 구성된 파노라마 영상의 정확도는 높다고 할 것이다.

표 7은 그림 27에 보이는 붉은색 선을 이용하여 그림 22, 23, 24에 적용시켜 생성된 파노라마 영상의 정확도를 측정, 비교 분석한 것이다. 이 표에서는 오류 대응점 필터링을 포함해서 수행했을 경우와 그렇지 않은 경우에 대한 결과를 제시하여 비교하였다. 필터링을 포함하여 실험한 결과 정확도의 평균값은 각도차이의 경우 2.7° , 그리고 일치율은 95.1%를 보였으며, 필터링이 포함되지 않았을 경우의 각도차이 평균값은 3.7° 그리고 일치율은 93.3%를 보였다.

실험을 통하여 오류 대응점 필터링을 통과한 대응점들을 이용하여 파노라마 영상 생성을 시도했을 시 그 정

표 7. 그림 22, 23, 24의 정확도 측정 및 비교
Table 7. Measurement and comparison of accuracy for Fig. 22, Fig. 23 and Fig. 24.

영 상			정확도 (degree)		일치율 (%)	
			필터링 포함	RANSAC only	필터링 포함	RANSAC only
그 림 22	크 기	1000x664	3	4	94.6	92.8
		2500x1661	3	4	94.6	92.8
		4000x2658	2.5	4	95.5	92.8
그 림 23	크 기	1000x664	3	4	94.6	92.8
		2500x1661	2.5	4	95.5	92.8
		4000x2658	3	3.5	94.6	93.7
그 림 24	크 기	1000x664	2.5	3	95.5	94.6
		2500x1661	2	3	96.4	94.6
		4000x2658	3	4	94.6	92.8

확도 또한 그렇지 않았을 경우보다 더 높다는 것을 알 수 있다.

VI. 결 론

파노라마 영상은 한 대의 카메라에서 얻을 수 있는 영상의 한계를 극복할 수 있는 좋은 방법으로서 광각의 영상이 요구되는 다양한 분야에서 응용되고 있다. 제한된 환경 내에서 파노라마 영상을 제작하는 데에는 다양한 프로그램 환경과 알고리즘이 사용되는데, 통상적으로 많이 사용되는 방법은 특징점 검출을 위한 SURF 알고리즘과 영상 변화를 위해 Homography Matrix를 구해내는 RANSAC 알고리즘이 있다. 본 논문에서는 SURF와 RANSAC 알고리즘을 이용한 파노라마 영상 제작 시 특징점 검출 후 남은 대응점에 의해 나타나는 몇 가지 제한사항을 오류 대응점 필터링을 통해 해결하는 방법을 연구하였고, 파노라마 영상 구성에 불필요한 대응점들은 제거하였다. 이로 인한 제한사항 해결 및 성능향상에 대한 내용은 다음과 같이 정리될 수 있다. 첫 째, 대응점 필터링을 통해 오류가 있는 대응점끼리의 대응을 방지하였다. 둘 째, 오류 대응점 필터링을 통해 줄어든 대응점 개수로 인해 파노라마 영상처리 전체 수행 속도가 향상되었으며, 작업 수행 시 CPU 점유율이 현저히 감소하였다. 마지막으로, 파노라마 영상 구성을 위해 상하 또는 좌우 영상을 입력할 시 순서를 지

켜야 하는 제한 문제를 해결하였고, 파노라마 영상 생성 후 그 정확도를 측정하기 위하여 영상 내의 직선을 이용, 정합 부분의 오류 각도를 비교분석하였다. 또한, MSE를 계산하여 원본영상과의 차이를 비교하였고 생성된 영상의 완성도를 측정하였다. 본 논문을 통하여 서술된 입력영상 순서제한 문제해결에 관한 기법과 오류 대응점 필터링 방식은 실시간 고화질 파노라마 영상 생성이 필요한 분야의 성능향상에 기여할 수 있으며, 그 밖에 다양한 컴퓨터 비전과 그래픽 분야에서 응용이 가능하다. 향 후, 다량의 영상을 이용하여 상하좌우를 자체적으로 판단하고, 최종적으로 대형 파노라마 영상을 생성해내는 연구를 진행할 계획이다.

REFERENCES

- [1] Daehyun Kim, 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, Nov. 2003.
- [2] Sehwan Kim, Kiyoun Kim, Woontack Woo, "Multiple Camera Calibration for Panoramic 3D Virtual Environment," IEIE Journal(CD), vol. 41, no. 2, pp. 137-148, Mar. 2004.
- [3] Seung-Hyen Rhee, Jae Ho Lee, Sang-Gil Lee, and Moon Gi Kang, "Panoramic image reconstruction based on a camera motion model," KOSBE Journal, pp. 41-46, 1999.
- [4] Jun-Seong Joe, Bum-jong Lee, and Jong-Seung Park, "Creating Full View Panorama Image from Multiple Images," KIIE Journal, vol. 34, no. 2(B), pp. 162-166, 2007.
- [5] Zhihui Xiong, Maojun Zhang, Yunli Wang, Tun Li, and Kijun Li, "Fast Panorama Unrolling of Catadioptric Omni-Directional Images for Cooperative Robot Vision," Computer Supported Cooperative Work in Design, 11th International Conference on, DOI 10.1109/SCCWD.4281594, pp. 1100-1104, Apr. 2007.
- [6] S. Kim and W. Woo, "Panoramic image-based virtual environment using the 3D generation," IPIU, vol. 13, pp. 111-116, 2003.
- [7] Y. Seunbeom, J. HyeokJun and H. Chijeion, "Features-based panoramic image combinations," KICS Journal, vol. 35, no. 8, pp. 179-184, Sep.

- 2010.
- [8] S. Richard, "Image Alignment and Stitching : A Tutorial," *Journal of foundations and trends in computer graphics and vision*, vol. 2, no. 1, Jan. 2006.
 - [9] J. M. Morel and G. Yu, "ASIFT : A New Framework for Fully Affine Invariant Comparison," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, 2009.
 - [10] K. Mikolajczyk and C. Schmid, "Scale and Affine invariant interest point detectors," *In IJC*, vol. 60, no. 1, pp. 63-86, 2004.
 - [11] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," *In BMVC* pp. 384-393, 2002.
 - [12] T. Tuytelaars and K. Mikolajczyk, "Local Invariant Feature Detectors," *In CVG*, vol. 3, no. 1, pp. 1-110, 2008.
 - [13] M. Brown and D. Lowe, "Automatic Panoramic Image Stitching using Invariant Features," *International Journal of Computer Vision*, vol. 74, no. 1, pp. 59-73, 2007.
 - [14] Chanil Park, Suhyun Lee, Yongjin Jeong, "A Hardware Design of Feature Detector for Realtime Processing of SIFT(Scale Invariant Feature Transform) Algorithm in Embedded Systems," *IEIE Journal(SD)*, vol. 46, no. 3, pp. 86-95, Mar. 2009.
 - [15] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," *Computer Vision of Lecture Notes in Computer Science*, Springer, vol. 3951, pp. 404-417, 2006.
 - [16] Sangwon Seo, Soowoong Jeong, Yunsang Han, Jongsoo Choi, Sangkeun Lee, "Efficient Homography Estimation for Panoramic Image Generation," vol. 50, no. 8, pp. 215-224, Aug. 2013.
 - [17] Luo Juan, Sungsik Shin, Hyun Ju Park, and Ou Bong Gwun, "Stitching for panorama based on SURF and Multi-band Blending," *KMMS Journal*, vol. 14, no. 2, pp. 201-209, 2011.
 - [18] Taehyub Kim, Yoonseok Choi, Bodam Nam, Hyunki Hong, "Tele-presence System using Homography-based Camera Tracking Method," *IEIE Journal(CI)*, vol. 49, no. 3, pp. 27-33, May 2012.
 - [19] Jayhyun Im, Joonki Paik, "Feature-Based Panoramic Background Generation for Object Tracking in Dynamic Video," *IEIE Journal(SP)*, vol. 45, no. 6, pp. 108-116, Nov. 2008.
 - [20] K. MinGoo, C. Wongook, and M. Seounbin, "SURF feature extraction algorithm for face recognition research," *IEIE Journal(CI)*, vol. 48, no. 3, pp. 46-53, May 2011.
 - [21] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting applications to image analysis and automated cartography," *Proc. Image Understanding Workshop*, pp. 71-88, Apr. 1980.
 - [22] Son Jung-In, Kang Minsung, Sohn Kwanghoon, "View invariant image matching using SURF," *KOSBE Conference*, pp. 222-225, Aug. 2011.
 - [23] O. Chum, T. Pajdla, and P. Sturm, "The geometric error for homographies," *Computer Vision and Image Understanding*, vol. 97, no. 1, pp. 86-102, 2005.

— 저 자 소 개 —



김 정 호(학생회원)
2012년 단국대학교 멀티미디어공
학과 학사 졸업.
2014년 단국대학교 대학원 컴퓨터
학과 석사 졸업.

<주관심분야 : Panorama Image Processing,
Mobile Platform, Cloud Computing, Mobile
Multimedia Processing>



김 대 원(정회원)-교신저자
1993년 중앙대학교 공과대학 전자
공학과 학사 졸업.
1996년 University of Southern
California, Electrical and
Computer Engineering,
M.S. 졸업.

2002년 Iowa State University, Electrical and
Computer Engineering, Ph. D. 졸업.
현재 단국대학교 멀티미디어공학과 부교수
<주관심분야 : Mobile Platform, Application
S/W, Digital Multimedia Data Processing,
Non-Destructive Evaluation, Ultrasonic Signal
Processing, Source Localization>