

## 다중 기술자를 이용한 잘못된 특징점 정합 제거

김재영\*, 전희성\*

# Filtering Feature Mismatches using Multiple Descriptors

Jae-Young Kim\*, Heesung Jun\*

### 요약

이미지 기술자(descriptor)를 이용한 정합은 최근까지 컴퓨터 비전과 패턴인식 분야에서 사용되고 있는 강력한 정합 방법이다. 그러나 3차원 시점이 변화되거나 밝기가 변화된 이미지, 반복된 패턴이 포함된 이미지 등에서 잘못된 정합들이 발생한다. 본 논문에서는 반복된 패턴이 포함되어 있는 이미지에서 잘못된 정합들이 많이 발생하는 문제점에 대해 기술하고 이를 분석하여 잘못된 정합들을 제거할 수 있는 방법을 제안한다. MDMF(Multiple Descriptors-based Mismatch Filtering) 방법은 각 특징점에 대해 인접한 여러 개의 특징점들의 기술자들을 사용하여 다중 기술자를 생성한 후 이를 활용하여 잘못된 정합들을 제거한다. 실험에서는 크기 변환, 회전 변환, 어파인 변환에 대해 기존 SIFT와 ASIFT의 정합율을 MDMF를 이용해 제거한 정합율과 비교하여 MDMF가 잘못된 정합을 성공적으로 제거할 수 있음을 보였다.

▶ Keywords : 특징점 정합, 특징점 기술자, 특징점 추출, 물체 인식, 반복 패턴

### Abstract

Feature matching using image descriptors is robust method used recently. However, mismatches occur in 3D transformed images, illumination-changed images and repetitive-pattern images. In this paper, we observe that there are a lot of mismatches in the images which have repetitive patterns. We analyze it and propose a method to eliminate these mismatches. MDMF(Multiple Descriptors-based Mismatch Filtering) eliminates mismatches by using descriptors of nearest several features of one specific feature point. In experiments, for geometrical transformation like scale, rotation, affine, we compare the match ratio among SIFT, ASIFT and MDMF, and we show that MDMF can eliminate mismatches successfully.

▶ Keywords : feature matching, feature descriptor, feature extraction, object recognition, repetitive pattern

•제1저자 : 김재영 •교신저자 : 전희성

•투고일 : 2013. 11. 15, 심사일 : 2013. 12. 04, 게재확정일 : 2013. 12. 19.

\* 울산대학교 전기공학부(School of Electrical Engineering, University of Ulsan)

※ 이 논문은 2012년 울산대학교 연구비에 의하여 연구되었음.

## I. 서론

컴퓨터 비전에서 서로 다른 이미지에 있는 두 점군들의 정합은 중요하고 복잡한 문제이다. 이와 같은 문제들을 해결하기 위해 주로 사용되는 기술은 특징점 정합이다. 특징점 정합은 3차원 복원(3D Reconstruction), 물체인식, 얼굴인식, 이미지 스티칭, 스테레오 정합 등과 같은 여러 영상처리 분야에 많이 사용된다[1-3]. 특징점 정합의 성능에서 중요한 부분은 올바른 정합과 잘못된 정합(mismatch)의 비율이다. 잘못된 정합에 비해 올바른 정합이 많을수록 높은 성능을 의미한다. 기존에 연구된 특징점 정합 기술들은 SIFT(Scale Invariant Feature Transform)[4], PCA-SIFT[5], GLOH[6], ASIFT[7] 등이 있다. 이와 같은 방법에서 특징점 정합에 사용되는 지역 이미지 기술자(local image descriptor)들은 특징점을 중심으로 고정된 범위내의 이웃 화소들을 사용하여 생성된다. 그러나 반복된 패턴이 포함된 이미지에서는 패턴의 유사성에 의해 고정된 범위 내의 기술자들을 사용할 경우 많은 잘못된 정합이 발생하게 된다. 이와 같은 잘못된 정합들은 기하학적인 방법으로 일부 제거할 수 있다. 기하학적인 정보를 사용한 잘못된 정합 제거 방법으로는 RANSAC 알고리즘과 호모그래피 또는 기초 행렬(fundamental matrix)을 사용하여 잘못된 정합을 제거한 연구[8, 9], 2개의 정합쌍을 이용한 스펙트럴 정합[10], 3개의 정합 관계를 이용한 텐서 기반 정합[11] 등이 있다. 그러나 이러한 방법들은 초기 정합들의 올바른 정합과 잘못된 정합의 비율에 매우 민감한 방법이므로 반복패턴에 의해 잘못된 정합이 많이 발생한 경우, 올바른 결과를 기대하기 어렵다.

본 연구의 목적은 동일 패턴이 반복적으로 나타나는 이미지에서 기존의 정합 방법들이 잘못된 정합들을 많이 생성하는 단점을 분석하여 그 원인을 찾고, 이를 해결하기 위한 방법으로 MDMF(Multiple Descriptor-based Mismatch Filtering) 방법을 제안한다. 반복패턴이 포함된 이미지에서는 특징점의 기술자가 다른 여러 특징점들과 유사하기 때문에 잘못된 정합들이 많이 생성된다. 예를 들어, 하나의 이미지에서 나타나는 특징점과 유사한 특징점이 다른 이미지에서 여러 개 보일 경우, 실제로는 잘못된 정합이라 하더라도 기술자의 유사성에 의해 정합될 수 있다. 이러한 점을 극복하기 위해 이웃한 특징점들의 기술자를 정합에 이용하여 잘못된 정합 수를 줄이는 알고리즘을 제안한다.

2장에서는 대표적인 특징점 정합 알고리즘인 SIFT의 기술자 생성과정을 설명한다. 3장에서는 반복패턴에 대한

SIFT 기술자의 단점과 분석에 대해 설명한다. 4장에서는 반복패턴에 강인한 기술자 생성과 정합방법에 대해 설명한다. 5장에서는 실험결과를 통해 다른 특징점 추출 알고리즘들의 성능과 제안한 방법의 성능을 비교 분석하여 제안한 방법의 우수성을 보인다. 마지막으로 제안한 알고리즘을 요약하고 앞으로의 연구 방향에 대해 기술하였다.

## II. SIFT 알고리즘

SIFT 알고리즘은 다음과 같이 4단계로 구현되어 있다 [4].

- 스케일 공간 극값 특징점 추출 : 모든 스케일의 DoG(가우시안의 차) 피라미드에서 극값을 갖는 특징점 후보들을 추출한다. 이 특징점 후보들은 크기 변화와 회전에 강한 요소이다.
- 특징점 위치 측정 : 추출한 특징점 후보 중에서 정합에 있어서 안정적이지 못한 코너에 가깝거나 이웃 픽셀들과 대비가 낮은 특징점들을 제거한 후, 각 특징점의 위치를 원본 이미지의 크기에 맞게 보간한다.
- 방향 설정 : 기울기 방향을 기반으로 각 특징점에서 하나의 방향성분을 구한다. 이 방향성분은 회전에 강한 요소이다.
- 특징점 기술자 : 각 특징점 위치에서 주변 픽셀 정보들을 통해 기울기 히스토그램을 생성하여 기술자를 생성한다.

SIFT의 특징점들은 가우시안 피라미드와 크기변환을 통해 크기 변화에 강한 점들로 추출된다. 추출된 특징점들은 특징점 정합을 위해 주변 픽셀들의 기울기 크기를 포함하는 기술자를 갖는다. 그림 1은 기술자의 생성과정을 보인 것이다.

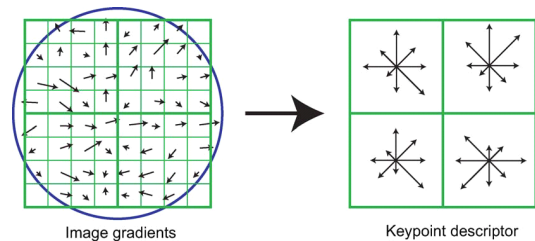


그림 1. SIFT의 기술자 생성 과정((4))  
Fig. 1. Process of descriptor creation in SIFT ((4))

먼저 특징점이 추출된 가우시안 레이어 이미지에서 특징점 주변 픽셀의 기울기 크기와 방향을 계산한다. 회전에 강인한

기술자를 생성하기 위해 특징점의 방향을 기준으로 상대적인 기술기 방향을 계산한다. 그림 1의 왼쪽 그림에서 화살표의 길이는 계산된 기술기의 크기를 의미하고 화살표의 방향은 기술기의 방향을 의미한다. 정합의 계산을 좀 더 단순화시키기 위해 기술기의 크기와 방향을 히스토그램을 사용하여 압축시킨다. 먼저 계산된 기술기에 특징점 위치를 중심으로 가우시안 원형 윈도우를 적용하여 기술기의 크기와 방향을 추출한다. 추출된 기술기의 크기와 방향으로 히스토그램을 만든다. 히스토그램의 빈은 하나의 기술기 방향을 나타내고 히스토그램의 크기는 그 방향의 기술기 크기의 합을 나타낸다. 히스토그램의 빈은 8개이다. 그림 1의 오른쪽 그림은 기술기의 크기와 방향을 히스토그램으로 압축하여 최종적인 기술자를 생성한 결과이다. 그림에서는 2x2의 기술자를 보였지만 SIFT의 실제 구현에서는 4x4의 기술자를 사용하였다.

### III. 반복패턴과 기술자

현재 연구된 기술자는 대부분 하나의 특징점을 기준으로 주변 픽셀들의 정보를 포함하지만 그 영역이 고정되어 있다. SIFT의 실험결과에 따르면 실험적으로 가장 안정된 영역 변경이 정해졌으나 반복패턴이 포함된 이미지 정합에서는 이러한 특성이 단점으로 작용한다. 반복패턴이 발생하는 부분에서 특징점이 추출될 경우 좁은 영역에서는 특징점의 주변 픽셀 정보가 유사하기 때문에 잘못된 정합을 생성할 수 있기 때문이다

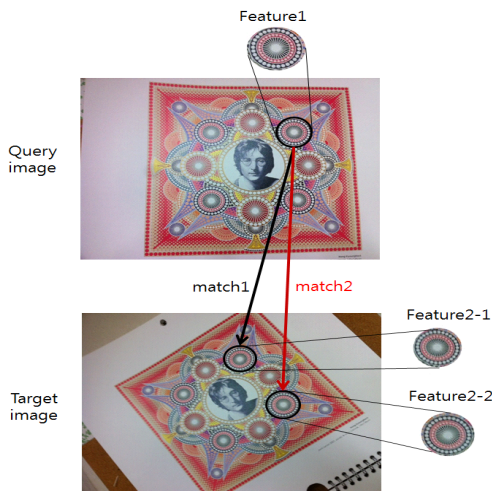


그림 2. 반복된 패턴에서 SIFT 기술자의 단점  
Fig. 2. Disadvantage of SIFT descriptor in repetitive pattern

다. 그림 2는 잘못된 정합이 발생할 수 있는 과정을 보인 것이다.

목적 이미지는 질의 이미지를 회전시켜 만들었다. 두 이미지에 나타나는 검은색 원들의 중심에서 특징점들이 추출되었을 때 검은색 원은 기술자의 추출(sampling) 범위를 나타낸다.

match1은 올바른 정합이고 match2는 잘못된 정합이다. 질의 이미지의 특징점 기술자와 목적 이미지의 두 특징점 기술자들은 매우 유사하다. 만약 match2가 match1보다 기술자 벡터의 유클리디언 거리가 더 가깝다면 match1이 제거될 것이다.

## IV. MDMF 알고리즘

### 1. 특징점 추출 및 다중 기술자 생성

특징점을 추출하기 위해서 SIFT[4], PCA-SIFT[5], GLOH[6], ASIFT[7], BRISK[12], ORB[13] 등 기존의 방법들을 사용할 수 있다. 정합 방법은 특징점 추출 방법에 따라 기술자 벡터의 유클리디언 거리 비교 방법과 비트패턴 비교 방법으로 나눌 수 있다. SIFT, PCA-SIFT, GLOH, ASIFT는 유클리디언 거리 비교 방법을 사용하고 BRISK와 ORB는 비트패턴 비교 방법을 사용한다. MDMF 알고리즘은 기존 특징점 추출 방법들 중 하나를 사용하여 특징점들을 추출하고 기술자들을 생성한 후 특징점의 종류에 맞는 기술자 비교 방법을 사용하여 정합한다.

기술자를 생성하기 위해 추출한 모든 특징점들 중 입력받은 정합에서 사용하고 있는 특징점들만 선택한다. 그림 3은 추출된 특징점들 중 0번 특징점의 다중 기술자를 생성하는 과정을 자세히 보인 것이다. k는 다중 기술자의 총 개수이며 k가 클수록 더 정확히 정합들을 제거하지만 올바른 정합들도 함께 제거될 수 있다. 본 논문의 실험에서는 k를 3이나 4로 설정하였다. 먼저 한 이미지 내에서 0번 특징점과 다른 특징점들 사이의 거리를 모두 계산하여 특징점 번호와 함께 거리 구조체 목록에 저장하고, 저장된 목록을 거리에 대한 오름차순으로 정렬한다. 정렬된 거리 구조체 목록 중 k번째까지의 특징점들에 대한 기술자를 기술자 벡터에서 읽어온다. 읽어온 기술자를 거리가 짧은 것부터 차례로 다중 기술자 배열에 저장한다. 이때 첫 번째 기술자는 0번 특징점 자신의 기술자이다. 첫 번째 기술자는 일반 정합에 사용되고 나머지 기술자는 잘못된 정합들을 제거하는데 사용된다.

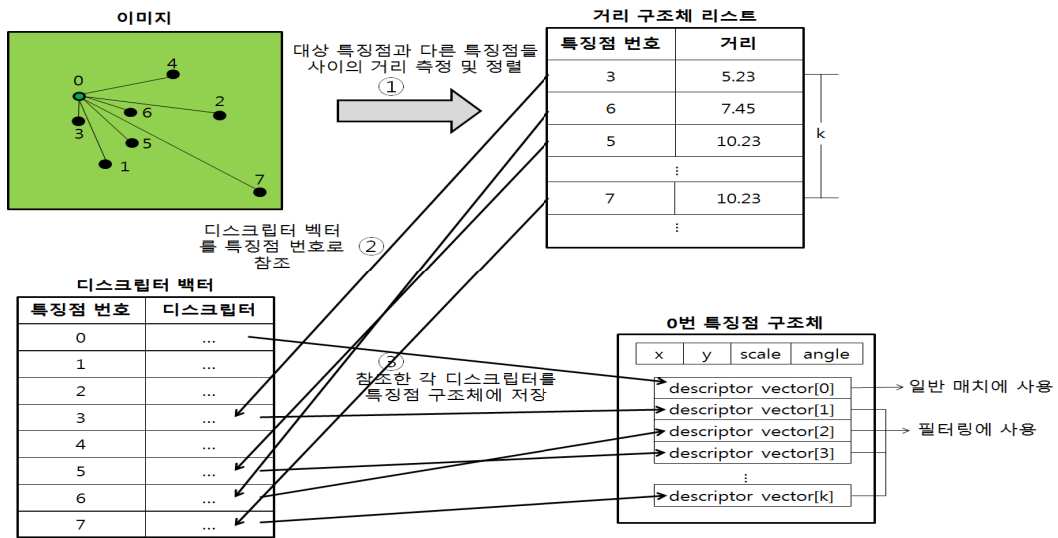
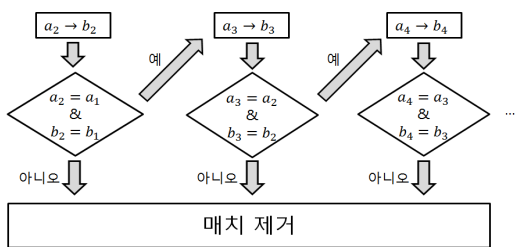


그림 3. MDMF의 다중 기술자 생성 과정  
Fig. 3. Process of multiple descriptors generation in MDMF

2. 정합과 잘못된 정합 제거

특징점을 정합할 때는 각 특징점의 첫 번째 기술자를 사용하여 계산한다. 정합 방법은 구현에 따라 바뀔 수도 있지만 본 논문의 실험에서는 SIFT에서 소개한 정합 방법을 사용하였다.

그림 4는 기술자들을 사용하여 잘못된 정합들을 제거하는 과정을 보인 것이다.



$a_k$ : k 번째 매치의 첫 번째 특징점.  
 $b_k$ : k 번째 매치의 두 번째 특징점.

그림 4. MDMF의 잘못된 정합 제거 과정  
Fig. 4. Process of filtering in MDMF

k번째 정합은 비교대상이 되는 두 특징점들의 k번째 기술자를 사용한다. 만약 k번째 기술자 정합의 결과가 k-1번째 기술자의 정합 결과와 다르다면 잘못된 정합으로 판단하고 제거한다. n-NN MDMF의 경우 이와 같은 과정을 n번째 기술자까지 반복한다.

V. 실험 결과 및 분석

1. 실험 방법

MDMF 프로그램은 Visual Studio 2010과 OpenCV2.4.4 오픈 라이브러리[15]를 사용해 구현하였다. 실험을 위해 구현한 프로그램은 SIFT 프로그램, MDMF 프로그램, 참값(ground-truth) 생성 프로그램, 비교 프로그램이 있다. SIFT 프로그램은 OpenCV에서 제공하는 SIFT 함수를 통해 SIFT를 수행하는 프로그램으로 정합 대상이 되는 두 이미지를 입력으로 받고 정합 정보와 정합된 이미지를 파일로 출력한다. MDMF 프로그램은 본 논문에서 제안한 알고리즘을 구현한 프로그램으로 SIFT에서 추출된 특징점 기술자 파일과 대상 이미지들을 입력으로 받고 MDMF의 결과를 정합정보 파일과 결과 이미지로 출력한다. 참값 생성 프로그램은 정합 결과와 대상 이미지들을 입력으로 받아서 사용자가 수동적으로 잘못된 정합과 올바른 정합을 구분해주면 결과를 파일로 저장시켜주는 프로그램이다. 비교 프로그램은 참값 생성 프로그램에서 출력된 올바른 정합 파일과 MDMF나 SIFT 프로그램에서 받아온 결과를 입력으로 받아서 정합수와 올바른 정합 수, 잘못된 정합 수를 출력하여준다.

## 2. 크기변화와 회전

실험에서는 기존 SIFT 알고리즘의 정합결과와 제안한 방법의 SIFT 정합 결과를 비교하였다. 정합결과를 평가하기 위한 지표로 올바른 정합 개수를 총 정합 개수로 나눈 값을 사용하며 이 값을 정합율이라 한다. 그림 5은 실험에 사용된 반복패턴이 있는 이미지 데이터들이다.

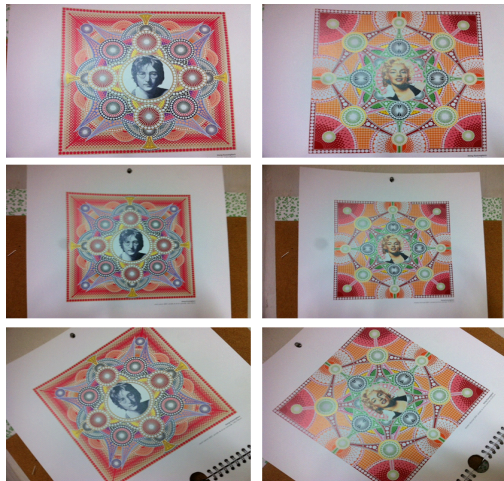


그림 5. 반복패턴이 있는 이미지들  
Fig 5. Repetitive pattern images

상단의 이미지가 원본이미지이고 가운데 단의 이미지는 조금 멀리서 촬영한 이미지이다. 하단의 이미지는 회전시켜 촬영한 이미지이다. 실험에서는 같은 대상 물체의 이미지들끼리 즉 왼쪽 이미지 또는 오른쪽 이미지끼리 서로 정합하였다.

그림 6은 두 이미지의 회전과 크기 변화에 대해 SIFT와 MDMF의 정합율을 그래프로 보인 것이다.

MDMF는 2-NN에서 5-NN까지 비교하였다. cal1은 그림5의 왼쪽 그림들이고 cal2는 그림5의 오른쪽 그림들이다. 점미알파벳인 r과 z는 각각 회전과 확대를 의미한다. MDMF로 제거한 결과 총 정합의 개수는 감소하지만 정합율은 증가하는 것을 볼 수 있다. 이것은 잘못된 정합의 수보다 올바른 정합의 수가 더 많아지는 것을 의미한다. 정합율과 총 정합 수의 증감을 비교해 보았을 때 cal1r과 cal2r, cal2z는 MDMF-2이 가장 적절하고 cal1z는 MDMF-4가 가장 적절하다. 정합시에 사용되는 이웃 기술자들의 수가 증가할수록 조금 더 정확한 정합들을 남기지만 올바른 정합들도 많이 제거된다.

그림 7은 cal1r에서 SIFT와 MDMF-2로 정합한 결과이다. 초록색 선은 올바른 정합들이고 빨간색 선은 잘못된 정합들이다.

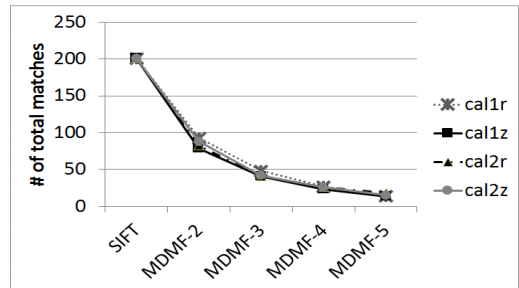
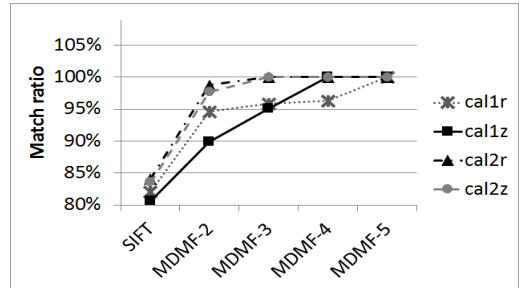


그림 6. 각 이미지별 정합율과 총 정합 수  
Fig. 6. Match ratio and the number of total matches for each image

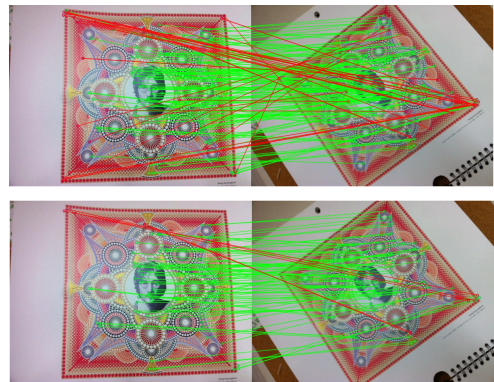


그림 7. cal1r 이미지에서 MDMF-2의 결과  
Fig. 7. Result of MDMF-2 in cal1r image

## 3. 원근 변환

SIFT는 왜곡이 심한 원근 변환에서는 성능이 좋지 않은 특징점 추출 방법이다. 따라서 원근 변환의 정도가 심한 경우에는 원근 변환에 강인한 ASIFT[7]를 사용하여 결과를 비교한다. 그림 8은 기존 이미지와 기존 이미지를 원근 변환한 이미지이다.

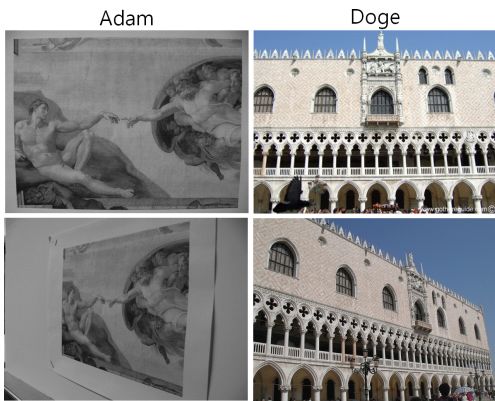


그림 8. 원본 이미지들과 원근 변환된 이미지들  
Fig. 8. Original images and perspective transformed images

왼쪽 이미지 쌍은 반복패턴이 없는 원근 변환 이미지이고 오른쪽 이미지 쌍은 반복패턴이 있는 이미지 쌍이다.

그림 9는 정합의 결과를 보인 것이다.

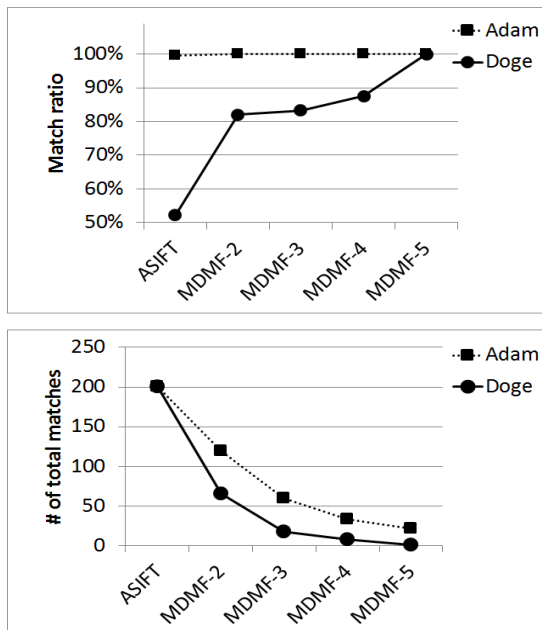


그림 9. 그림 8의 이미지들에 대한 ASIFT와 MDMF의 결과  
Fig. 9. The result of ASIFT and MDMF for the images of Fig. 8

Adam 이미지의 경우 ASIFT에서 사용된 입력 이미지들로 거의 모든 정합들이 올바르게 정합된다. k-NN을 선택할 때 계산되는 유클리디언 거리는 원근 변환에 민감하기 때문에

k-NN에 기반하여 구현된 MDMF도 원근 변환에 다소 민감하다. 따라서 결과에서 MDMF를 이용해 제거할 경우 총 정합 수가 감소하는 것을 볼 수 있다. Doge의 경우 ASIFT의 홈페이지[14]에서 소개하고 있는 ASIFT가 실패하는 경우 중 하나인 반복패턴이 있는 경우이다. 결과에서 볼 수 있듯이 MDMF-2에서 상당히 많은 잘못된 정합들이 제거된 것을 볼 수 있다. 총 정합 수와 정합율을 비교해볼 때 MDMF-2가 가장 적절하다.

그림 10은 Doge에서 ASIFT와 MDMF-2로 정합한 결과이다. 초록색 선은 올바른 정합들이고 빨간색 선은 잘못된 정합들이다.

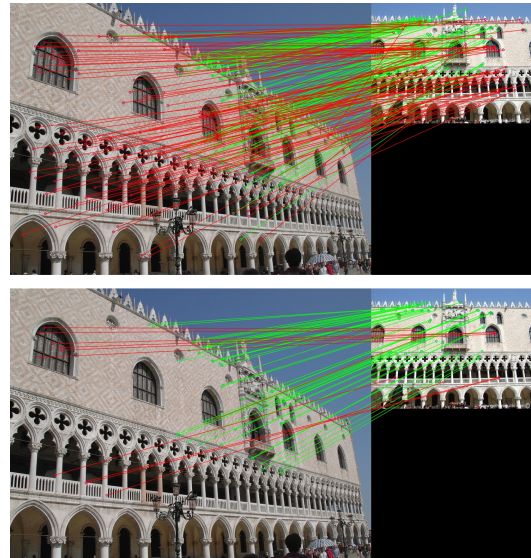


그림 10. Doge 이미지에서 ASIFT와 MDMF-2를 적용한 결과  
Fig. 10. The result of ASIFT and MDMF in Doge image

특징점 정합을 수행한 후 기하학적인 방법들을 후속 처리하여 잘못된 정합들을 제거할 수 있다. 하지만 잘못된 정합의 비율이 더 높다면 오히려 올바른 정합들이 더 많이 제거될 수 있다. 그림 9의 두 결과 모두 총 정합 수가 감소하지만 정합율이 증가한다. 따라서 ASIFT보다 MDMF에서 후속 처리하는 것이 더 나은 결과를 보일 것이다.

#### 4. 정합율에 따른 기하학적 처리의 효율

그림 11은 올바른 정합과 잘못된 정합의 비율에 따라 RANSAC 알고리즘과 펀더멘털 행렬을 통해 잘못된 정합을 제거한 결과를 정합율로 보인 것이다.

call과 callr을 대상으로 SIFT를 사용하여 100개의 올바른 정합을 추출한 후 그 중 일부를 잘못된 정합으로 바꿔

RANSAC 알고리즘을 수행한 것이다. x축은 잘못된 정합으로 바뀐 정합의 개수를 의미한다. 그림에서 확인할 수 있듯이 잘못된 정합이 많을 경우 잘못된 정합들이 제대로 제거되지 않는 것을 볼 수 있다. 이는 잘못된 정합들이 호모그래피 행렬 생성 시 잘못된 정보를 주기 때문이다.

## VI. 결론

본 논문에서는 반복패턴이 있는 이미지에 대해 많은 잘못된 정합이 발생하는 SIFT 기술자를 개선하는 방안인 MDMF를

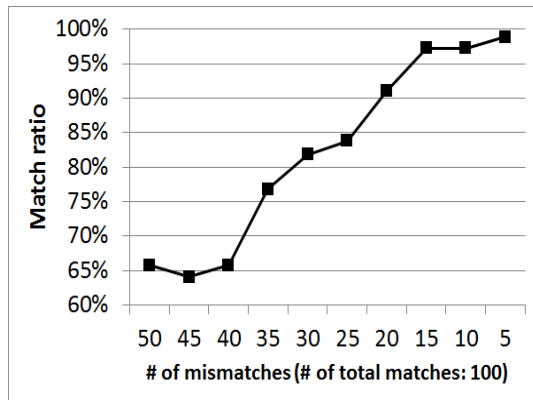


그림 11. 입력정합에서 잘못된 정합의 개수에 따른 RANSAC 알고리즘의 정합율

Fig. 11. Match ratio of RANSAC algorithm for the number of mismatches of input matches

제시하였다.

본 논문에서는 하나의 특징점에 대해 k-NN의 특징점 기술자들을 취하여 다중 기술자를 생성하고 이를 잘못된 정합 제거에 사용하여 반복패턴이 있는 이미지에 적용하였다. 실험에서는 반복패턴이 있는 이미지에 기존 SIFT와 ASIFT, MDMF를 각각 적용하여 정합율과 총 정합 수를 비교하였다. 실험 결과에서는 MDMF의 단계가 증가할수록 총 정합수가 감소하는 경향을 보였으나 정합율은 증가하는 것을 볼 수 있었다. 정합율과 총 정합 수의 증감을 비교하여 적절한 단계의 MDMF를 사용하여 좋은 결과를 얻을 수 있었다. 그러나 제안한 방법은 실행시간이 오래 걸리고 3차원 시점 변화에 대해 민감하다는 단점이 있다. 실행시간은 병렬처리와 같은 기법을 통해 개선할 수 있고 3차원 시점의 변화에 대해서는 k-NN 이외의 방법으로 보완할 수 있는 방안을 연구할 계획이다.

## 참고문헌

- [1] T. Tuytelaars, L. Van Gool, "Matching widely separated views based on affine invariant region," International Journal of Computer Vision, Vol.59, No. 1, pp. 56-75, 2004.
- [2] G. Zhang, Y. Wang, "Robust 3D face recognition based on resolution invariant features," Pattern Recognition Letters Vol. 32, No. 7, pp. 1009-1019, 2011.
- [3] J. Wu, J.M. Rehg, "CENTRIST: a visual descriptor for scene categorization," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, No. 8, pp. 1489-1501, 2011.
- [4] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," International Journal of Computer Vision, Vol. 60, No. 2, pp. 91-110, Nov. 2004.
- [5] Y. Ke, R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," IEEE International Conference on Computer Vision and Pattern Recognition, pp 506-513, 2004.
- [6] K. Mikolajczyk, C. Schmid, "A performance evaluation of local descriptors," IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 27, No. 10, pp. 1615-1630, 2005.
- [7] J. M. Morel, G. Yu, "ASIFT: a new framework for fully affine invariant image comparison," SIAM Journal on Imaging Sciences, Vol. 2, No. 2, pp. 438-469, 2009.
- [8] P. Pritchett, and A. Zisserman, "Wide baseline stereo matching," IEEE International Conference on Computer Vision, pp. 754-760, Jan. 1998.
- [9] W. Wei, H. Jun, T. Yiping, "Image Matching for Geomorphic Measurement Based on SIFT and RANSAC Methods," International Conference on Computer Science and Software Engineering, vol. 2, pp. 317-320, Dec. 2008.
- [10] M. Leodeanu, M. Hebert, "A Spectral Technique

for Correspondence Problems Using Pairwise Constraints,” IEEE International Conference on Computer Vision, vol. 2, pp. 1482-1489, Oct. 2005.

[11] O. Duchenne, F. Bach, K. Inso, J. Ponce, “A Tensor-Based Algorithm for High-Order Graph Matching,” IEEE International Conference on Computer Vision and Patter Recognition, pp. 1980-1987, June 2009.

[12] S. Leutenegger, M. Chli, R. Y. Siegwart, “BRISK : Binary Robust invariant scalable keypoints,” IEEE International Conference on Computer Vision, pp. 2548-2555, Nov. 2011.

[13] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” IEEE International Conference on Computer Vision, pp. 2564-2571, Nov. 2011.

[14] Affine-SIFT(ASIFT) web site, <http://www.cmap.polytechnique.fr/~yu/research/ASIFT/demo.html>

[15] OpenCV (Open Source Computer Vision), <http://opencv.org/>

## 저 자 소 개



**김 재 영**  
 2012: 울산대학교  
 컴퓨터정보통신공학부 공학사  
 현 재: 울산대학교 대학원  
 전기전자컴퓨터공학과 석사과정  
 관심분야: 디지털영상처리, GPU 컴퓨팅  
 Email : k jy7097@naver.com



**전 희 성**  
 1981년 : 서울대학교  
 전기공학과 공학사  
 1983년 : 서울대학교 대학원  
 전기공학과 공학석사  
 1992년 : 미국 Rutgers - The State  
 University of New Jersey  
 컴퓨터공학과 공학박사  
 현 재: 울산대학교 전기공학부 교수  
 관심분야 : 디지털영상처리,  
 GPU 컴퓨팅, 컴퓨터비전,  
 증강현실  
 Email : hsjun@ulsan.ac.kr