

부분 해밍 거리의 순차적 분석을 통한 이진 특징 기술자의 고속 정합에 관한 연구

A Study on Fast Matching of Binary Feature Descriptors through Sequential Analysis of Partial Hamming Distances

박한훈*, 문광석**

Hanhoon Park*, Kwang-Seok Moon**

요약

최근, 이진 특징 기술자를 생성하는 방법에 대한 연구가 많이 진행되고 있다. 이진 특징 기술자의 정합은 비트 연산에 기반한 해밍 거리를 이용하므로 실수 연산에 기반한 유클리디안 거리를 이용하는 기존의 일반적인 특징 기술자의 정합보다 훨씬 효율적이기 때문이다. 그러나, 특징 수의 증가는 정합 속도를 선형적으로 감소시키는 원인이 되기 때문에, 객체 추적과 같은 실시간 처리가 중요한 응용 분야에서는 이진 특징 기술자의 정합 속도를 더욱 향상시킬 수 있는 방법에 대한 요구가 증가해 왔다. 이에 본 논문에서는 고차원의 이진 특징 기술자를 여러 저차원의 이진 특징 기술자로 나누어 부분 해밍거리를 계산하고 순차적으로 분석함으로써, 정합 속도는 크게 개선하면서도 정확도는 유지할 수 있는 방법을 제안한다. 제안된 방법의 효율성을 분석하기 위해 기존의 정합 방법들과의 비교 실험을 수행한다. 아울러, 제안된 고속화 방법의 성능을 극대화하기 위한 이진 특징 기술자 생성 방법에 대해서도 논의한다. 몇 가지 생성 방법에 대한 성능을 분석함으로써, 가장 효과적인 방법을 모색한다.

ABSTRACT

Recently, researches for methods of generating binary feature descriptors have been actively done. Since matching of binary feature descriptors uses Hamming distance which is based on bit operations, it is much more efficient than that of previous general feature descriptors which uses Euclidean distance based on real number operations. However, since increase in the number of features linearly drops matching speed, in applications such as object tracking where real-time applicability is a must, there has been an increasing demand for methods of further improving the matching speed of binary feature descriptors. In this regard, this paper proposes a method that improves the matching speed greatly while maintaining the matching accuracy by splitting high dimensional binary feature descriptors to several low dimensional ones and sequentially analyzing their partial Hamming distances. To evaluate the efficiency of the proposed method, experiments of comparison with previous matching methods are conducted. In addition, this paper discusses schemes of generating binary feature descriptors for maximizing the performance of the proposed method. Based on the analysis on the performance of several generation schemes, we try to find out the most effective scheme.

Keywords : Binary feature descriptor, partial Hamming distance, fast feature matching, speed-oriented binary feature descriptor generation

I. 서론

영상에서 특징이라 함은 코너(corner)나 블롭(blob) 같이 밝기, 색상, 모양 등이 주변과 크게 다른 점이나 영역을 의

미한다. 그러므로, 특징은 영상 처리, 이해(understanding) 분야에서 매우 중요한 요소이다. 예를 들어, 물체를 인식하기 위해서는 물체의 특징을 파악하고 영상으로부터 관련된 특징을 찾아야 한다. 특징 사이의 비교 혹은 구별은 각 특징을 표현하는 특징 기술자(feature descriptor)를 이용하게 되는 데, 이는 영상에서 검출된 각 특징 주변의 지역적인 텍스처(texture) 정보를 표현하는 벡터를 의미한다. 특징 기술자를 생성하는 대표적인 방법으로 SIFT[1]와 SURF[2]가 있다. SIFT와 SURF는 영상의 회전, 이동, 조명 변화에 매우 강건하게 특징을 검출, 기술할 수 있으므로, 영상 인식, 객체 추적, 3차원 재구성 등 다양한 분야에 활용되어 왔다. 하지만,

* 부경대학교

** 부경대학교 (교신저자)

투고 일자 : 2013. 7. 25 수정완료일자 : 2013. 10. 22

게재확정일자 : 2013. 10. 31

※ 본 논문은 2013년도 부경대학교 자율창의학술연구비 지원에 의하여 수행되었음(과제번호: CD20130533).

이와 같은 기존의 대부분의 특징 기술자는 실수 값을 사용하기 때문에, 이와 관련된 연산(예, 특징 기술자 사이의 비교를 위한 거리 계산)은 많은 연산 시간과 메모리 공간을 필요로 한다. 따라서 보다 작은 자료형으로 구성된 특징 기술자에 대한 연구가 활발히 진행되어왔으며, 최근에는 비트열로 구성된 이진 특징 기술자를 생성하는 방법인 BRIEF[3], BRISK[4], ORB[5], FREAK[6] 등이 제안되었다. BRIEF는 특징 주변의 일정한 크기를 가지는 영역에서 주어진 특징 기술자의 차원 수만큼 반복해서 임의로 선택된 두 샘플 픽셀의 밝기 값을 비교하여 그 결과를 이진 값(0 또는 1)으로 나열한 비트열로 특징을 기술한다. 하지만, BRIEF는 영상의 기하 변환에 불변성을 갖지 않기 때문에, ORB, BRISK, FREAK는 특징의 주방향 계산, 스케일 공간(scale space) 생성 등을 통해 회전 및 크기에 대한 불변성을 제공하고 샘플링 패턴을 개선하여 특징 기술자의 성능을 개선한 방법이다. 이러한 이진 특징 기술자는 실수 벡터인 기존의 특징 기술자에 비해 연산 속도도 빠르고 매우 작은 메모리 공간을 차지한다. 또한, 특징 기술자의 정합을 위해 해밍 거리(Hamming distance)를 이용하는데, 이는 XOR 비트연산으로 빠르게 수행할 수 있으며 SIMD(Single Instruction Multiple Data) 명령어를 사용할 경우 한 번의 명령어로 128개의 요소에 대한 XOR 연산이 가능하므로 더욱 빠르게 정합을 수행할 수 있다[7]. 하지만, 이러한 이진 특징 기술자의 장점에도 불구하고 특징의 개수가 많아지면 이진 특징 기술자의 정합 속도는 특징의 수에 비례적으로 감소하기 때문에, 객체 추적과 같은 실시간 처리가 중요한 응용 분야에 적용하기 힘들다. 그러므로, 최근 이진 특징 기술자의 정합 속도를 더욱 향상시킬 수 있는 방법에 대한 요구가 크게 증가해 왔다.

이진 특징 기술자의 고속 정합을 위한 대표적인 방법은 k -means와 같은 군집화 기법을 활용한 방법으로[7,8], 군집화 기법 및 군집의 개수를 적절히 선택함으로써 정합 속도를 크게 개선할 수 있다. 그러나, 이러한 방법은 군집화를 위한 추가적인 연산 시간이 필요하기 때문에, 군집화 수행 빈도가 증가할 경우 비효율적일 수 있다. 이러한 문제를 해결하기 위해서 본 논문에서는 고차원의 이진 특징 기술자를 여러 저차원의 이진 특징 기술자로 나누어 순차적으로 부분 해밍 거리를 분석함으로써, 추가적인 연산을 거의 필요로 하지 않고 정합 속도는 크게 개선하면서도 정확도는 유지할 수 있는 방법을 제안한다. 이는 특징 기술자의 차원 수가 아주 작지 않다면, 대부분의 특징 기술자의 정합 정확도는 크게 다르지 않다는 사실에 기반한 것으로, 고차원 특징 기술자의 경우 드문(sparse) 거리 분포로 인해 안정성이 떨어질 수도 있기 때문에[9] 저차원 특징 기술자를 이용하는 것이 오히려 효과적일 수도 있다. 물론, 제안된 방법에서는 고차원 특징 기술자의 정보를 그대로 유지하기 때문에 정확도 향상을 기대할 수는 없을 것이다.

제안된 고속화 방법의 성능을 극대화하기 위해 기존의 이진 특징 기술자 생성 방법을 개선하는 방안에도 논의를 한다. 즉, 제안된 방법은 순차적으로 부분 해밍 거리를 계산하기 때문에, 특징 기술자의 앞쪽으로 보다 분별력이 높은 기

술자를 배치하는 것이 효과적일 것이다. 본 논문에서는 몇 가지 이진 특징 기술자를 생성하는 방법을 마련하고, 각각의 성능을 분석함으로써, 가장 효과적인 방법을 모색한다.

기존의 군집화 기반 고속화 방법들은 정합을 위한 탐색 공간을 줄이는(즉, 정합의 수를 줄이는) 방법이기 때문에, 특징의 수가 많은 경우에 효과적이지만, 제안된 방법은 각 특징 정합에 걸리는 시간을 줄이는(XOR 연산의 수를 줄이는) 방법이기 때문에, 특징의 수에 상관없이 일정한 효과를 가진다.

본 논문은 다음과 같이 구성된다. II 장에서는 기존의 이진 특징 기술자를 정합하는 방법에 대해 간단히 설명한다. III 장에서는 이진 특징 기술자의 정합을 고속화하기 위한 제안 방법을 설명하고, IV 장에서는 실험을 통해 제안 방법의 성능을 검증한다. 마지막으로 V 장에서 결론을 제시한다.

II. 기존의 이진 특징 기술자 정합

이진 특징 기술자는 이진 값을 가지는 비트열로 표현되기 때문에, 일반적으로 이진 특징 기술자 사이의 정합은 해밍 거리를 이용하여 고속으로 수행된다. 즉, 서로 다른 D 차원의 이진 특징 기술자($vector_A$, $vector_B$)가 주어지면, 각 비트 위치에서의 XOR 연산을 D 번 수행한 후, 결과에 나타나는 1의 개수가 두 이진 특징 기술자의 해밍 거리를 나타내며, 해밍 거리의 역수가 정합 정도를 표현한다. 결과적으로, 두 D 차원의 이진 특징 기술자를 정합하는 과정은 D 번의 XOR 연산과 한 번의 정수 카운팅 연산으로 이루어진다(의사코드 1 참조). 물론, 효과적인 정합을 위해서 기존의 실수 특징 기술자와 마찬가지로 NNDR[1]을 이용할 수 있다. 결과적으로, 한 쌍의 이진 특징 기술자를 정합하는 데 걸리는 시간 T_m 은 아주 작다.

정합에 사용되는 특징의 수가 늘어나면, 연산의 수는 선형적으로 증가한다. 즉, N 개의 특징과 N 개의 특징을 정합하기 위해서는 $N*N*T_m$ 만큼의 시간이 걸리므로, T_m 이 작아도 아주 큰 N 에 대해 전체적인 정합 시간은 여전히 크다. 이러한 문제를 해결하기 위해 N 의 크기를 줄이려고 한 것이 군집화를 이용한 방법이다[7,8]. 즉, 미리 N 개의 특징을 k 개의 군집으로 나누고 각 군집의 대표 기술자를 결정해 두면(이때 걸리는 시간을 T_c 라고 하자), 주어진 한 특징과 정합하는 특징을 찾는 과정은 N 개의 특징과의 해밍 거리를 다 계산하지 않고, k 개 군집의 대표 기술자와의 해밍 거리만을 계산한 후 가장 작은 해밍 거리를 가지는 군집 내의 N/k 개 특징과의 해밍 거리를 계산한다. 결과적으로, 군집화 방법에 따라 차이가 날 수도 있으나 평균적으로 $(N/k + k)$ 번의 해밍 거리 계산 과정이 수행된다. 하나의 특징을 정합하는 시간은 $\{(N/k + k)*T_m + T_c\}$ 가 되고, 전체 정합 시간은 $[(N/k)*(N/k) + k*k]*T_m + 2T_c\}$ 가 된다.

의사코드 1: 일반적인 이진 특징 기술자 정합

```

Hamming_distance = 0;
FOR i = 1 TO D
    Hamming_distance += XOR(vector_A[i], vector_B[i]);
END
    
```

III. 제안 방법

3.1. 순차적 부분 비교를 이용한 이진 특징 기술자 정합

앞 장에서 설명한 대로, 두 D 차원의 이진 특징 기술자를 정합하기 위해서는 항상 D 번의 XOR 연산이 필요하다. 반면, 이진 특징 기술자를 저차원의 부 기술자 단위로 나누고(sub_vector_A , sub_vector_B), 각 부 기술자를 따로 정합하면 XOR 연산의 수를 크게 줄일 수 있다. 예를 들어, 부 기술자의 차원을 seg 라고 하면, D 차원의 이진 특징 기술자는 $S=D/seg$ 개의 부 기술자로 나누어진다. 첫 번째 부 기술자부터 부분 해밍 거리(seg 번의 XOR 연산)를 계산한 후, 부분 해밍 거리가 일정한 값(Th)보다 작으면 다음 부 기술자의 부분 해밍 거리를 계산하고, 부분 해밍 거리가 일정한 값(Th)보다 크면 두 이진 특징 기술자는 서로 유사하지 않다고 판단하고 이후의 부 기술자에 대한 부분 해밍 거리는 더 이상 계산하지 않는다(의사코드 2 참조). 결국, 서로 유사한 이진 특징 기술자의 경우에는 마지막 부 기술자까지 부분 해밍 거리 계산이 수행되지만(이때 전체 XOR 연산의 수는 D 와 같다), 서로 유사하지 않은 이진 특징 기술자의 경우에는 일부 부 기술자 간의 부분 해밍 거리 계산만을 수행하기 때문에 전체 XOR 연산의 수는 D 보다 훨씬 작아진다. 그러므로, 기존의 군집화를 이용하는 방법과 달리 각 특징을 정합하기 위한 시간 T_m 을 줄이는 효과가 있다.

의사코드 2: 순차적 부분 비교를 통한 이진 기술자 정합

```

Hamming_distance = 0;
FOR n = 1 TO S
    partial_Hamming_distance = 0;
    FOR i = 1 TO seg
        partial_Hamming_distance
            += XOR(sub_vector_A[i], sub_vector_B[i]);
    END
    IF (partial_Hamming_distance > Th)
        Hamming_distance = D;
        BREAK;
    END
    Hamming_distance += partial_Hamming_distance;
END
    
```

3.2. 순차적 부분 비교의 효율성을 극대화하기 위한 이진 특징 기술자 생성

이진 특징 기술자를 생성하는 과정을 변형함으로써 앞 절에서 제안된 고속화 방법의 성능을 극대화할 수 있다. 즉, 제안된 고속화 방법은 순차적으로 부분 해밍 거리를 계산하기 때문에, 특징 기술자의 앞쪽으로 보다 분별력이 높은 부 기술자를 배치하는 것이 효과적일 것이다. 앞쪽에 분별력이 높은 부 기술자가 있으면 서로 유사성이 없는 특징 기술자 사이의 정합은 아주 적은 수의 부 기술자 사이의 부분 해밍 거리 계산만으로 종료된다. 여기서 문제는 분별력이 높은 부 기술자는 어떻게 정의될 수 있는가 하는 것이다. 기존 연구[6]에서 특징을 기술하기 위한 샘플링 패턴

은 특징에 가까운 샘플의 밀도를 크게 하는 것이 보다 효과적이라는 것을 알 수 있다. 이는 특징에 가까운 샘플을 포함하는 부 기술자의 분별력이 더 높을 수 있다는 것을 나타낸다. 본 논문에서는 이와 관련하여 다음과 같은 네 가지의 이진 특징 기술자를 생성하는 방법을 마련하고, 각각의 성능 분석을 통해 가장 효과적인 방법을 모색한다.

- ① 두 샘플의 위치가 특징에 가까울수록 앞쪽으로 배치한다. 두 샘플 사이의 거리는 고려되지 않는다.
- ② 두 샘플의 위치가 특징으로부터 멀수록 앞쪽에 배치한다. 두 샘플 사이의 거리는 고려되지 않는다.
- ③ 두 샘플 중 하나는 특징에 가깝게 위치하고 다른 하나는 특징으로부터 멀 경우 앞쪽으로 배치한다.
- ④ 샘플의 위치에 상관없이 두 샘플의 거리가 클 경우 앞쪽으로 배치한다.

이진 특징기술자를 생성하는 기본적인 방법은 특징 주변의 일정한 영역에서 임의의 혹은 정해진 패턴으로 기술자의 차원 수만큼의 샘플 쌍을 결정한 후, 샘플 쌍의 밝기 비교를 통해 비트열을 저장한다. 이때, 각 샘플 쌍은 서로 독립적이기 때문에 저장되는 순서가 바뀌더라도 상관없다. 그러므로, 제안된 생성 방법에 의한 정합의 정확도 저하는 전혀 없다.

IV. 실험 결과

제안된 방법의 성능을 검증하기 위해 주어진 영상에 대해 정합에 걸리는 시간과 정합의 정확도를 구하는 실험을 수행하였다. 실험을 위해 [7]에서 사용된 참조 영상 및 참조 영상을 다양한 시점에서 촬영한 비디오 시퀀스를 사용하였다(그림 1 참조). 참조 영상은 640×512 의 해상도를 가지고, 비디오 시퀀스는 640×480 의 해상도를 갖는 총 708프레임의 영상으로 구성되었다. 실험은 i5-2467M 1.6 GHz CPU와 4 GB RAM을 가진 노트북 PC에서 수행되었다.



그림 1. 실험에 사용된 참조 영상(좌)과 비디오 시퀀스의 일부(우) [7].

Fig. 1. Left: A reference image used in experiments, right: its video sequence images [7].

실험에서 사용된 처리 과정 역시 [7]과 유사하다. 특징 검출은 FAST[10]를 사용하였고, 특징 기술은 BRIEF를 변형하여 사용하였다. BRIEF는 회전 및 스케일 불변성이 없기 때문에, 특징을 중심으로 반경 3픽셀 이내에서 계산한 평균 그래디언트 벡터의 방향을 특징의 주방향으로 사용하고 SIFT에서 사용하는 것과 유사한 스케일 공간을 사용함으로써, 회전 및 스케일 불변성을 가지도록

하였다. 실험에 사용된 참조 영상에서는 3258개, 비디오 시퀀스의 각 프레임 영상에서는 평균 200개의 특징이 검출, 기술되었다.

제안된 방법의 성능을 검증하기 위해 기존의 고속화 기법을 사용하지 않는 일반적인 정합 방법 및 군집화를 사용하는 방법[7]의 결과와 비교하였다. 군집화를 사용하는 방법의 경우 군집의 수에 따라 성능이 달라지며, 제안된 방법의 경우, 부 기술자의 차원 (*seg*)과 *Th* 값에 따라 성능이 달라지기 때문에 여러 값에 대한 결과를 도출하였다. 그림 2와 표 1은 각 방법에 의한 정합에 걸린 시간 및 정합의 정확도를 보여준다. 여기서 정확도는 고속화 기법을 사용하지 않는 일반적인 정합 방법에 의해 정합된 특징과 각 고속화 방법에 의해 정합된 특징을 비교하여, 서로 같은 특징을 찾은 후, 그 수를 일반적인 정합 방법에 의해 정합된 특징의 수로 나눈 값의 백분율을 의미한다. 제안 방법의 경우 부 기술자의 차원이 작고 *Th* 값이 작으면 정합 속도가 크게 증가하지만, 정확도가 크게 떨어질 수 있다. 특히, 부 기술자의 차원이 너무 작으면 (*seg*=8의 경우) *Th* 값에 상관없이 정합이 제대로 이루어질 수 없었다. 군집화를 이용한 방법은 군집의 수가 증가할수록 정합 속도가 크게 개선되었으나, 정확도 저하는 크지 않았다. 비슷한 정합 속도에 대해 군집화를 이용한 방법의 정확도가 더 높았다. 예를 들어, 40ms의 정합 속도를 가질 경우 제안 방법의 정확도는 80% 정도였지만, 군집화를 이용한 방법은 90% 정도였다. 그러므로, 정합 속도 면에서는 군집의 수가 클 경우 군집화를 이용하는 방법이 보다 효과적일 수 있다. 그러나, 이는 군집화를 참조 영상에 대해서만 수행한 결과로 군집화 과정의 횟수가 늘어날 경우 정합 속도는 급격히 감소할 것이다(예를 들어, 참조 영상을 사용하지 못하고 이전 프레임과 현재 프레임 사이의 특징 정합이 필요한 상황이라면 매 프레임마다 군집화가 이루어져야 한다). 정합의 정확도 면에서 주목할 점은 제안 방법의 경우 부 기술자의 차원과 *Th* 값이 어느 정도 크다면 손실이 거의 없다는 것이다. 결과적으로, 제안 방법을 사용하면(예, *seg*=32, *Th*=16) 실험 환경에 상관없이 정확도 손실없이 2 배 정도의 속도 향상을 기대할 수 있었다.

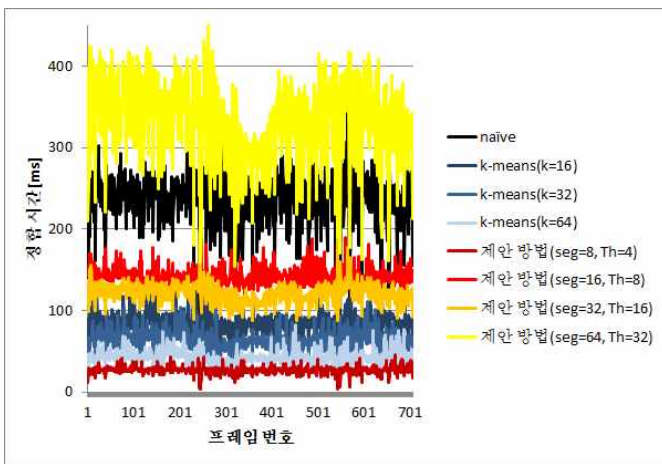


그림 2. 각 정합 방법에 따른 처리 시간.

Fig. 2. Processing time of each matching method.

표 1. 각 정합 방법에 따른 평균 처리 시간 및 정확도.

Table 1. Average processing time and accuracy of each matching method.

정합 방법	평균 처리 시간 [ms]	정확도 [%]	
일반적인 방법	236.52	x	
k-means	k=16	85.80	98.6
	k=32	64.38	98.6
	k=64	42.61	92.3
제안 방법	seg=8, Th=4	26.80	0.0
	seg=8, Th=2	12.81	0.0
	seg=16, Th=8	142.86	99.3
	seg=16, Th=5	40.90	77.0
	seg=16, Th=4	22.66	48.7
	seg=32, Th=16	122.04	100.0
	seg=32, Th=10	91.37	94.8
	seg=32, Th=8	49.78	80.7
	seg=64, Th=32	331.78	100.0
	seg=64, Th=21	419.89	99.8
seg=64, Th=16	225.58	95.1	

실험 결과 분석을 통해 기존의 군집화 기반의 고속화 방법과는 다른 제안 방법의 특성을 파악할 수 있다. 군집화 방법의 경우, 군집의 수(*k*)는 정합 속도에 비례하고 정합의 정확도에 반비례하는 특성을 보였다. 그러나, 제안된 방법의 경우 부 기술자의 차원이 너무 크면(*seg* = 64 이상), 순차적 부분 비교를 통해 얻어지는 속도 향상보다 순차적 부분 비교에 필요한 부가 처리로 인한 속도 저하의 효과가 더 커져 오히려 고속화 기법을 사용하지 않은 일반적인 정합 방법보다 비효율적이었다. 반대로, 부 기술자의 차원이 너무 작으면(*seg* = 16 이하) 순차적 부분 비교에 얻어지는 속도 향상에 비해 정합의 정확도 저하 효과가 크기 때문에 효과적이지 못했다. 결과적으로, 부 기술자의 차원 및 *Th* 값을 적절히 설정해야 한다.

제안 방법의 성능을 향상시키기 위해 III.2절에서 제시된 네 가지 샘플링 방법을 이용하여 특정 기술자를 생성한 후, 정합 속도 및 정확도를 측정하였다. 표 2에서 보는 것처럼, 샘플링 방법에 의해 정합 속도는 달라졌으며, 예상과는 달리 샘플의 위치와 상관없이 서로 거리가 먼 샘플 쌍을 앞쪽으로 배치하는 것, 즉 ④가 가장 효율적이었다. [6]에서 특징에 가까운 샘플의 밀도를 크게 하는 것은 ①이나 ③의 방법에 해당되는데, 두 방법의 효과는 크지 않았다. 특히, ①은 개선 효과가 없었으며 오히려 특징에서 멀리 떨어진 샘플 쌍을 앞쪽으로 배치하는 ②의 방법이 보다 효과적이었다. 결과적으로, 제안 방법의 성능을 향상시키기 위해서는(즉, 분별력이 높은 부 기술자를 앞쪽으로 배치하기 위해서는) 샘플 쌍의 위치보다는 샘플 쌍 사이의 거리가 중요한 의미를 가지며 거리가 먼 샘플 쌍을 앞쪽으로 배치해야 한다. 이와 같이 기술자 생성 방법을 개선함으로써 제안 방법의 정합 속도는 최대 약 13% 향상되었다. 정합 정확도는 샘플링 방법에 의해 거의 변하지 않았다. 이는 앞서 설명한대로 제안 방법은 *seg*와 *Th* 값이 일정 값 이상이 되면 정확도 저하가 거의 없으며, 이는 샘플링 방법에 의해서도 영향을 받지 않기 때문이다.

표 2. 샘플링 방법에 따른 제안 방법의 평균 처리 시간 및 정확도.
Table 2. Average processing time and accuracy of the proposed method according to each sampling method.

	샘플링 방법	평균 처리 시간 [ms]	정확도 [%]
seg=16, Th=8	변형 없음	142.86	99.3
	①	133.41	99.8
	②	120.48	100.0
	③	111.19	99.8
	④	107.98	100.0
seg=32, Th=16	변형 없음	122.04	100.0
	①	135.19	100.0
	②	117.31	100.0
	③	122.86	100.0
	④	113.74	100.0
seg=64, Th=32	변형 없음	331.78	100.0
	①	332.26	100.0
	②	325.66	100.0
	③	323.07	100.0
	④	305.23	100.0

V. 결론

본 논문에서는 고차원의 이진 특징 기술자를 저차원의 여러 부분 기술자로 나눈 후, 순차적으로 부분 해밍거리를 계산, 비교함으로써 이진 특징 기술자의 정합 속도를 크게 개선하면서도 정확도는 유지할 수 있는 방법을 제안하였다. 실험을 통해 제안된 방법은 고속화 기법을 사용하지 않은 일반적인 정합 방법에 비해 처리 속도를 크게 개선되었으며, 기존의 *k*-means 군집화 등을 이용한 고속화 방법에도 경쟁력 있는 성능을 가짐을 확인하였다. 또한, 제안된 고속화 방법의 속도를 더욱 향상시키기 위해 특징 기술자를 생성하는 방법의 개선에 대한 실험 및 논의를 통해 효율적인 기술자 생성 방법을 제안하였다.

특징 기술자를 생성하는 방법의 경우, 비교적 간단한 방법을 통해 본 논문에서는 최대 13% 정도의 개선 효과를 확인했으나, 향후 보다 큰 개선 효과를 기대할 수 있는 정교한 생성 방법에 대한 연구를 진행할 계획이다. 또한, 제안된 방법은 이진 특징 기술자가 아닌 실수 값을 가지는 일반적인 특징 기술자에도 적용이 가능하기 때문에, 향후 이에 대한 성능 분석을 수행할 계획이다.

참고 문헌

[1] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91-110, 2004.
 [2] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "SURF: Speeded-up robust features," *CVIU*, vol. 110, no. 3, pp. 346-359, 2008.
 [3] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," *Proc. of ECCV*, 2010.
 [4] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," *Proc. of ICCV*, pp. 2548-2555, 2011.

[5] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *Proc. of ICCV*, pp. 2564-2571, 2011.
 [6] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint," *Proc. of CVPR*, pp. 510-517, 2012.
 [7] 박정식, 박종일, "이진 특징 기술자의 군집화를 이용한 특징점 고속 정합," 한국방송공학회 추계학술대회 논문집, pp. 9-10, 2012.
 [8] M. Muja and D. Lowe, "Fast matching of binary features," *Proc. of CRV*, pp. 404-410, 2012.
 [9] M. Stommel, "Binarising SIFT-descriptors to reduce the curse of dimensionality in histogram-based object recognition" *IJSIP*, vol. 3, no. 1, pp. 25-36, 2010.
 [10] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," *Proc. of ECCV*, 2006.



박 한 훈(Han-hoon Park)

正會員

2000년 2월 한양대학교 전자통신전공학과 (공학사)
 2002년 2월 한양대학교 전자통신전공학과 (공학석사)
 2007년 8월 한양대학교 전자통신전공학과 (공학박사)

2007년 9월 ~ 2008년 10월 한양대학교 BK21박사후연구원
 2008년 11월 ~ 2011년 10월 NHK방송기술연구소 박사후연구원
 2011년 11월 ~ 2012년 2월 한양대학교 전기정보통신기술연구소 연구교수
 2012년 3월 ~ 현재 부경대학교 전자공학과 조교수
 ※주관심분야 : 3차원 영상처리, 증강현실, 인간컴퓨터 상호작용



문 광 석(Kwang-Seok Moon)

正會員

1979년 2월 경북대학교 전자공학과 (공학사)
 1981년 2월 경북대학교 전자공학과 (공학석사)
 1989년 2월 경북대학교 전자공학과 (공학박사)
 1988년 일본 동경대학교 학부연구원

1997년 8월 ~ 1998년 7월 미국 Jackson State University 객원교수
 1990년 3월 ~ 현재 부경대학교 전자공학과 교수
 ※주관심분야 : 영상신호처리, 적응신호처리 등