

마커리스 트래킹을 위한 특징 서술자의 데이터베이스 생성 및 검색방법*

윤요섭^{*}, 김태영^{**}
서경대학교 컴퓨터공학과
yysddt@hanmail.net, tykim@skuniv.ac.kr

A Database Creation and Retrieval Method of Feature Descriptors
for Markerless Tracking

Yo-Seop Yun^{*}, Tae-Young Kim^{**}
Dept. of Computer Engineering, Seokyeong University

요 약

본 논문에서는 증강 현실 환경에서 실시간 마커리스 트래킹을 수행하기 위한 특징 서술자 데이터베이스 생성 및 검색 방법을 제안한다. 먼저, 특징 서술자를 효율적으로 검색하기 위하여 특징 서술자의 형태를 기준으로 정수 부호화 하여 총 4 단계의 인덱스 데이터베이스를 구성한다. 특징 서술자의 검색은 데이터베이스에서 각 단계별로 유사성 있는 후보 특징 서술자의 인덱스를 탐색하고 입력된 특징 서술자와 탐색된 모든 후보 특징 서술자들의 유클리드 거리 값 비교를 통해 이루어진다. 본 연구에서 제안한 검색방법은 형태를 기반으로 유사하지 않은 특징 서술자들을 검색 대상에서 제외하여 검색의 효율을 높였다. 제안된 방법은 기존 KD-Tree 방법에 비해서 특징 서술자당 약 16ms의 검색 속도 개선이 있었음을 확인할 수 있었다.

ABSTRACT

In this paper, we propose a novel database creation and retrieval method of feature descriptors to support real-time marker-less tracking in the augmented reality environments. Each feature descriptor is encoded by integer and multi-level database is created in order to retrieve a feature descriptor efficiently. The retrieval of a feature descriptor is performed as follows: Firstly, candidate feature descriptors are searched by traversing the multi-level database. Secondly, the euclidean distance between input feature descriptor and each candidate one is compared. The shortest one is retrieved. The proposed method is 16 ms faster than previous KD-Tree method for each feature descriptor.

Keywords : Augmented Reality, Feature Descriptor, Database Retrieval

접수일자 : 2011년 01월 31일 일차수정 : 2011년 03월 16일 이차수정 : 2011년 04월 12일 심사완료 : 2011년 04월 15일
교신저자(Corresponding Author) : 김태영

* 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2010-0017206)

** 본 연구는 문화체육관광부 및 한국콘텐츠진흥원의 문화기술(CT)연구소 우수기술 활성화 사업의 연구결과로 수행되었음.

1. 서 론

증강현실 기술이 발전되면서 게임, 교육, 원격의료진단, 방송, 건축설계 등 다양한 분야에서 증강현실 기술이 응용되고 있다. 최근 스마트폰이 널리 보급되면서 모바일 솔루션·교육 분야 등에서도 다양하게 응용되고 있다. 특히 게임 분야에서는 증강현실 기술을 이용하여 기존의 입력 인터페이스인 조이스틱이나 키보드 없이 일상생활의 도구나 특정 형태의 카드를 사용하여 명령 및 제어를 할 수 있는 입력 인터페이스 연구가 활발히 진행되고 있다[1]. Sony PlayStation에서 개발한 Eye Pet[2]은 플레이어를 PlayStation EYE 카메라로 찍어 게임 화면 안에 등장시켜 가상의 애완동물 Pet과 함께 노는 증강현실 게임으로 플레이어의 손과 PlayStation Move Motion Controller를 사용하여 운동, 목욕, 놀아주기 기능 등을 할 수 있다. 또한 Android Application인 Sky Siege[3]라는 게임은 1인칭 시점의 증강현실 슈팅 게임으로 지상군이 되어 상/하/좌/우 등 360도 방향에서 공격해오는 적군의 비행기나 헬리콥터를 격추하는 게임이다. 게임의 조작은 원형의 가상 스크린이 있다고 생각하고, 360도 회전할 수 있는 의자에 앉아서 원하는 방향으로 향하게 되면, 그 방향의 화면을 볼 수 있게 조작이 된다.

이와 같이 증강 현실기술[4,5]을 응용하기 위해서는 관측공간에서 카메라의 위치나 가상 모델에 대한 트래킹이 필요한데, 이러한 트래킹 기술은 크게 마커기반 트래킹[6] (Marker-based Tracking)과 마커리스 트래킹[7,8](Markerless Tracking)으로 나뉘어 진다. 마커 기반 트래킹은 주로 특정한 형태의 마커를 사용하므로 인식이 용이하고 처리속도가 빠르다는 장점이 있지만, 시각적인 마커를 사용하므로 현실감이 떨어지고 마커의 영역이 차단되는 경우 트래킹에 실패한다는 단점이 있다. 반면에 마커리스 트래킹은 특정 형태의 마커를 사용하지 않고 영상에서 특징을 추출하므로 마커 기반 방법에 비교하여 좀 더 안정적인 트래킹이 가능하다는

장점이 있다. 하지만 마커리스 트래킹을 위해서는 추적 대상에 대한 특징정보인 특정 서술자를 미리 데이터베이스로 구성할 필요가 있는데, 추적대상이 증가 할수록 데이터베이스의 크기가 방대해져 특징 정보를 실시간으로 검색하기가 어렵다는 단점이 있다. 따라서 게임과 같이 다양한 멀티미디어 데이터를 동시에 다루면서 실시간 처리를 요구하는 응용에서는 아직까지 마커리스 트래킹을 적용하기가 어려운 실정이다.

본 논문에서는 앞서 언급한 문제점을 해결하여 실시간 마커리스 트래킹 및 증강 현실 환경 구축이 가능한 특정 서술자 데이터베이스 생성 및 검색 방법을 제안한다. 특정 서술자 데이터베이스는 SURF[9] 방법으로 추출된 특정 서술자와 특정 서술자에 대한 정수형태의 16Byte 인덱스와 4Byte 인덱스를 이용하여 총 4단계로 구성이 된다. 특정 서술자 데이터베이스를 생성하기 위해 먼저, 특정 서술자로부터 16Byte 인덱스를 생성하고, 16Byte 인덱스로부터 4Byte 인덱스를 생성한다. 생성된 4Byte 인덱스를 이용해 데이터베이스의 단계 1~2를 구성하고, 16Byte 인덱스를 이용해 단계 3을 구성, 마지막으로 특정 서술자로 단계 4를 구성한다. 데이터베이스의 각 단계는 배열의 형태로써, 단계 1은 유사성이 낮은 데이터들의 집합이고, 단계 2~4는 각 각 상위 단계와 유사성이 높은 데이터들의 집합이다.

특정 서술자 데이터베이스의 검색은 각 단계별로 4Byte 인덱스와 16Byte 인덱스를 사용하여 유사성 있는 특정 서술자를 탐색하고, 탐색된 모든 특정 서술자들의 유클리드 거리 값 비교를 통해 이루어진다. 인덱스를 사용한 특정 서술자 검색은 형태를 기반으로 유사하지 않은 특정 서술자들을 검색 대상에서 제외하여, 검색의 효율을 높였다.

제안된 방법은 일반적인 특정 서술자 데이터베이스 검색 방법인 KD-Tree 방법과 비교하여 특정 서술자당 16 msec 정도의 성능개선을 보였다. 특정 서술자가 일반적으로 장면 당 200개 검출된다고 가정했을 때 약 3.2초의 속도차이를 보여 성능

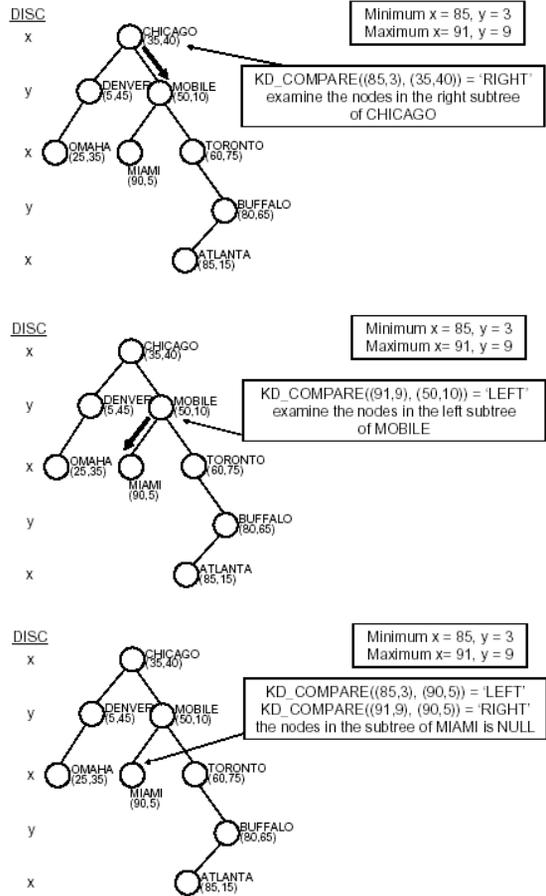
개선이 이루어졌음을 확인할 수 있었다. 본 연구 결과 약 8만개의 데이터베이스를 가지고 실험한 결과 초당 20프레임 이상으로 실시간으로 사용된다고 판단 할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구와 관련된 연구인 KD-Tree기반 데이터베이스 구축방법에 관하여 설명한다. 3장에서는 본 논문에서 제안하는 특징서술자의 데이터베이스 생성 및 검색 방법에 대해서 설명한다. 4장에서는 실험 결과에 대하여 기술하고, 5장에서 결론을 맺는다.

2. 관련 연구

본 장에서는 관련 연구로 SURF에 의한 특징점 추출방법[10]에 관하여 소개한다. SURF에 의한 특징점 추출방법은 일반적으로 속도가 빠른 KD-Tree 기반을 많이 사용한다.

KD-Tree는 데이터를 저장되어있는 데이터를 찾을수 있는 트리를 말한다. 본 방법은 이진검색트리를 확장한 것이므로 이진트리와 같이 각 노드의 데이터 값은 모두 달라야 한다. 그리고 최상위 레벨에 루트 노드가 있고, 각 노드는 최대 두 개의 자식 노드를 갖는다 또 임의의 노드의 데이터값은 자식의 왼쪽 자식 노드의 데이터 값보다 크고, 오른쪽 자식 노드의 데이터 값보다 작아야 한다. [그림 1]은 KD-Tree를 이용하여 특징 서술자를 검색하는 예이다.



[그림 1] KD-Tree를 통한 데이터 검색

본 방법은 각 필드를 차례대로 사용해서 트리를 검색해 내려가면 되기 때문에 검색이 쉽지만 트리를 구성하는 각 노드에 왼쪽과 오른쪽 서브트리에 대한 포인터를 저장할 링크 영역을 할당하여 표현하므로 부모노드에 접근하기가 어렵고 이진검색 트리와 마찬가지로 노드 개수에 따라 삭제 방법이 나누어 처리 되지만 다차원 이므로 이진검색트리보다 복잡하게 동작한다.

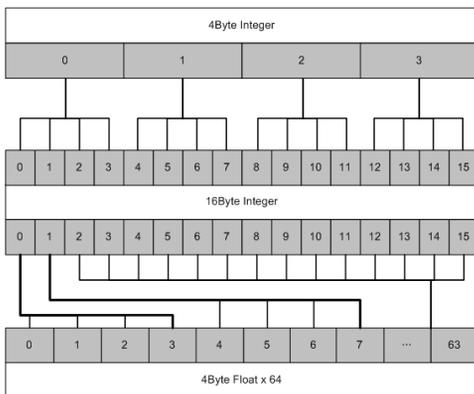
따라서 본 논문에서는 다량의 특징 서술자들을 실시간으로 검색하기 위해, 특징 서술자의 데이터베이스 검색 방법을 보완하여 기존의 방법인 KD-Tree와 유사한 정확도를 유지하면서 더 빠르게 검색 및 매칭이 가능한 특징서술자 검색 방법

을 제시하고 이와 성능 비교를 하여 결과를 보여 준다.

3. 특징 서술자의 데이터베이스 생성 및 검색 방법

3.1 특징 서술자의 부호화

일반적인 SIFT[7]와 SURF의 특징 서술자는 64~128 차원의 부동소수점 데이터로 구성되어 있기 때문에 특징 서술자 비교에 있어 정수 연산에 비해 상대적으로 높은 처리 비용이 든다. 따라서 검색의 가속화를 위해 64차원의 특징 서술자에 대해 2단계에 걸쳐 차원수를 줄이고, 부동 소수점 형태의 데이터를 정수 형태로 부호화 하여 유사 인덱스로 사용했다. [그림 2]는 부호화 과정을 통해 특징 서술자를 각각 16차원과 4차원 인덱스로 표현한 예 이다.

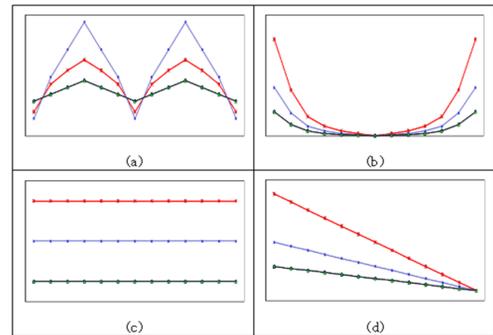


[그림 2] 3단계 부호화 과정

최초 단계에서 특징 서술자를 64차원의 부동소수점 데이터에서 16차원 정수 데이터로 변환되고, 다시 16차원 정수 데이터는 4차원 정수데이터로 변환된다. 이 때 서술자의 차원이 줄어들면서 손실된 데이터로 인해 특징 서술자 간에 중복이 발생하게 되는데, 이를 역 이용하여 유사한 특징 서술

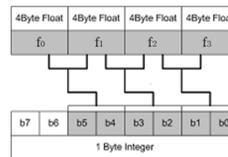
자를 동일한 정수 데이터로 변환하여 그룹화하고, 해당 정수 데이터를 특징 서술자 그룹의 대표 인덱스로 사용했다.

[그림 3]은 형태가 유사한 특징 서술자의 예를 보여준다. [그림 3]의 (a)-(d)는 각각 서로 다른 특징 서술자를 그래프화 한 것인데, 모두 상이한 데이터로 구성되어 있으나, 특징 서술자의 각 원소 간 증감의 방향이 동일한 것을 알 수 있다. 이와 같이 유사한 형태로 간주되는 특징 서술자들을 동일한 정수로 부호화 하여 대표 인덱스로 사용하면, 인덱스 비교만으로도 대략적인 유사성 판단이 가능하여, 검색의 효율을 높일 수 있다.

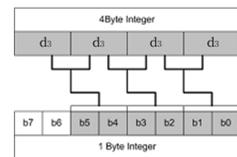


[그림 3] 특징 서술자 원소들의 증감 방향을 이용한 분류

형태를 기반으로 부동 소수점 데이터에서 정수 데이터로 변환하는 구체적인 방법은 [그림 4]와 [그림 5]와 같다.



[그림 4] 16Byte 인덱스 생성



[그림 5] 4Byte 인덱스 생성

$$d = f_{n+1} - f_n \text{ for each } n = 0, 1, 2, \dots \text{ (식 1)}$$

$$Bits = \begin{cases} 11 & d > t \\ 00 & d < t \\ 01 & otherwise \end{cases} \dots \text{(식 2)}$$

특징 서술자의 64개 원소를 [그림 4]와 같이 4개의 단위로 잘라 1Byte 정수로 부호화 한다. 먼저 (식 1)를 이용하여 특징 서술자의 인접된 각 원소 f 간 차이 d 를 구하고, (식 2)을 이용하여 임계값 t 와 비교하여 부호화할 비트를 결정, 해당 비트 값을 [그림 4]에서 보는 바와 같이 1Byte 정수 데이터에 저장한다. 그림에서 보는 바와 같이 상위 2비트는 사용되지 않는다. 특징 서술자 64개 원소들에 대해서 4개 단위로 같은 연산을 수행하면 총 16 Byte의 정수 인덱스가 생성되고, 16Byte 정수 인덱스에 대해서 [그림 5]와 같이 다시 한 번 부호화 과정을 거치면 4Byte 정수 인덱스가 생성되어, 하나의 특징 서술자에 대해 4Byte 인덱스와 16Byte 인덱스가 생성된다.

3.2 특징 서술자의 유사성 판단

64차원 특징 서술자의 부호화 단계를 거치면, 4Byte 인덱스와 16Byte 인덱스가 생성되고, 유사성을 판단하기 위해 각각의 인덱스와 특징 서술자를 비교하는 방법이 필요로 하다. 일반적으로 특징 서술자의 경우 유클리드 거리(Euclidean distance)를 비교하여 일정 임계값 미만의 값을 가지면 유사한 디스크립터로 판단한다.

$$d = \sum_{k=0}^n (f_k - f'_k)^2 \dots \text{(식 3)}$$

$$likeness = \begin{cases} Near & d < t \\ Far & d > t \end{cases} \dots \text{(식 4)}$$

(식 3)은 유클리드 거리 값 비교를 통해 두 특징 서술자의 유사도를 판단하는 것으로, n 은 특징 서술자의 차원수이고, f_k 와 f'_k 는 각각 비교할 특징 서술자의 k 번째 원소이다. 각 원소간의 차를 제

곱하여 합산한 값이 임계값 d 미만일 경우 유사한 특징 서술자로 판단한다.

하지만 정수 데이터로 부호화된 인덱스들에 대해서는 동일한 방법을 적용할 수 없으므로, 다음의 방법을 통하여 인덱스간의 유사도를 비교한다.

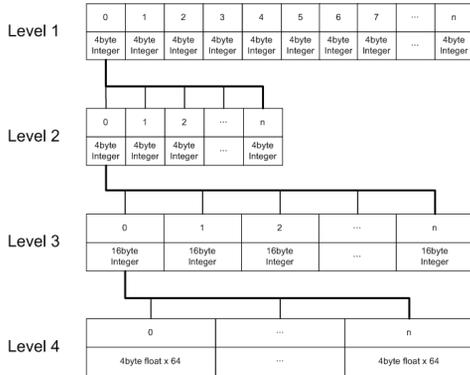
$$bitDistance = CountOfTrueBits(i \oplus i') \dots \text{(식 5)}$$

$$likeness = \begin{cases} Near & bitDistance < t \\ Far & bitDistance \geq t \end{cases} \dots \text{(식 6)}$$

정수 인덱스는 특징 서술자의 방향을 비트로 부호화한 결과 이므로 단순한 대소 비교가 불가능하다. 하지만 유사한 특징 서술자를 부호화한 결과의 비트열이 유사하게 나타난다는 점을 이용하여, 비트단위로 차이를 구하여 인덱스의 유사성을 판단한다. (식 5)에서 i 와 i' 는 각각 비교할 정수 인덱스를 의미하고, exclusive-OR 연산으로 두 정수를 비트 단위로 비교하여 각 비트값이 서로 다른 비트의 개수를 구한다. 다음으로 (식 6)을 이용하여 임계값 t 미만의 차이를 보이는 경우에 대해서 정수 인덱스간의 유사성이 있다고 판단하게 된다.

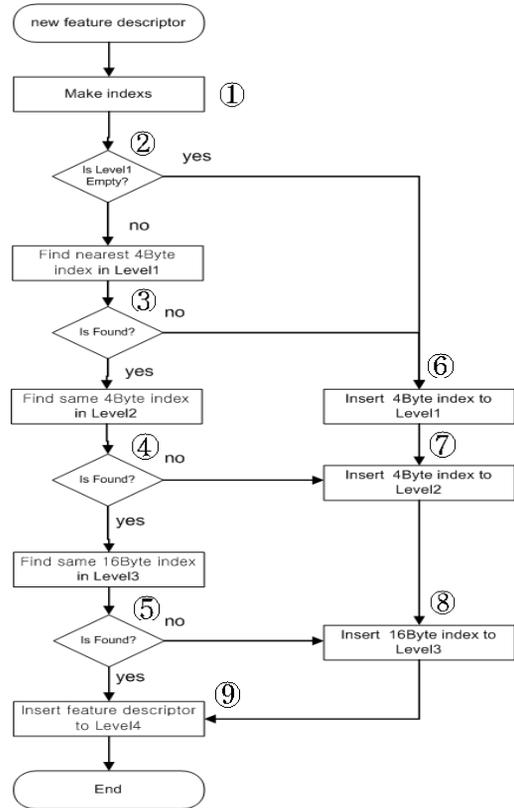
3.3 특징 서술자 데이터베이스 생성

특징 서술자 데이터베이스는 정수 데이터로 부호화된 인덱스와 원본 특징 서술자를 사용하여 총 4 단계로 구성된다. [그림 6]은 특징 서술자 데이터베이스의 구조도를 보여주는 예 이다.



[그림 6] 특정 서술자 인덱싱을 위한 다단계 배열 구조

단계 1은 가장 단순화된 4Byte 인덱스의 배열로, 각 원소는 유사성이 멀다고 판단되는 대표 인덱스들로 구성되고, 단계 2는 마찬가지로 4Byte 인덱스의 배열로, 단계 1의 대표 인덱스와 유사성이 있다고 판단되는 인덱스들로 구성되어 있다. 단계 3은 16Byte 인덱스 배열로, 부호화의 결과인 4Byte 인덱스가 동일하게 생성되는 16Byte 인덱스들의 집합이며, 단계 4는 최초의 원본 특정 서술자들의 배열로, 부호화의 결과인 16Byte 인덱스가 동일하게 생성되는 특정 서술자 집합이다. 이와 같이 단계 1에서 단계 4까지 유사성을 기준으로 인덱스 및 특정 서술자가 분류 저장되기 때문에 유사도의 기준이 되는 임계값에 따라 데이터베이스의 구조가 결정된다고 할 수 있다. 임계값이 작은 경우 대부분의 인덱스가 그룹화 되지 못하여 Level1로 데이터가 편중되어 저장되며, 임계값이 큰 경우 대부분의 인덱스가 하나의 그룹으로 모이기 때문에 Level2로 데이터가 편중되어 데이터베이스가 전반적으로 균형을 이루지 않게 된다. 다음 [그림 7]은 특정 서술자 데이터베이스 생성의 전체 과정을 나타낸다.



[그림 7] 데이터베이스 생성 흐름도

- ① 데이터베이스에 삽입될 새로운 특정 서술자가 입력으로 들어오면, 앞서 언급한 방법으로 4Byte 인덱스와 16Byte 인덱스로 부호화 한다.
- ② Level1 배열의 크기가 0 이라면, ⑥으로 분기한다.
- ③ Level1 배열에서 유사하다고 판단되는 4Byte 인덱스를 찾지 못한 경우 ⑥으로 분기한다.
- ④ 유사성이 있는 4Byte 인덱스의 자식인 Level2 배열에서 동일한 4Byte 인덱스를 찾아, 존재하지 않는 경우 ⑦로 분기한다.
- ⑤ Level2 4Byte 인덱스의 자식인 Level3 배열에서 동일한 16Byte 인덱스를 찾아, 존재하지 않는 경우 ⑧로 분기하고, 존재한다면 ⑨로 분기한다.
- ⑥ 4Byte 인덱스를 Level1 배열의 마지막에 추가

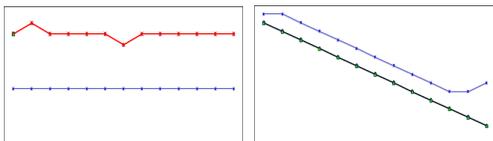
하고, Level2에 해당하는 자식배열을 생성 한 뒤, Level1 배열의 원소와 Level2 배열을 연결 한다.

- ⑦ 4Byte 인덱스를 Level2 배열의 마지막에 추가 하고, Level3에 해당하는 자식배열을 생성 한 뒤, Level2 배열의 원소와 Level3 배열을 연결 한다.
- ⑧ 16Byte 인덱스를 Level3 배열의 마지막에 추가 하고, Level4에 해당하는 자식배열을 생성 한 뒤, Level3 배열의 원소와 Level4 배열을 연결 한다.
- ⑨ Level4 배열의 마지막에 원본 특징 서술자를 추가한다.

3.4 특징 서술자 검색

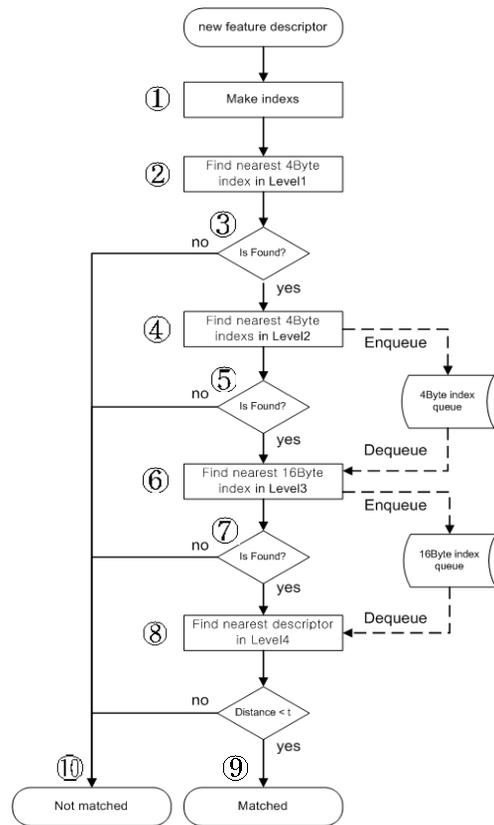
데이터베이스에서 특징 서술자를 검색 하는 방법은 데이터베이스 생성 단계와 유사하게 처리된다. 데이터베이스의 최초 단계인 Level1부터 유사도가 높은 데이터를 기준으로 하위 단계로 내려가며 유사도를 비교하고, 마지막 단계인 Level4에서는 원본 특징 서술자들의 유클리드 거리 값을 비교하여 그 중 최소의 거리를 가지는 특징 서술자들을 매칭후보로 선정한다.

특징 서술자들의 데이터는 특징점 주변의 형태를 기술하기 때문에, 다소 비교를 통한 정렬을 하기에는 부적합한 형태이다. 따라서 데이터베이스의 각 단계를 특징 서술자들의 유사도를 기준으로 그룹화 하였으나, [그림 8]과 같이 특징 서술자를 추출하는 과정에서 잡음이 포함되는 경우, 실제 매칭이 가능한 두 개의 서술자가 서로 다른 대표 그룹에 속하는 오차가 발생하였다.



[그림 8] 오차를 포함한 특징 서술자의 증감 형태

이와 같은 문제를 해결하기 위해, 검색 시 그룹의 대표 인덱스를 비교하여 유사도가 일정 임계값 이상이 되는 모든 그룹을 검색 대상에 포함하여, 실시간 처리가 가능한 처리 속도를 보이면서 특징 서술자 검색의 정확도를 높였다. [그림 9]는 특징 서술자 데이터베이스에서 검색을 수행하는 순서도이다.



[그림 9] 특징 서술자 검색 흐름도

- ① 데이터베이스와 비교할 새로운 특징 서술자가 입력으로 들어오면 앞서 언급한 방법으로 4Byte 인덱스와 16Byte 인덱스로 부호화 한다.
- ② Level1 배열에서 4Byte 인덱스를 이용하여 가장 유사한 원소를 찾는다.
- ③ 4Byte 인덱스간의 거리 값이 임계값 이상이면, ⑩으로 분기한다.

- ④ Level2 배열에서 4Byte 인덱스간의 거리 값이 임계값 미만인 모든 원소를 큐1에 삽입한다.
- ⑤ 큐1이 비어 있다면 ⑩으로 분기한다.
- ⑥ 큐1의 모든 원소에 대해서 자식인 Level3 배열을 16Byte 인덱스로 탐색하여, 인덱스간 거리 값이 임계값 미만인 모든 원소를 큐2에 삽입한다.
- ⑦ 큐2가 비어 있다면 ⑩으로 분기한다.
- ⑧ 큐2의 모든 원소에 대해서 자식인 Level4 배열을 탐색하며, 원본 특징 서술자와 데이터베이스간의 유클리드 거리 값을 비교하여, 거리 값이 최소인 특징 서술자를 찾는다.
- ⑨ 최소 거리 값이 임계값 보다 작은 경우, 입력된 특징 서술자와 현재 특징 서술자가 매칭 되었다고 가정하고 처리를 종료한다.
- ⑩ 입력된 특징 서술자를 매칭 후보에서 제외하고 처리를 종료한다.

4. 실험

4.1 하드웨어 구성

본 논문에서 제시한 시스템으로 실시간 처리가 가능한지의 여부를 판단하기 위해 AMD Phenom(tm) II X4 920 2.81GHz의 CPU를 장착한 PC에서 측정하였다.

4.2 구현 결과

제안된 방법이 실시간 처리에 적합한지 판단하기 위해 기존의 특징 서술자 검색 방법과 비교하여, 성능을 검증하였다. 실험용 카메라를 사용하여 동적으로 변화하는 화면을 얻어 특징 서술자 약 10만개를 추출하여 데이터베이스로 구성하고, 검색의 성능을 비교하기 위해 KD-Tree 방법 과 본 논문에서 제안하는 방법으로 각각 검색하여 속도를 측정하였으며, 타 연구[10] 결과의 성능을 포함하여 [표 1]로 정리하였다.

[표 1] 데이터베이스 검색시간 비교

	전체 검색 (타연구 인용)	KD-Tree (타연구 인용)	KD-Tree (범위=3)	KD-Tree (범위=0.5)	제안된 방법
DB Size	약 6만개	약 6만개	약 10만개	약 10만개	약 10만개
평균	8.043ms	4.199ms	16.61ms	0.3ms	0.2ms
최대	-	-	21.94ms	0.88ms	2.14ms
최소	-	-	6.58ms	0.007ms	0.009ms

비교 결과 타 연구의 검색시간에 비해서는 데이터베이스의 크기를 고려하지 않더라도 약 20~40배의 성능 개선이 이루어 졌음을 확인할 수 있다. 그리고 동일한 성능의 하드웨어에서의 비교를 위해 직접 구현한 KD-Tree 방법과 비교하면 약 1.5~80배의 성능 개선을 확인할 수 있다. KD-Tree 방법은 데이터베이스의 검색 범위가 적은 경우 제안된 방법보다 비교적 빠른 검색 시간을 보였으나 범위가 넓어짐에 따라 검색 시간의 증가 폭이 매우 높은 특징을 보였는데, 이는 영상 내 잡음 요소를 고려하여 특징 서술자를 매칭 하는 경우 검색 범위가 넓어져 성능저하의 원인으로 작용한다. 또한 검색 범위가 넓은 경우 여러 개의 특징 서술자가 검색되므로 결과 내에서 유클리드 거리가 가장 작은 서술자를 다시 검색하기 위해 추가적인 검색 시간이 필요로 하다는 문제점이 발생하게 된다.

따라서 KD-Tree 방법은 데이터베이스의 검색 범위가 적은 경우 유리하고 본 방법은 검색범위가 넓은 경우 유리하다. 일반적으로 영상내 잡음이 있는 경우 검색범위가 넓어져서 여러개의 특징서술자가 검색되므로 본 방법이 성능적인 면에서 효율적이다

본 논문에서 제안한 방법을 증강현실 기반 발달장애아동을 위한 상황훈련 서비스 중 돌발행동에 대한 상황인식을 수행하는데 적용한 결과 초당 평균 20프레임 이상으로 실시간으로 수행할 수 있음을 알 수 있었다. [그림 10]은 발달장애 아동의 상황훈련 수행과정을 찍은 사진이다.



[그림 10] 발달장애아동을 위한 상황훈련

5. 결 론

본 연구는 증강현실 응용을 위하여 다량의 데이터베이스에서 실시간으로 특징 서술자들을 검색하기 위해, 데이터베이스 검색 방법을 보완하여 기존의 방법인 KD-Tree와 유사한 정확도를 유지하면서 더 빠르게 검색 및 매칭이 가능한 데이터베이스 생성 및 검색 방법을 제안한다. 먼저 특징 서술자 데이터베이스는 정수 데이터로 부호화된 인덱스와 유사한 특징 서술자를 인덱스 별로 그룹화 하여 총 4 단계로 구성한다. 특징 서술자 데이터베이스의 검색은 각 단계별로 4Byte 인덱스와 16Byte 인덱스를 사용하여 유사성 있는 특징 서술자를 탐색하고, 탐색된 모든 특징 서술자들의 유클리드 거리 값 비교를 통해 이루어진다. 인덱스를 사용한 특징 서술자 검색은 형태를 기반으로 유사하지 않은 특징 서술자들을 검색 대상에서 제외하여, 검색의 효율을 높였다.

제안된 방법은 전체 검색 방법과 다른 연구의 실험결과로 인용된 KD-Tree 방법과 비해서 각 특징서술자 당 약 16ms의 검색 속도 개선이 있었음을 확인할 수 있었으나, 전체 검색 방법에 비해서 매칭의 정확도가 떨어질 수 있는 확률이 있으므로 향후 잡음이 포함된 특징 서술자 검색 시 매칭확률을 높이는 연구가 필요하다.

참고문헌

- [1] <http://us.playstation.com/eyeofjudgment/>
- [2] <http://us.playstation.com/games-and-media/games/eyes/eyepet-ps3.html>
- [3] http://www.androidzoom.com/android_games/arcade_and_action/sky-siege_icbp.html
- [4] 최영주, 서영덕, 박준, 이상국 “A Study to develop the augmented reality navigation system that can be applied to automobile,” Journal of the Korea society of computer and information vol.6 , no.3, pp.139-148, 2008.
- [5] Juan, M., Botella, Alcaniz, M., Banos, R., Carrion, Melero, M., Lozano, J.A., An Augmented Reality System for treating psychological disorders: Application to phobia to cockroaches, proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2004), 2004
- [6] Kato, H., Billinghurst, M. “Marker Tracking and HMD Calibration for a video-based Augmented Reality Conferencing System”, IWAR'99, 2nd IEEE and ACM International Workshop on (1999), pp. 85-94. 1999
- [7] G. Klein and D. Murray, “Parallel Tracking and Mapping for Small AR Workspaces”, Proc. Int'l Symp. Mixed and Augmented Reality (ISMAR '07), 2007.
- [8] David G. Lowe, “Distinctive image features from scale-invariant keypoints”, International Journal of Computer Vision, 60, 2, pp.91-110, 2004
- [9] J. J. Foo, J. Zobel, R. Sinha, and S. Tahaghoghi, “Detection of near-duplicate images for web search,” In Proceedings of the 6th ACM International Conference on Image and Video Retrieval, pp.557-564, 2007.
- [10] 복윤수, 황영배, 권인소, “영상 매칭 및 자세 추정을 이용한 무인 차량의 위치 추정”, 한국군사과학기술학회 종합학술대회, August 2007



윤 요 섭 (Yun, Yo Seop)

2009년 2월 서경대학교 컴퓨터공학과 학사
2011년 2월 서경대학교 전자컴퓨터공학과 석사
2011년 3월-현재 XL GAMES 연구원

관심분야 : 증강현실 컴퓨터 그래픽스



김 태 영 (Kim, Tae Young)

1991년 2월 이화여자대학교 전자계산학과 학사
1993년 2월 이화여자대학교 전자계산학과 석사
1993년 3월-2002년 2월 한국통신 멀티미디어연구소
신임 연구원
2001년 8월 서울대학교 전기컴퓨터공학부 박사
2002년 3월-현재 서경대학교 컴퓨터공학과 부교수

관심분야 : 증강현실, 실시간 렌더링, 모바일3D
