3차원 물체 재구성 과정이 통합된 실시간 3차원 특징값 추출 방법

(Real-time 3D Feature Extraction Combined with 3D Reconstruction)

홍 광 진 † 이 철 한 † 정 기 철 †† 오 경 수 ††
(Kwangjin Hong) (Chulhan Lee) (Keechul Jung) (Kyoungsu Oh)

요 약 상호작용이 가능한 컴퓨팅 환경에서 사람과 컴퓨터 사이의 자연스러운 정보 교환을 위해 동작 인식과 관련한 연구가 활발하게 이루어지고 있다. 기존의 2차원 특징값을 이용하는 인식 알고리즘은 특징값 추출과 인식 속도는 빠르지만, 정확한 인식을 위해서 많은 환경적인 제약이 따른다. 또한 2.5차원 특징값을 이용하는 알고리즘은 2차원 특징값에 비해 높은 인식률을 제공하지만 물체의 회전 변화에 취약하고, 3차원 특징값을 이용하는 인식 알고리즘은 특징값 추출을 위해 3차원 물체를 재구성하는 선행 과정이 필요하기 때문에 인식 속도가 느리다. 본 논문은 3차원 물체 재구성 단계와 특징값 추출 단계를 통합하여 실시간으로 3차원 정보를 가지는 특징값 추출 방법을 제안한다. 제안하는 방법은 기존의 GPU 기반 비주얼 헐 생성 방법의 세부 과정 중에서 동작 인식에 필요한 데이타 생성 부분만을 수행하여 임의의 시점에서 3차원 물체에 대한 3종류의 프로젝션 맵을 생성하고, 각각의 프로젝션 맵에 대한 후-모멘트(Hu-moment)를 계산한다. 실험에서 우리는 기존의 방법들과 단계별 수행 시간을 비교하고, 생성된 후-모멘트에 대한 혼동 행렬(confusion matrix)을 계산함으로써 제안하는 방법이 실시간 동작 인식 환경에 적용될 수 있음을 확인하였다.

키워드 : 실시간 3차원 특징값 추출, 동작 인식, 컴퓨터 비전, 프로젝션 맵, 비주얼 헐

Abstract For the communication between human and computer in an interactive computing environment, the gesture recognition has been studied vigorously. The algorithms which use the 2D features for the feature extraction and the feature comparison are faster, but there are some environmental limitations for the accurate recognition. The algorithms which use the 2.5D features provide higher accuracy than 2D features, but these are influenced by rotation of objects. And the algorithms which use the 3D features are slow for the recognition, because these algorithms need the 3d object reconstruction as the preprocessing for the feature extraction. In this paper, we propose a method to extract the 3D features combined with the 3D object reconstruction in real-time. This method generates three kinds of 3D projection maps using the modified GPU-based visual hull generation algorithm. This process only executes data generation parts only for the gesture recognition and calculates the Hu-moment which is corresponding to each projection map. In the section of experimental results, we compare the computational time of the proposed method with the previous methods. And the result shows that the proposed method can apply to real-time gesture recognition environment.

Key words : Real-time 3D Feature Extraction, Gesture Recognition, Computer Vision, Projection Map, Visual Hull

· 이 논문은 2005년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원(KRF-2005-003-D00336)과 2006년도 한국과학재단의 특정기초 연구 프로그램의 지원(R01-2006-000-11214-0)을 받아 연구되었음
· 이 논문은 2008 한국컴퓨터종합학술대회에서 '명시적인 3차원 물체 재구성 과정이 생략된 실시간 3차원 특징값 추출 방법'의 제목으로 발표된 논문을 확장한 것임

† 학생회원 : 숭실대학교 미디어학과
hongmsz@ssu.ac.kr
dashans@ssu.ac.kr
†† 종신회원 : 숭실대학교 미디어학과 교수
kcjung@ssu.ac.kr
oks@ssu.ac.kr

논문접수 : 2008년 8월 25일
심사완료 : 2008년 10월 21일

정보과학회논문지 : 소프트웨어 및 응용 제35권 제12호(2008.12)

1. 서 론

상호작용이 가능한 컴퓨팅 환경의 보급과 함께 보다 자연스러운 컴퓨터 사용 환경을 제공하기 위해 사용자 동작 인식과 관련된 연구가 활발히 이루어지고 있다. 일반적으로 동작 인식을 위한 알고리즘의 성능은 인식 속도와 인식률을 기준으로 평가가 이루어지는데, 이 두 가지 평가 요소는 알고리즘에서 사용하는 특징값의 종류와 그 특징값을 추출 방법에 많은 영향을 받는다.

가장 기본적인 컴퓨터 비전 기반의 동작 인식 방법은 한 대의 카메라를 통해 입력받은 2차원 영상에서 추출된 특징값을 이용하는 방법이다. 이 방법은 특징값 추출과 비교를 위한 계산량이 적기 때문에 빠른 인식 결과를 얻을 수 있다. 그러나, 2차원 정보로 구성된 영상에서 추출된 특징값은 3차원 공간 안에 존재하는 인식 대상을 표현하는 데에 한계가 있기 때문에, 일반적으로 카메라의 위치와 시점이 고정된 환경에 적용하여 사용한다[1,2]. 이러한 문제를 해결하기 위해, 다양한 시점에서 촬영된 영상을 이용한 특징값 추출 방법이 제안되었다[3]. 이 방법은 다시점 카메라를 통해 입력받은 다수의 영상 세트와 미리 저장된 다양한 동작에 대한 실제 또는 가상의 다시점 카메라 영상 세트를 비교하여 가장 많은 수의 영상이 일치하는 세트를 입력 영상의 동작으로 결정한다. 그러나, 인식률은 카메라 시점의 수가 증가할수록 높아지기 때문에 특징값 비교 시간이 오래 걸린다는 단점이 있다.

스테레오 영상은 평면적인 2차원 영상에 깊이 값을 추가함으로써 인식 대상에 대한 부분적인 3차원 정보를 추출할 수 있고, 이를 동작 인식 시스템에 적용할 경우 보다 높은 인식률을 얻을 수 있다[4,5]. 그러나 스테레오 영상은 카메라에서 보이는 면에 대한 정보만 추출이 가능하기 때문에 인식 대상의 이동, 회전 변화에 영향을 많이 받게 된다.

최근에는, 보다 다양한 동작에 대한 정확한 인식 결과를 얻기 위해, 인식 대상을 3차원 물체로 재구성한 뒤 특징값을 추출하여 동작을 인식하는 다양한 방법이 제안되고 있다[6-13]. 재구성된 3차원 물체는 물체를 구성하는 각 부분에 대한 명시적인 3차원 데이타를 가지고 있기 때문에, 3차원 공간 안에서의 사람의 동작을 잘 표현할 수 있는 특징값을 추출할 수 있고, 2차원 영상을 이용하는 방법에 비해 보다 정확한 인식 결과를 얻을 수 있다.

표 1은 재구성된 3차원 물체를 이용하는 특징값 추출 방법들을 특징값의 종류와 사용된 알고리즘을 기준으로 분류하고, 각 알고리즘들을 이용하여 특징값 추출 시간을 비교한 결과를 정리한 것이다. 표 1에서의 추출 시간은 3차원 물체가 재구성된 후 3차원 정보를 이용하여 특징값을 추출하기 위해 걸리는 대략적인 시간을 의미하며, 각 알고리즘마다 3차원 물체를 재구성하는 방법이 다르기 때문에 보다 객관적인 비교를 위해 3차원 물체 재구성 시간은 포함시키지 않았다.

재구성된 3차원 물체를 이용하는 알고리즘은 추출되는 특징값의 종류에 따라 크게 그래프 기반과 히스토그램 기반의 방법으로 구분할 수 있다. 그래프 기반 특징값 추출 알고리즘[9-13]은 3차원 물체를 뼈대 형태로 표현하기 때문에 신체 각 부분들 간의 관계와 위치를 계산할 수 있고 따라서 매우 정확한 동작 인식이 가능하며, 특히 고해상도의 입력 영상을 이용하여 3차원 물체를 재구성할 경우 추출된 특징값 중 신체 각 부분을 의미하는 일부 값만을 이용하여 비교하는 부분 영역 인식(partial matching)도 가능하다는 장점이 있다. 그러나 그래프 기반의 방법들은 골격 생성을 위해 3차원 물체를 구성하는 모든 점을 탐색하기 때문에 특징값 추출 시간이 매우 오래 걸린다는 단점을 가진다. spherical harmonic 알고리즘[8]과 3차원 bin-distribution 알고리즘[6,7] 등의 히스토그램 기반 특징값 추출 알고리즘은 물체를 구성하는 3차원 점들의 분포를 나타내는 히스토그램을 이용하기 때문에 그래프 기반 알고리즘에 비해 특징값을 추출하는 시간이 적게 걸린다. 그러나, 물체를 구성하는 점들 간의 관계를 고려하지 않고 분포만으로 3차원 물체를 표현하기 때문에, 물체의 종류(의자, 비행기, 자동차 등)를 나타내는 전체적인 특징의 표현만 가능하고, 각각의 종류 안에서 세부적인 차이점을 표현하는 것은 어렵다. 또한, 3차원 특징값 추출 시간이 빨라

표 1 재구성된 3차원 물체를 이용한 동작 인식 알고리즘의 3차원 특징값 추출 시간 비교

추출된 특징값	특징값 추출을 위한 알고리즘	저자[논문]	추출 시간(초)
히스토그램	3차원 Bin-distribution	C. Chu and I. Cohen [6]	≤ 0.1
		D. Kyoung et al. [7]	≤ 1
	Spherical harmonic	T. Funkhouser et al. [8]	≤ 1
그래프	Reeb graph	M. Hilaga et al. [9]	1
	3차원 세선화	H. Sundar et al. [10]	10
	Curve-skeleton	N. D. Cornea et al. [11,12]	10^3
		A. Brennecke and T. Isenberg [13]	10^3

졌음에도 불구하고, 앞서 언급한 것과 같이 3차원 물체 재구성 시간이 여전히 필요하기 때문에, 기존에 제안된 그래프 및 히스토그램 기반 방법들은 3차원 물체에 대한 실시간 인식 환경(초당 15장 이상의 영상 처리가 가능한 환경)에 적용하기 어렵다.

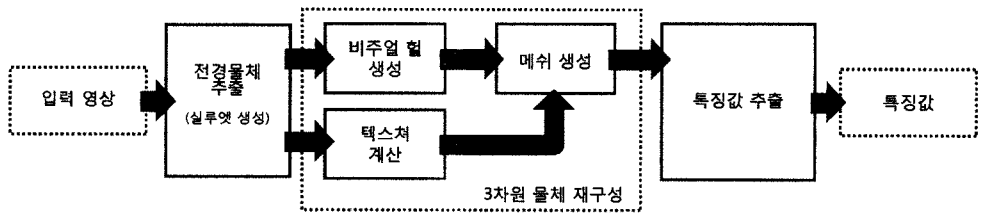
그림 1에서 보는 것처럼, 우리는 기존의 3차원 물체를 이용한 특징값 추출 방법과 달리, 3차원 물체 재구성 과정과 특징값 추출 과정을 통합하여 실시간 3차원 특징값 추출이 가능한 방법을 제안한다. 제안하는 방법은 3차원 물체에 대한 프로젝션 맵을 이용하여 3차원 정보를 포함하는 특징값을 추출한다. 프로젝션 맵은 높은 차원의 공간에 존재하는 데이타들을 낮은 차원의 공간으로 투영하고 그 값을 누적하여 생성되는 2차원 영상으로 영상의 각 픽셀은 재구성된 물체의 3차원 정보로 구성된다. 프로젝션 맵의 생성을 위해 우리는 GPU 기반의 비주얼 헐 렌더링 방법[14]을 사용한다. GPU 기반의 비주얼 헐 렌더링 방법은 GPU 내부의 렌더링 파이프라인을 따라 임의의 시점에 대한 비주얼 헐을 렌더링하기 때문에, 렌더링 과정 중에 파이프라인 내부에서 깊이 버퍼(z-버퍼)에 저장된 데이타를 이용하여 프로젝션 맵을 생성할 수 있다. 제안하는 방법에서 프로젝션 맵을 이용하여 최종적으로 추출되는 특징값은 7개의 데이타로 구성되는 후-모멘트(Hu-moment)[15]이다. 우리는 프로젝션 맵을 이용한 후-모멘트 계산을 GPU 내부에서 수행함으로써 3차원 물체 재구성 단계와 특징값 추출 단계를 통합한다.

논문의 구성은 다음과 같다. 2장에서는 비주얼 헐에 대해 설명하고, 3장에서는 제안한 방법을 구성하는 각 단계인 실루엣 추출(3.1절)과 비주얼 헐 렌더링(3.2절)에 대해 상세하게 설명한다. 4장에서는 실험 결과를 보여주고, 5장에서 제안한 방법에 대한 전체적인 결론을 이야기한다.

2. 비주얼 헐(Visual Hull)

컴퓨터 비전 기반의 3차원 특징값 추출 알고리즘은 특징값 추출을 위해 인식 대상을 온라인 상의 3차원 물체로 재구성하는 과정이 필요하다. 3차원 물체를 재구성하기 위해 기존의 연구들은 주로 다시점 입력 영상을 이용하여 비주얼 헐을 생성하는 방법[16]을 이용하는데, 비주얼 헐 생성 알고리즘은 미리 동기화 된 다양한 시점의 카메라 입력 영상을 이용하여 비교적 간단하게 실제 물체에 근접한 3차원 물체를 재구성하고 시각화할 수 있다는 장점을 가지기 때문이다. 비주얼 헐을 생성하기 위해, 다시점 카메라를 통해 입력받은 영상 내의 물체에 대한 실루엣 영상을 생성하여 실루엣에 대한 외곽선을 추출하고, 각 외곽선 상의 점을 카메라 캘리브레이션 데이타를 이용하여 3차원 공간으로 역투영(back-projection)시킨다. 이렇게 역투영된 실루엣 영상들은 3차원 공간 안에서 원뿔(cone)의 형태를 띄게 되는데, 공간 안에서 모든 실루엣 콘들이 교차하는 점을 계산하여 얻어지는 결과가 비주얼 헐이다(그림 2).

일반적으로 비주얼 헐을 생성하기 위해서 복셀 볼륨 기반[17]이나 폴리곤 메쉬 기반[18] 방법을 사용하는데, 이들 방법은 비주얼 헐을 구성하는 모든 점의 기하 정보를 재구성한 후 임의의 시점에 대한 영상을 렌더링하기 때문에 재구성된 3차원 물체에 새로운 동작을 적용시키거나, 형태를 변화시켜 새로운 물체를 생성할 수 있다는 특징이 있다. 그러나 단순히 입력 영상 내의 사물



(a) 기존의 특징값 추출 과정

(b) 제안된 방법

그림 1 3차원 특징값 추출 과정 비교

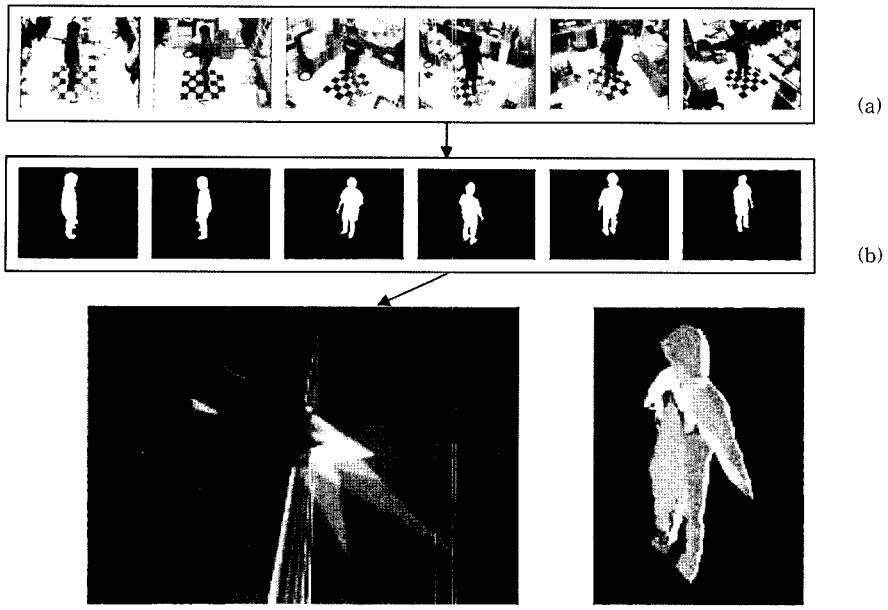


그림 2 비주얼 헐 생성 과정: (a) 다시점 카메라 입력 영상, (b) (a)에 대한 실루엣 영상, (c) 실루엣 영상을 이용하여 생성된 실루엣 콘들의 교차 영역, (d) 생성된 비주얼 헐

을 임의의 시점에 대해 렌더링하여 새로운 시점의 영상을 생성하는 것이 목표인 경우, 비주얼 헐의 모든 점을 계산하는 것은 매우 비효율적이다.

3. 특징값 추출

추출에 이용될 수 있다. 본 논문에서 사용하는 특징값은 3차원 공간 안의 물체를 구성하는 모든 점을 하나의 2차원 평면 위로 투영시키고 이를 누적하여 얻어지는 프로젝션 맵에서 추출한 후-모멘트이다. 프로젝션 맵은 물체를 구성하는 점들의 분포와 밀도를 이용하여 2차원 평면에 물체를 표현하기 때문에, 점들간의 관계를 이용하는 그래프 기반의 특징값들과 달리 간단하게 생성할 수 있으며, 히스토그램 기반 특징값들과 달리 사람의 머리, 팔, 다리와 같이 물체를 구성하는 각 부분의 위치 변화를 효율적으로 표현하는 것이 가능하다.

제안하는 방법은 GPU를 이용하여 비주얼 헐을 렌더링하는 과정에서 특징값의 추출이 가능하다. 이 방법은 비주얼 헐을 구성하는 모든 점의 기하정보를 계산할 필요가 없기 때문에, HAVH(Hardware-Accelerated Visual Hull) 방법을 이용하여 입력 영상에 나타난 물체의 실루엣으로부터 임의의 카메라 시점에서 보이는 비주얼 헐 영역만을 렌더링하게 되는데, 렌더링 파이프라인 내부의 래스터라이징 단계와 픽셀 쉐이더를 거치면 렌더링 결과(새로운 카메라 시점의 영상)를 생성하기 전에 해당 물체의 3차원 정보(카메라 촬상면으로부터 가장 가까운 경계면까지의 거리, 가장 먼 경계면까지의 거리, 3차원 물체의 두께)를 포함하는 프로젝션 맵을 얻을 수 있고(그림 3), 후-모멘트의 계산도 행렬 연산으로 변환하면 정점(vertex) 쉐이더 내부의 좌표계 변환 단계에서 수행이 가능하다.

따라서, 제안하는 방법은 크게 비주얼 헐 렌더링을 위한 입력 데이타로 사용될 실루엣 영상을 추출하는 단계와 렌더링 파이프라인 내부에서 임의의 카메라 시점에 대한 프로젝션 맵을 생성하고 후-모멘트를 계산하는 단계로 나눌 수 있다(그림 4).

3.1 실루엣 추출

다시점 카메라로부터 입력받은 영상에서 배경과 물체를 구분하여 실루엣 영상을 생성한다. 본 논문에서 우리는 실루엣 영상을 생성하기 위해서 차영상 알고리즘을 사용한다. 촬영 공간이 비어있는 상태에서 촬영된 영상을 배경 영상(I_b)으로 미리 저장하여두고, 사람이 들어와서 동작을 취할 때 촬영한 영상(I_c)과 비교하여, 배경인 경우 실루엣 영상(S)의 해당 픽셀 값을 0으로 설정하고 전경 물체인 경우 1로 설정한다(식 (1)).

$$S(x,y)=\begin{cases}1 & \text{if}|I_c(x,y)-I_b(x,y)|>threshold\\ 0 & \text{otherwize}\end{cases} \quad (1)$$

실루엣 영상을 생성한 후, 실루엣 영상 내에서 배경과 전경 물체를 구분하는 경계선의 좌표를 계산한다. 계산

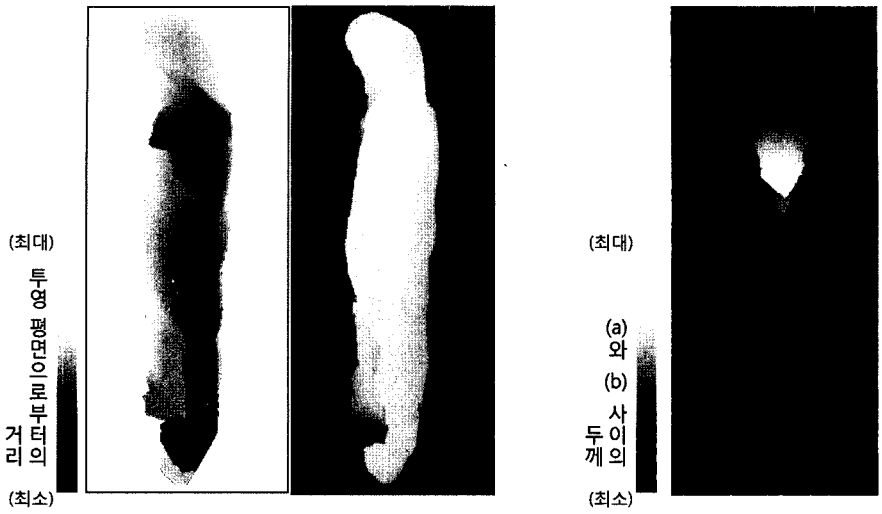


그림 3 프로젝션 맵: (a) 카메라 촬상면으로부터 가장 가까운 경계면까지의 거리, (b) 가장 먼 경계면까지의 거리, (c) (a)와 (b)사이의 거리(두께)

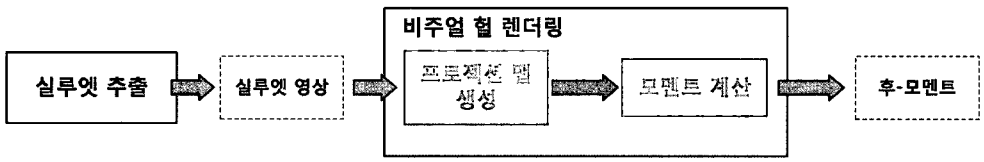


그림 4 특징값 추출 과정

된 실루엣의 경계선 좌표는 카메라 캘리브레이션 데이타를 이용하여 3차원 공간으로 역투영(back projection)함으로써 실루엣 콘을 생성하기 위해 사용된다.

3.2 비주얼 헐 렌더링

3.2.1 프로젝션 맵 생성

여러 카메라의 시점으로부터 생성된 실루엣 콘들의 교차 영역이 비주얼 헐이다. HAVH 알고리즘을 이용할 경우, 실루엣 콘에 0과 1로 구성된 실루엣 영상들을 투영시킴으로써 교차 영역의 기하정보를 계산하지 않고 비주얼 헐을 렌더링할 수 있다. 그림 5(a)에서 보는 것과 같이, n번째 카메라(C_n)에 대한 실루엣 콘을 렌더링할 때, C_n에 해당하는 실루엣 영상을 제외한 다른 모든 실루엣 영상(S_1, S_2,…,S_{n-1})를 실루엣 콘에 투영한다. 투영된 실루엣 영상들의 알파 값을 모두 곱하여 그 결과 값이 1인 영역만 렌더링함으로써 C_n에 대한 실루엣 콘 중에서 비주얼 헐 영역에 속한 부분만 그려지게 된다

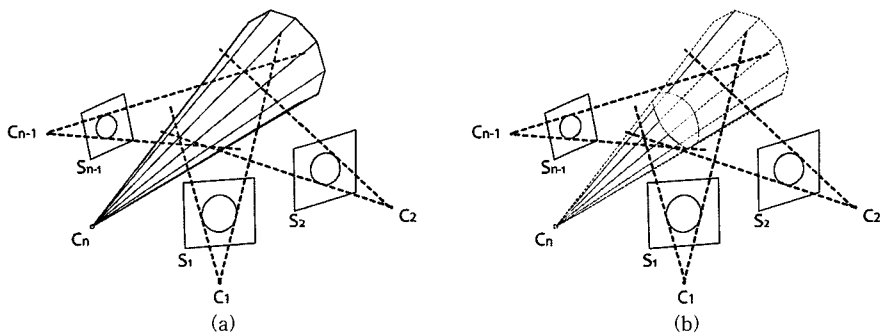


그림 5 실루엣 콘 렌더링

(그림 5(b)). C_1부터 C_n까지 n개의 카메라에 대한 각각의 실루엣 콘에 동일한 과정을 수행함으로써 임의의 시점에 대한 비주얼 헐 영상을 생성할 수 있다.

일반적으로 프로젝션 맵이란 3차원 공간 안에 분포하는 모든 점을 임의의 평면에 투영시키고 이를 누적하여 얻어지는 2차원 영상이다. 본 논문에서 우리는 임의의 카메라 시점에 대한 이미지 평면(프로젝션 평면)을 구성하는 각각의 점에서 수직선을 그려 3차원 물체와 만나는 가장 가까운 점까지의 거리와 가장 먼 점까지의 거리, 두 점 사이의 거리(3차원 물체의 두께)를 프로젝션 평면에 저장하고, 이를 그레이 스케일로 표현한 영상을 프로젝션 맵이라 정의한다.

물체의 가장 가까운 경계면과 가장 먼 경계면 프로젝션 맵은 프로젝션 평면으로부터 물체 앞면을 구성하는 픽셀까지의 거리와 물체 뒷면을 구성하는 픽셀까지의 거리를 구함으로써 얻을 수 있다. 또한 물체의 두께 정보를 포함하는 프로젝션 맵(두께 맵)의 경우, 앞면을 이루는 픽셀까지의 거리와 뒷면을 이루는 픽셀까지의 거리의 차를 계산함으로써 구할 수 있다. 이와 같은 프로젝션 맵 계산은 아래의 식으로 표현할 수 있다.

$$P_f[i,j]=\begin{cases}\min(z_p) & if\ \ \vec{v}[i,j,z_v]\cdot\vec{p}[i,j,z_p]=-|\vec{v}||\vec{p}| \\ z_p \le z_v & \\ 0 & otherwise\end{cases} \quad (2)$$

$$P_r[i,j]=\begin{cases}\max(z_p) & if\ \ \vec{v}[i,j,z_v]\cdot\vec{p}[i,j,z_p]=|\vec{v}||\vec{p}| \\ z_p \le z_v & \\ 0 & otherwise\end{cases} \quad (3)$$

$$P_d[i,j]=P_r[i,j]-P_f[i,j] \quad (4)$$

식에서 $P_f[i,j]$는 가장 가까운 경계면을, $P_r[i,j]$는 가장 먼 경계면을, $P_d[i,j]$는 두께를 나타낸다. $\vec{v}[i,j,z]$은 프로젝션 평면 상의 한 점(i, j)에서 수직인 벡터이고, $\vec{p}[i,j,z]$는 $\vec{v}[i,j,z]$에 대응하는 3차원 물체를 구성하는 한 점에 대한 노말(normal) 벡터이며, z_p는 $\vec{p}[i,j,z]$의 z 성분의 크기를 의미한다.

그림 6과 같이 기준면에 수직한 벡터 $\vec{v}$가 주어졌을 때, $\vec{v}$와 물체의 교차점 P를 찾을 수 있다. $\vec{v}$와 물체의 교차점 P_1은 물체의 앞면 상에 존재하는 점이고, 교차점 P_2는 물체의 뒷면 상에 존재하는 점이다. 교차점 P_1과 P_2의 차를 이용하여 두 점 사이의 거리를 구할 수 있으며, 이는 물체의 두께를 의미한다.

프로젝션 맵은 특징점을 추출하고자 하는 물체를 렌더링함으로써 획득할 수 있다. 임의의 시점에서 보이는 물체를 렌더링하기 위하여 시점의 위치와 방향 등의 정보가 주어진다. 주어진 시점 정보에 의하여 프로젝션 평면이 결정되며, 이는 프로젝션 맵을 생성하는 기준면이 된다. 렌더링되는 물체는 프로젝션 평면 위로 투영되어 그려진다. 이때, 원근 투영으로 인한 왜곡을 없애기 위하여 직교 투영을 이용하도록 설정한다. 이로 인하여 프로젝션 평면 상의 모든 픽셀은 자신이 속한 평면에 수직한 시선의 방향 $\vec{v}$를 갖게 된다. GPU 내부의 렌더링 파이프라인 중 래스터라이징 단계에서는 렌더링되는 물체의 기하 정보가 픽셀화되며, 이 과정에서 생성된 각 픽셀마다 시선의 방향 $\vec{v}$와 만나는 물체의 교차점 P가 결정된다. 또한, 기준면으로부터의 깊이(z) 정보는 렌더링 파이프라인 중 정점 쉐이더 내부의 시점 변환 단계를 거쳐 래스터라이징 단계로 넘어갈 때 깊이 버퍼에 저장된다[19]. 이 값을 텍스처에 저장한 것이 가장 가까운 경계면 프로젝션 맵이다. 가장 먼 경계면 프로젝션 맵은 깊이 테스트의 설정을 반대(GL_LESS 대신 GL_GREATER를 사용)로 하여 물체를 다시 렌더링하여 생성된다. 이렇게 생성된 두 프로젝션 맵의 차를 이용하여 물체의 가장 가까운 경계면으로부터 가장 먼 경계면까지의 거리를 저장한 두께 맵을 구할 수 있다.

가장 가까운 경계면과 가장 먼 경계면 프로젝션 맵의 획득은 렌더링시 하드웨어에 의하여 생성되는 깊이 버퍼의 값만을 필요로 하기 때문에 일반적인 렌더링에 사용되는 컬러 버퍼 갱신이나 텍스처 연산, 조명 계산 등을 하지 않음으로써 생성 시간을 더욱 단축시킬 수 있다. 획득한 두 프로젝션 맵의 차 연산을 통한 두께 맵의 생성 또한 다중 텍스처 합성(multi-texture blending) 함수(GL_SUBTRACT)를 이용하여 GPU에서 처리할 수 있다.

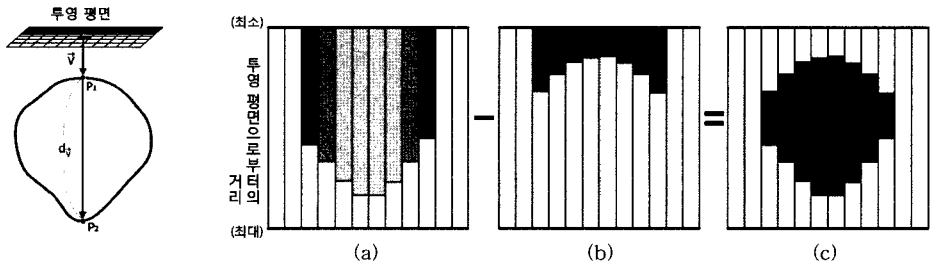


그림 6 기준면으로부터 물체 앞면까지의 거리(a)와 뒷면까지의 거리(b) 및 물체의 두께(c)

3.2.2 후-모멘트 계산

제안하는 방법에서 최종적으로 생성되는 특징값은 프로젝션 맵으로부터 추출된 후-모멘트[15]이다. 후-모멘트는 2차원 영상에서 계산되는 모멘트를 조합하여 생성되는 7차원의 데이타로, 영상의 회전과 이동, 크기 변환의 영향을 적게 받는다는 특성을 가진다. 후-모멘트의 계산을 위해서는 다음의 과정을 거치게 되는데, 각 과정은 기본적인 모멘트가 위치, 크기, 회전 변환에 강인한 특성을 가지도록 변환시켜주는 역할을 한다. 먼저, 식 (5)와 같이 정의 되는 기본적인 2차원 영상의 모멘트(M_{pq})를 계산한다.

$$M_{pq} = \sum_x \sum_y x^p y^q I(x,y) \tag{5}$$

위 식에서 p, q는 x, y 각 성분에 대한 모멘트의 차원을 의미하고, $I(x, y)$는 해당 픽셀의 밝기를 의미한다. 다음은 계산된 모멘트를 중심 모멘트(central moment)로 변환하는 식이다(식 (6)).

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q I(x,y), \tag{6}$$
$$\bar{x} = \frac{M_{10}}{M_{00}}, \quad \bar{y} = \frac{M_{01}}{M_{00}}$$

또한, 중심 모멘트의 계산을 간략히 하기 위해 $0 \le p \le 3$, $0 \le q \le 3$의 범위에 대해 다시 정리를 하면 다음과 같이 정리할 수 있다.

$$\mu_{00} = M_{00} = \mu,$$
$$\mu_{10} = \mu_{01} = 0,$$
$$\mu_{20} = M_{20} - \mu\bar{x}^2,$$
$$\mu_{11} = M_{11} - \mu\bar{x}\bar{y},$$
$$\mu_{02} = M_{02} - \mu\bar{y}^2,$$
$$\mu_{30} = M_{30} - 3M_{20}\bar{x} + 2\mu\bar{x}^3,$$
$$\mu_{21} = M_{21} - M_{20}\bar{y} - 2M_{11}\bar{x} + 2\mu\bar{x}^2\bar{y},$$
$$\mu_{21} = M_{12} - M_{02}\bar{x} - 2M_{11}\bar{y} + 2\mu\bar{x}\bar{y}^2,$$
$$\mu_{03} = M_{30} - 3M_{02}\bar{y} + 2\mu\bar{y}^3 \tag{7}$$

최종적으로 위치, 크기, 회전 변환에 강인한 특성을 가지는 후-모멘트는 다음의 식 (8)을 이용하여 계산된다.

$$h_1 = \eta_{20} + \eta_{02}$$
$$h_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$
$$h_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$
$$h_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$
$$h_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]$$
$$h_6 = (\eta_{20} - \eta_{02})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$
$$h_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \tag{8}$$

이 때, η_{pq}는 중심 모멘트가 크기 변환에 강인한 특성을 가지도록 정규화한 모멘트이다(식 (9)).

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\left(1 + \frac{p+q}{2}\right)}} \tag{9}$$

제안하는 방법은 프로젝션 맵이 생성된 후, GPU의 프레임 버퍼에 저장된 데이타를 이용하여 렌더링 파이프라인 상에서 후-모멘트를 계산한다. 모멘트 계산과 같이 영상을 구성하는 모든 픽셀을 동시에 이용하는 연산을 GPU에서 구현하기 위해 파이프라인 상의 정점 쉐이더를 이용한다. 정점 쉐이더는 행렬 연산을 이용한 좌표계 변환을 위한 하드웨어이기 때문에, 이를 이용하여 후-모멘트를 계산하기 위해서는 위의 수식을 행렬 연산으로 변환하여야 한다. 그러나, 모든 수식을 변환할 경우, 식의 개수만큼 렌더링 과정을 거쳐야 하기 때문에 CPU를 사용할 때와 비교하여 매우 비효율적이다. 그러나 위의 수식을 살펴보면, 실제로 영상을 구성하는 픽셀 값을 이용하는 식은 식 (5)이고, 그 이후의 모든 식은 이전 단계의 계산 결과를 이용함을 알 수 있다. 따라서 제안하는 방법에서는 기본 모멘트를 계산하는 식 (5)에 대해서만 정점 쉐이더를 이용하여 행렬 연산을 수행하고, 그 이후의 계산은 픽셀 쉐이더 상에서 병렬로 수행하도록 구현함으로써 한번의 렌더링 과정으로 후-모멘트의 계산이 가능하도록 한다.

4. 실험 및 결과

본 장에서는 실시간 특징값 추출의 결과를 보여준다. 모든 결과 영상은 2.13GHz CPU와 2Gbyte 메모리, 그리고 nVidia GeForce 8800GTX 그래픽 카드로 구성된 시스템에서 Direct3D와 고수준 쉐이더 언어(High-Level Shader Language: HLSL)을 이용하여 생성하였다. 입력 영상은 하나의 좌표계로 정확하게 캘리브레이션 되어있는 8개의 카메라를 사용하여 촬영되고, 각 카메라는 시스템 안의 물체(사람) 주변에 일정한 간격으로 배치되어있다. 또한, 입력 영상과 결과 영상의 해상도는 모두 640×480 픽셀이다. 실험은 특징값 추출 시간 비교와 추출된 특징값을 이용한 인식률 평가의 두 가지로 실시하였으며, 결과를 통해 제안된 방법이 3차원 특징값을 이용하는 실시간 동작 인식 환경에 적용할 수 있음을 확인하였다.

4.1 특징값 추출 시간 비교

표 2는 기존의 3차원 물체를 이용한 3차원 특징값 추출 방법과 제안하는 방법의 수행 시간을 비교한 결과이

다. 제안된 방법과의 수행 시간 비교를 위해 사용된 방법은 그래프 기반 특징값 추출 알고리즘인 세선화 기반 뼈대 추출 알고리즘과 히스토그램 기반 추출 알고리즘인 3차원 bin-distribution 생성 알고리즘이다. 실험에서 사용하는 인식 대상에 대한 3차원 모델은 GPU를 이용하여 300×300×300 크기의 복셀 공간 안에 생성된 비주얼 헐이다. 이 때, 비주얼 헐 생성에 사용되는 실루엣 영상을 계산하는 과정(1개 영상당 8ms)은 모든 알고리즘에서 공통적으로 필요한 과정이기 때문에 수행 시간 비교에서 제외하였다.

제안하는 방법은 3차원 물체 재구성 과정에서 임의의 카메라 시점에 해당하는 프로젝션 맵 생성과 후-모멘트 계산이 이루어지기 때문에 3차원 물체를 구성하는 모든 점을 미리 계산하고 특징값을 추출하는 기존 방법에 비해 수행 시간이 매우 빠르다. 표에서 보는 것처럼, 기존 방법의 경우 3차원 물체를 구성하는 모든 점을 재구성하기 위해서 370ms의 시간과 별도의 특징값 추출 시간이 필요한 반면, 제안하는 방법은 사용자가 정한 카메라 시점을 기준으로 두 개의 경계면 프로젝션 맵을 생성하고 둘 사이의 거리를 계산하는 과정으로 구성되는 프로젝션 맵 생성 단계에서 비주얼 헐을 두 번 렌더링하는 3ms 정도의 시간과, 프로젝션 맵을 이용하여 기본적인 모멘트를 계산하고, 계산된 결과를 이용하여 후-모멘트를 계산하는 단계에서 약 2ms의 시간을 필요로 한다. 따라서, 제안하는 방법을 동작 인식 시스템에 적용할 경우, 다시점 카메라 시스템을 통해 영상을 입력받고 실루엣을 추출하는 시간(8ms×8 = 64ms)을 포함하여 약 70ms의 시간이 소요되기 때문에, 1초에 14개의 입력 영상 집합에서 특징값을 추출하는 것이 가능하다.

4.2 후-모멘트를 이용한 인식 정확도 측정

그림 7은 카메라 입력 영상(그림 7(a))에서 전경 물체를 추출한 실루엣 영상(그림 7(b))과 재구성된 3차원 물체를 이용하여 생성된 프로젝션 맵(그림 7(c, d, e))을 보여준다. 실험에서 우리는 후-모멘트를 이용한 인식의 정확도 측정을 위해 사람의 8개 자세(양팔 벌리기, 양팔 위로 들기, 오른팔 위로 들기, 왼팔 위로 들기, 앞으로 나란히, 차려, 앉기, 허리 숙이기)를 사용하며, 각각의 자세마다 7명을 3회에 걸쳐 촬영하여 168개의 영상 집합에 대한 3종류의 프로젝션 맵을 생성하였다. 사람의 동작은 z=0인 평면에 한정되고, top-view 카메라 입력 영상은 회전, 이동 및 크기 변화에 영향을 적게 받기 때문에, 프로젝션 맵은 top-view 카메라를 기준으로 생성된다. 이렇게 생성된 프로젝션 맵에서 추출된 후-모멘트는 각각의 자세에 대해 계산한 후-모멘트 군집의 공분산과 평균을 이용하여 마할라노비스 거리(식 (10))를 계산하고 유사도를 측정한다.

$$D_M(x) = \sqrt{(x-\mu)^T \sum^{-1} (x-\mu)} \qquad (10)$$

위의 식에서 보는 것처럼, 마할라노비스 거리는 단순히 계산된 후-모멘트(x)와 각 자세에 대한 평균 후-모멘트(μ) 사이의 기하학적인 거리만을 계산하는 유클리드 거리와 달리, 데이타 간의 공분산(Σ)을 이용하기 때문에 본 논문에서 사용하는, 구성 데이타 간의 편차가 매우 큰, 후-모멘트 세트 간의 유사도를 보다 정확하게 계산할 수 있다. 그림 8은 8개 자세에 대해 생성된 3종류의 프로젝션 맵 각각에서 추출한 후-모멘트를 이용할 때의 인식률을 나타낸 혼동 행렬을 시각화한 유사도 맵이다. 맵을 구성하는 모든 값은 0과 255 사이의 값으로 정규화되어 표현되었으며, 255에 가까울수록 높은 유사도를 나타낸다. 그림 8에서 보는 것과 같이, 가장 가까운 경계면에 대한 프로젝션 맵(그림 8(a))과 두께 맵(그림 8(c))을 이용한 인식 결과가 가장 먼 경계면에 대한 프로젝션 맵(그림 8(b))을 이용한 결과와 비교하여 더 높은 유사도를 보임을 알 수 있는데, 이는 가장 먼 경계면 프로젝션 맵은 주로 하체에 대한 정보로 구성되어 있지만, 가장 가까운 경계면 프로젝션 맵과 두께 맵은 상체에 대한 정보를 포함하고 있어서, 주로 상체 부분에 대해서만 형태적 차이를 보이는 8개 자세에 대한 특징을 잘 표현할 수 있기 때문이다.

표 3은 기존의 3차원 오브젝트를 이용하여 특징값을 추출하는 방법과 제안하는 방법 중 가장 가까운 경계면 프로젝션 맵을 이용한 방법의 8개 동작 그룹에 대한 마할라노비스 거리를 비교한 것이다. 표에서 보는 것과 같이 제안하는 방법을 이용하여 추출된 각각의 동작에 대한 특징값 사이의 거리는 기존의 3차원 오브젝트를 이용하여 추출한 특징값 사이의 거리에 비해 그 차이가 적게 나타나지만 각각의 동작을 구분하기에는 충분한

표 2 제안한 방법과 기존의 3차원 특징값을 추출하는 방법의 각 세부 단계별 수행 시간 비교

특징값 추출 방법	3차원 물체 재구성 시간 (비주얼 헐 생성)	특징값 추출 시간	전체
세선화 기반 Skeleton 추출	370	10^7	$10^7 + 370$
3차원 bin-distribution 생성		10	380
제안된 방법	특징값 추출 시간		전체
	프로젝션 맵 생성 시간	후-모멘트 계산 시간	
	3	2	5

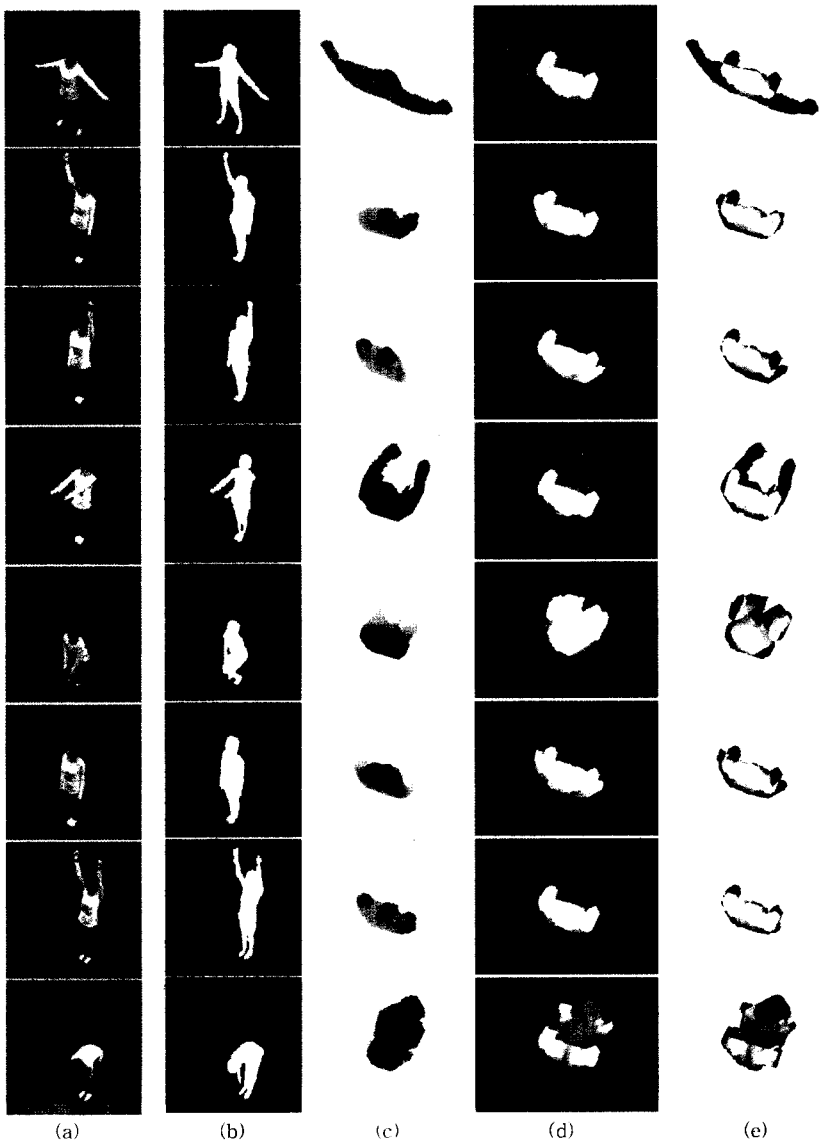


그림 7 8개 자세에 대한 입력 영상에서 추출된 특징값: (a) 카메라 입력 영상, (b) 실루엣 영상, (c) 가장 가까운 경계면 프로젝션 맵, (d) 가장 먼 경계면 프로젝션 맵, (e) 두께 맵

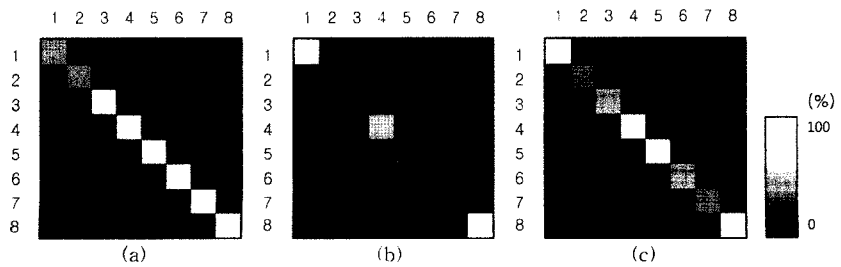


그림 8 후-모멘트를 이용한 유사도 맵: (a) 가장 가까운 경계면 프로젝션 맵을 이용한 결과, (b) 가장 먼 경계면 프로젝션 맵을 이용한 결과, (c) 두께 맵을 이용한 결과

표 3 기존 방법과 제안한 방법을 이용하여 추출된 특징값의 8개 동작에 대한 마할라노비스 거리 비교 결과

특징값 추출방법	저장된 동작 / 입력된 동작	1	2	3	4	5	6	7	8
3차원 Bin-distribution	1	**0.8**	1	0.96	0.96	0.95	0.97	0.99	0.95
	2	0.99	**0.23**	0.62	0.96	1	0.97	0.72	0.98
	3	0.76	1	**0.07**	0.76	0.77	0.76	0.69	0.76
	4	0.69	0.39	1	**0.35**	0.81	0.75	0.69	0.65
	5	0.65	1	0.54	0.65	**0.19**	0.65	0.51	0.65
	6	0.72	0.51	0.84	0.71	0.64	**0.26**	1	0.73
	7	0.73	1	0.63	0.72	0.73	0.72	**0.11**	0.73
	8	0.84	0.58	1	0.85	0.88	0.86	0.92	**0.74**
제안하는 방법	1	**0.15**	1	0.45	0.26	0.18	0.4	0.23	0.26
	2	1	**0.0002**	0.001	0.02	0.001	0.03	0.001	0.04
	3	1	0.002	**0.0006**	0.02	0.002	0.04	0.001	0.04
	4	1	0.01	0.01	**0.003**	0.015	0.02	0.01	0.02
	5	1	0.01	0.004	0.02	**0.001**	0.04	0.003	0.04
	6	1	0.08	0.04	0.02	0.02	**0.004**	0.02	0.01
	7	1	0.01	0.002	0.02	0.003	0.03	**0.0006**	0.03
	8	1	0.05	0.03	0.02	0.03	0.01	0.02	**0.004**

차이를 보여줌을 확인할 수 있다. 따라서 본 논문에서 제안하는 GPU 기반의 프로젝션 맵을 이용하여 추출된 특징값은 동작 인식을 위한 시스템에 적용할 경우 기존의 3차원 오브젝트를 사용하는 방법만큼의 정확도를 제공하는 것이 가능하다고 할 수 있다.

5. 결 론

본 논문에서 우리는 3차원 물체 재구성 과정과 특징값 추출 과정을 통합하여 실시간으로 3차원 정보를 가지는 특징값의 추출이 가능한 방법을 제안하였다. 제안된 방법은 3차원 물체 전체를 재구성하는 대신, HAVH 렌더링 기법을 이용하여 임의의 시점에 대한 3차원 정보만을 재구성함으로써 3차원 모델에 대한 3종류(가장 가까운 경계면, 가장 먼 경계면, 두께)의 프로젝션 맵을 생성하고, 이를 이용하여 최종적으로 후-모멘트를 추출한다. 제안하는 방법은 GPU 내부의 렌더링 파이프라인 단계에서 특징값 추출에 필요한 데이타를 계산하기 때문에, 입력 영상 내의 물체에 대한 3차원 정보를 빠르게 계산할 수 있고, 따라서, 실시간 인식 시스템에 적용이 가능하다. 그러나 제안된 방법은 현재 GPU 하드웨어의 제약 때문에 16개 이상의 카메라를 동시에 사용할 수 없고, CPU와 GPU를 동시에 사용할 경우 데이타 전송에 지연이 발생한다. 그리고, 인식 대상에 대한 3차원 정보로 카메라 촬상면에서 가장 가까운 경계면까지의 거리와 가장 먼 경계면까지의 거리, 두 맵 사이의 거리를 사용하기 때문에, 동작 중에 신체 일부가 겹쳐지는 경우 영역 사이의 거리를 계산하지 못한다. 또한, 특징값 추출을 위해서 렌더링 과정을 여러 번 거쳐야 한다는 문제가 있다. 따라서, 현재 우리는 제안한 방법의 성능을 향상시키기 위해, 후-모멘트 계산의 병렬화 방법과 메모리 전송 시간 감소 및 정확도 향상을 위한 연구를 진행하고 있다.

참 고 문 헌

[1] D. Ballard, C. Brown, "Computer Vision," Prentice-Hall, pp. 24-30, 1982.

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

[3] P. O. Hoyer, A. Hyvarinen, "Feature Extraction from Color and Stereo Images using ICA," The International Joint Conference on Neural Networks, Vol. 3, pp. 369-374, 2000.

[4] D. Torkar, N. Pavesic, "Feature Extraction from Aerial Images and Structural Stereo Matching," The 13th International Conference on Pattern Recognition, Vol. 3, pp. 880-884, 1996.

[5] C.M. Cyr, B.B. Kimia, "3D Object Recognition Using Shape Similarity-based Aspect Graph," The 8th IEEE International Conference on Computer Vision, Vol. 1, pp. 254-261, 2001.

[6] C. Chu, I. Cohen, "Posture and Gesture Recognition using 3D Body Shapes Decomposition," The IEEE Computer Society Conference on CVPR 2005, Vol. 3, pp. 69, 2005.

[7] D. Kyoung, Y. Lee, W. Baek, E. Han, J. Yang, K. Jung, "Efficient 3D Voxel Reconstruction using Pre-computing Method for Gesture Recognition," Korea-Japan Joint Workshop 2006, pp. 67-73, 2006.

[8] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A.

Halderman, D. Dobkin, D. Jacobs, "A Search Engine for 3D Models," ACM Transactions on Graphics, Vol. 22, pp. 83-105, 2003.
[9] M. Hilaga, Y. Shinagawa, T. Kohmura, T. Kunii, "Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes," The 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 203-212, 2001.
[10] H. Sundar, D. Silver, N. Gagvani, S. Dickinson, "Skeleton based Shape Matching and Retrieval," International Conference on Shape Modeling International, pp. 130-139, 2003.
[11] N.D. Cornea, D. Silver, P. Min, "Curve-Skeleton Properties, Applications and Algorithms," IEEE Transactions on Visualization and Computer Graphics, Vol. 13, pp. 530-548, 2007.
[12] N.D. Cornea, D. Silver, X. Yuan, R. Balasubramanian, "Computing Hierarchical Curve-Skeletons of 3D Objects," The Visual Computer, Vol. 21, pp. 945-955, 2005.
[13] A. Brennecke, T. Isenberg, "3D Shape Matching using Skeleton Graphs," Simulation and Visualization 2004, Vol. 13, pp. 299-310, 2004.
[14] M. Li, M. Magnor, H. Seidel, "Hardware-Accelerated Visual Hull Reconstruction and Rendering," Graphics Interface 2003, pp. 65-71, 2003.
[15] M. Hu, "Visual Pattern Recognition by Moment Invariants," IRE Transaction on Information Theory, Vol. 8, No. 2, pp. 179-187, 1962.
[16] A. Laurentini, "The Visual Hull Concept for Silhouette-based Image Understanding," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, pp. 150-162, 1994.
[17] R. Szeliski, "Rapid Octree Construction from Image Sequences," CVGIP: Image Understanding, Vol. 58, pp. 23-32, 1993.
[18] W. Matusik, C. Buehler, L. McMillan, "Polyhedral Visual Hulls for Real-time Rendering," The 12th Eurographics Workshop on Rendering Technique, pp. 115-126, 2001.
[19] C. Everitt, A. Rege, C. Cebenoyan, "Hardware Shadow Mapping," Technical Report, NVIDIA, 2002.

홍 광 진

2004년 2월 숭실대학교 컴퓨터학과 공학사. 2006년 2월 숭실대학교 미디어학과 공학석사. 2006년 3월~현재 숭실대학교 미디어학과 박사과정. 관심분야는 HCI, 증강현실, 카메라 기반 3차원 물체 모델링

이 철 한

2007년 2월 숭실대학교 미디어학부 공학사. 2007년 3월~현재 숭실대학교 미디어학과 석사과정. 관심분야 실시간 영상기반 모델링 및 렌더링

정 기 철

2000년 2월 경북대학교 컴퓨터공학과 공학박사. 2003년 3월 숭실대학교 미디어학과 전임강사. 2005년 3월~현재 숭실대학교 미디어학과 조교수. 관심분야는 HCI, 컴퓨터비전, 인공지능

오 경 수

2001년 2월 서울대학교 전기 컴퓨터공학부 공학박사. 2001년~2002년 (주)조이멘트 개발팀장. 2003년 3월~현재 숭실대학교 미디어학과 조교수. 관심분야는 Real-time Rendering, Computer Game