

논문 2007-44IE-4-8

반사 대칭을 이용한 3차원 오브젝트 검색에 관한 연구

(A Study on 3D Object Retrieval using Reflective Symmetry)

송 주 환*, 최 성 희**, 권 오 봉***

(Ju-Whan Song, Seong-Hee Choi, and Ou-Bong Gwun)

요 약

멀티미디어 활용이 다양해지면서 3차원 오브젝트 데이터 처리(저장, 편집, 검색 등)에 대한 관심이 높아지고 있다. 본 논문은 대칭면을 활용한 3차원 오브젝트의 검색 시스템을 제안·구현하였다. 검색기법으로는 투영선을 이용한 반사대칭축을 이용하였다. 반사대칭에 의해 대칭면을 구하고, 대칭면에 대한 깊이버퍼를 구하였으며, 깊이버퍼에 푸리에 변환을 적용하여 오브젝트에 대한 특징벡터를 생성하여 검색하였다. 질의에 의한 샘플은 콘스탄츠 대학(Konstantz University)의 벤치마크 데이터를 이용하여 유사 대칭성의 모델을 추출하였다. 대부분의 물체들이 대칭의 특성을 갖는 오브젝트임을 고려해 볼 때 본 논문이 제안한 반사대칭면을 이용한 3차원 오브젝트 검색에 의한 방법은 우수한 검색 시스템임을 알 수 있었다.

Abstract

Due to the diverse utilization of multimedia, interests are increasing towards processing (saving, editing, searching, etc) three dimensional object data. This paper proposes and realizes a retrieval system for three dimensional objects using reflective symmetry. For the retrieval method, a reflective symmetric axis with a projector is used. The symmetric plane is calculated by the reflection symmetry, and the depth buffer is calculated for the symmetric plane. Then, by applying the Fourier Transform to the depth buffer, the feature vector for the object is generated and retrieved. For the sample of inquiry, the model of similar symmetry was extracted using the benchmark data from Konstantz University. Considering that most of the objects have symmetrical characteristics, the proposed method of retrieving three dimensional objects using a reflective symmetric surface is an outstanding retrieval system.

Keywords : 반사대칭면, 3차원오브젝트 검색, 대칭축, 특징 추출

I. 서 론

디지털의 발달로 의료, 쇼핑, 오락 등 여러 분야에서 3차원의 영상 활용이 일반화되고 있다. 3차원 영상 사용이 범용화 되면서 다양한 검색기법에 대한 연구가 이루어지고 있다. 3차원 검색시스템은 정규화(normalization), 특징 추출(feature extraction), 유사성 검색

(similarity search)등으로 이루어진다. 특히 특징 추출은 3차원 오브젝트에서 특징 벡터를 구하는 과정으로 이동, 회전, 확대, 반사등의 변환 등에서 일정해야 한다^[1]. 이때 특징 벡터를 추출하는 방법은 다양한 기법이 있으나 이 기법 중 하나가 대칭축의 방법이다.

대칭은 물체 고유의 속성으로 우리에게 안정과 조화로움을 주며, 종류로는 반사 대칭과 회전대칭, 축대칭이 있다.

이러한 대칭 중 본 논문에서는 반사 대칭축을 이용한 특징 벡터 방법을 제안하였다. 반사대칭은 오브젝트 모델링의 경우 반쪽을 모델링하여 반사대칭하면 둘이 합성되어 하나의 완성된 모델을 얻는 간단하고 실제적인 특성을 갖는다. 반사대칭은 절대 대칭성과 상대 대칭성을 판단하는 2가지 방법이 있다. 절대적인 대칭성은 물

* 정회원, 전주대학교 교양학부
(School of Liberal Arts, Jeonju University)

** 정회원, 전주비전대학 유아교육과
(Department of Early Childhood Education, Vision College of Jeonju)

*** 정회원, 전북대학교 전자정보공학부
(Division of Electronics & Information Engineering, Chonbuk National University)

접수일자: 2007년7월27일, 수정완료일: 2007년12월1일

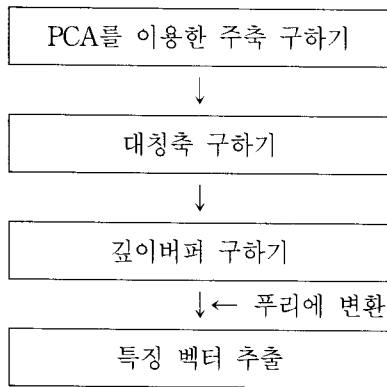


그림 1. 특징벡터추출 흐름도
Fig. 1. Pipeline of feature vector extraction.

체가 대칭여부를 판단하고, 상대적인 대칭성은 물체가 몇 %의 대칭성을 갖고 있느냐하는 유사대칭성을 의미한다. 절대 대칭성의 연구에서 Wolte^[2]는 포인트, 다각형, 다면체로 이루어진 물체에 대하여 회전대칭을 구하는 알고리즘을, Jiang^[3]은 다면체를 그래프로 변환하여 그래프 아이소모피즘을 이용한 회전대칭을 구하는 알고리즘을 연구하였다. 위 절대 대칭성의 알고리즘은 토폴로지의 변화에 매우 민감함을 알 수 있었다.

상대 대칭성 연구에서 Sun^[4-5]는 완전한 대칭성을 갖는 인위적인 와이어프레임 모델에 대한 알고리즘을, MinovicBreps^[6]은 표현된 3D 물체를 옥트리(octree) 구조로 변환하고 이를 추적하여 대칭축을 구하는 알고리즘을, Kazhdan^[7]은 구 함수의 집합을 이용하여 복셀 모델의 대칭축을 찾는 알고리즘을 연구하였다. 위 상대 대칭성 알고리즘은 유사 대칭성을 갖는 물체에 대해서 적용이 가능하고 잡음에 강한 특징을 갖고 있으나 계산량이 많은 단점이 있다.

이에 본 논문에서는 효율적인 3차원 오브젝트 검색을 위해 아래와 같이 가정한다.

- (1) 폴리곤 메쉬 모델이다.
- (2) 대칭성을 가지고 있는 물체이다.
- (3) 유사 대칭인 경우 한 부분만이 엇갈려 정의된다.
- (4) 비대칭성인 물체는 난수 또는 절차적 방법 등 특정한 규칙을 바탕으로 처리한다.

본 논문은 위와 같은 가정을 가지고 3차원 오브젝트를 검색하는 대칭축 알고리즘을 제안하였으며, 논문의 구성은 II. 대칭면의 깊이를 이용한 특징벡터 추출하기, III. 구현 및 평가, IV. 결론을 기술하였다.

II. 대칭면의 깊이를 이용한 특징 벡터 추출하기

<그림 1>에 대칭면에서 특징 벡터를 추출하는 흐름도를 나타내었다.

1. PCA를 이용한 주축 구하기

PCA를 이용한 폴리곤 메쉬 모델의 주축을 구하는 방법으로 순서는 다음과 같다.

- (1) 식 (1)을 이용하여 각 모델의 크기를 정규화 한다.
- (2) 물체의 중심(모든 정점들의 x, y, z 좌표를 평균)을 구하여 이를 물체좌표계의 원점으로 한다.
- (3) 식 (2)를 이용하여 공분산 배열(covariance matrix)을 구한다.

$$\begin{pmatrix} m_x \\ m_y \\ m_z \end{pmatrix} = \sum_{i=1}^N \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} / N \quad (1)$$

여기에서, (x_i, y_i, z_i) 는 물체 정점의 좌표, (m_x, m_y, m_z) 는 물체 중심 점의 좌표, N은 물체를 구성하는 정점의 수를 의미한다.

- (3) 식 (2)를 이용하여 공분산 배열(covariance matrix)을 구한다.

$$C = \frac{1}{N-1} \sum_{i=1}^N \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix} \quad (2)$$

여기에서,

$$cov(x,y) = \frac{1}{N-1} \sum_{i=1}^N (x_i - m_x)(y_i - m_y)$$

이고

$$cov(x,y) = cov(y,x)$$

이다.

- (4) 공분산 배열 C의 고유 값(eigenvalue)을 구한다.
- (5) (4)의 고유 값에 의해 고유벡터를 구한다.
- (6) 고유 값을 이용하여 물체의 주축을 구한다.

2. 대칭축 구하기

- (1) 투영선을 이용하여 반사대칭 지표구하기

물체는 완전 대칭성을 갖는 물체보다 유사 대칭성을 갖는 물체가 많다. 이러한 유사 대칭성을 갖는 오브젝트를 빠른 시간에 검색하기 위해서는 측정값이 필요하

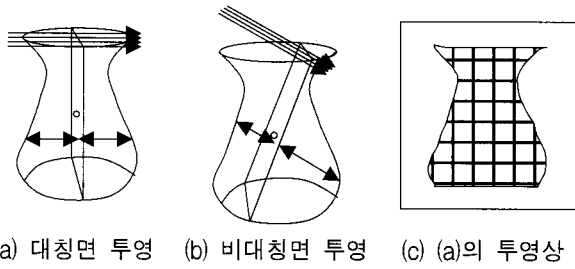


그림 2. 평행투영을 통한 반사대칭면 검출
Fig. 2. Detection of reflective symmetry plane using parallel projection.

며, 좌우가 다른 물체의 토폴로지 대칭성도 판별할 수 있어야 한다.

<그림 2>의 (a)와 (b)의 병 내부에 있는 판은 투영면이며 화살표는 투영선을 표시한다. 투영면 내부의 점은 물체 중심으로, 투영선이 물체에 들어가면 투영면을 통과하여 물체 밖으로 나온다. 이때 투영선의 입력 점에서 투영면까지의 거리를 e_{ij} , 투영면에서 출력 점까지의 거리를 o_{ij} 라고 하면, 투영면이 오브젝트의 대칭면일 경우에는 $e_{ij} = o_{ij}$ 이고, 그렇지 않은 경우에는 $e_{ij} \neq o_{ij}$ 가 된다. 이러한 성질을 이용하여 다음과 같이 물체의 대칭성을 판정하는 측정값 Sdpr(symmetry distance of projector ratio)을 정의하면 식 (3)과 같다^[8].

$$Sdpr(x,y,z,\Phi,\Theta) = 1 - \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n Sdpr(e_{ij} - o_{ij}) \quad (3)$$

$$Sdpr(e_{ij} - o_{ij}) = \frac{|e_{ij} - o_{ij}|}{\max(e_{ij} - o_{ij})}$$

식 (3)에서 x, y, z는 물체 중심점의 좌표를, Φ, Θ 는 투영면 y축을 기준으로 하는 물체 좌표계의 x, z축과의 각도를, mn는 투영상의 그림자를 지나는 투영선의 수를 표시한다. 이 대칭성 측정값은 0~1의 값이 되고 완전한 대칭일 경우 값은 1이 되며, 이 값은 물체의 대칭 정도를 판단하는 측정값으로 사용된다.

본 논문은 대칭면이 존재하는 오브젝트에 대한 검색 알고리즘으로 모든 오브젝트는 대칭면이 존재한다는 가

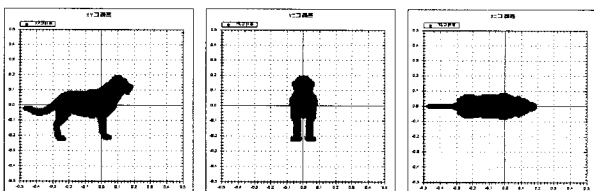


그림 3. 3주축 방향에서의 투영상
Fig. 3. Projection images perpendicular to 3 axes.

정 하에 검색하였다. <그림 3>은 3개의 주축 방향에서 오브젝트를 투영하여 나타난 영상이다. 3개의 영상중 하나는 반드시 그 오브젝트 대칭면에 근사(approximate)한 영상이 된다. 대칭축은 3개의 축 가운데 대칭면에 가장 근사한 축을 의미하며, 근사한 축에서 점차적으로 정확한 대칭축을 찾아가는 방법을 사용하였다.

(2) 반사대칭 측정값을 이용한 투영면 구하기

반사대칭은 물체 중심에서 Φ 와 Θ 를 변화시켜 모든 방향으로 향하는 대칭면을 구하고, 여기에 물체를 투영하면서 식 (3)에 의해 측정값을 구하여, 측정값이 제일 작은 값을 대칭면으로 취한다. <그림 4>는 구면을 동일면적의 구역으로 분할하고, 구역의 중심에 투영선을 입력하여 투영선에 수직인 투영면을 설치한다.

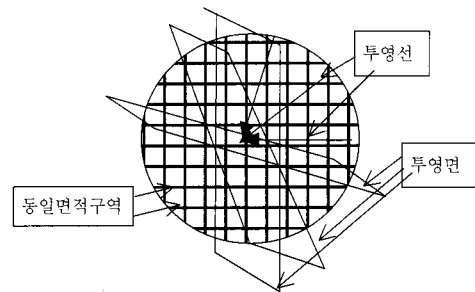


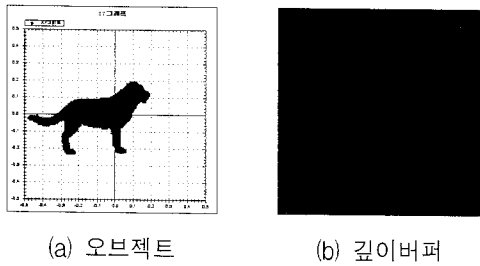
그림 4. 구면의 분할
Fig. 4. Subdivision the surface of a unit sphere.

투영면을 계산하는 과정은 다음과 같다.

- (1) <그림 4>와 같이 단위구의 구면을 같은 크기의 구역으로 분할한다.
- (2) 객체 중심에 투영선을 긋고 이것과 수직하게 투영면을 세운다.
- (3) 각 투영면에 객체를 평행 투영하여 식 (3)의 측정값을 구한다.
- (4) (3)에서 구한 값 중에서 제일 작은 측정값을 갖는 투영면을 대칭면으로 한다.

3. 깊이 버퍼 구하기

오브젝트를 대칭면의 수직인 방향에서 평행투영하면 <그림 5>와 같이 나타난다. 즉 투영면을 대칭면으로 정의 할 수 있다. 이때 투영광선이 오브젝트를 처음 만나는 z값을 깊이 버퍼에 저장할 경우 깊이 버퍼의 값을 오브젝트의 특징 벡터로 할 수 있다. 이때 투영면의 해상도와 깊이의 정도에 따라서 깊이 버퍼의 용량은 다르게 나타난다. 예를 들어 256x256해상도의 깊이 범위가



(a) 오브젝트 (b) 깊이버퍼

그림 5. 오브젝트에 대한 깊이버퍼
Fig. 5. Depth buffer of object.

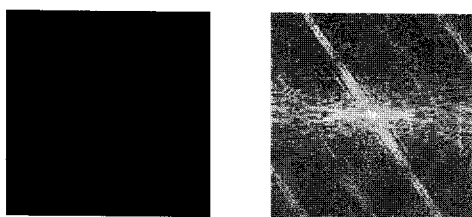
0~255이면 1byte의 깊이정보는 65,536(256×256)개의 메모리가 필요하게 된다.

4. 특징 벡터 추출하기

깊이 버퍼의 모든 값을 특징벡터로 사용하여 검색 할 경우, 전체 특징벡터를 비교해야하기 때문에 검색시간이 많이 요구되는 문제가 발생한다. 해결방안으로 깊이 버퍼의 값들을 2차원 이산 푸리에 변환을 사용하여, 변화가 많은 부분의 데이터를 한쪽으로 이동한 후, 앞쪽에 해당하는 일부 데이터만 오브젝트의 특징벡터로 사용한다^[9]. M×N 데이터에 대한 2차원 이산 푸리에 변환식은 다음과 같다.

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})} \quad (4)$$

<그림 6(a)>는 128×128 해상도의 깊이버퍼를 그림으로 표시했으며, <그림 6(b)>는 깊이버퍼에 의한 데이터를 푸리에 변환을 이용한 같은 해상도의 특징벡터이다. 특징벡터 추출은 깊이버퍼 데이터를 특징벡터로 전체 사용하기에는 데이터의 양이 대량이므로, 이를 단순화시키기 위하여 푸리에 변환을 이용하여 일부 데이터를 선택하였다. 푸리에 변환 후 벡터는 왼쪽 상단에 가장 큰 변화 값을 가지며, 오른쪽 하단으로 갈수록 변화가 적은 값이 나타났다. 이 벡터 중 본 논문은 검색을 위해



(a) 깊이버퍼 (b) 특징벡터

그림 6. 푸리에 변환을 통해 구한 특징벡터
Fig. 6. Extraction feature vector using 2D-FFT.

왼쪽 상단에서 x와 y축에서 각각 8개씩 총 64개 데이터를 선택하여 특징벡터로 추출하여 구현하였다.

III. 구현 및 평가

본 논문은 1GB의 메모리와 3.2GHz의 Pentium 4 CPU를 가진 개인용 컴퓨터에서 Microsoft사의 Visual C#.Net 2003으로 알고리즘을 구현하였다. 검색을 위한 특징벡터는 데이터베이스 Microsoft SQL Server 2000을 사용하였으며, 검색 데이터는 콘스탄츠(Konstantz) 대학의 벤치마크 데이터를 이용하여 3차원 오브젝트 검색 알고리즘을 평가하였다. 검색 데이터는 472개를 55개의 클래스 나누어 처리하였다^[10].

<그림 7>은 472개의 3차원 오브젝트 검색 데이터 중 하나로 04_dog에 대한 처리과정이다. <그림 7>과 같이 본 논문에서 제안한 방법은 3차원 오브젝트에 대해 반사대칭에 의해 대칭면을 구하고, 대칭면에 대한 깊이버퍼를 구한다. 깊이버퍼에 푸리에 변환을 적용하여 오브젝트에 대한 특징벡터를 생성한다. 이 특징벡터는 Microsoft SQL Server에 저장하여 검색 시 특징벡터의 키로 사용하였다.

표 1. 472개의 3차원 오브젝트
Table 1. 472 3D objects.

| 클래스 | 개수 | 클래스 | 개수 | 클래스 | 개수 |
|---------------|----|--------------|----|--------------------|----|
| 01_ant | 6 | 20_plane | 26 | 39_knife | 3 |
| 02_bunny | 4 | 21_plane | 10 | 40_screw | 3 |
| 03_cow | 7 | 22_plane | 4 | 41_spoon | 3 |
| 04_dog | 4 | 23_zeppelin | 6 | 42_table | 6 |
| 05_fish | 13 | 24_bike | 5 | 43_head | 3 |
| 06_bee | 5 | 25_car | 6 | 44_head | 8 |
| 07_chips | 4 | 26_car | 23 | 45_head | 4 |
| 08_keyboards | 8 | 27_formula | 9 | 46_book | 4 |
| 09_cans | 4 | 28_galeons | 4 | 47_clock | 2 |
| 10_bottles | 14 | 29_submarine | 5 | 48_clock | 4 |
| 11_bowl | 4 | 30_warship | 5 | 49_sword | 25 |
| 12_bucket | 4 | 31_bed | 7 | 50_barrel | 3 |
| 13_cup1 | 8 | 32_chair | 24 | 51_birch | 4 |
| 14_cup2 | 9 | 33_chair | 6 | 52_planted_flowers | 9 |
| 15_teapot | 4 | 34_chair | 4 | 53_trees | 11 |
| 16_biplane | 5 | 35_couch | 3 | 54_weed | 9 |
| 17_helicopter | 9 | 36_couch | 11 | 55_human | 56 |
| 18_missile | 16 | 37_axe | 4 | | |
| 19_plane | 18 | 38_glasses | 7 | | |

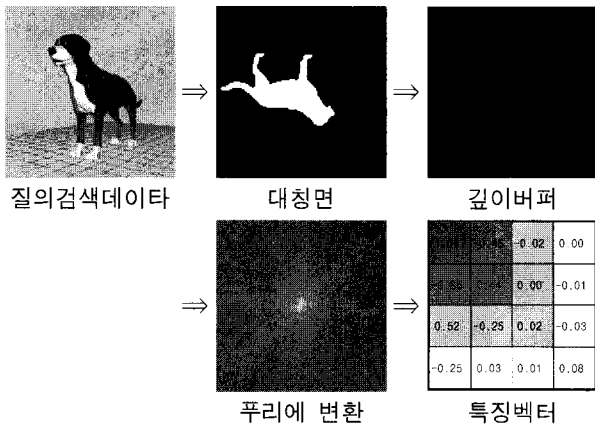


그림 7. 특징벡터 생성과정
Fig. 7. Procedure of feature vector extraction.

검색과정은 쿼리(query)로 선택된 오브젝트를 <그림 7>에 의해 특징벡터를 구하고, 특징벡터와 데이터베이스에 저장된 특징벡터의 차이의 제곱을 누적하여 더하였다. 여기에서 특징벡터는 대칭면의 앞과 뒤 두 방향 중 어느 방향에 대한 깊이를 적용하느냐에 따라 다르게 나타난다. 본 논문은 쿼리로 선택된 오브젝트에 대해서 대칭면의 앞과 뒤 두 방향에 대한 2개의 특징벡터를 구하고, 각각을 데이터베이스에 저장된 특징벡터와의 차이의 제곱의 합을 구하여, 작은 값을 특징벡터의 차이 값으로 정하였다.

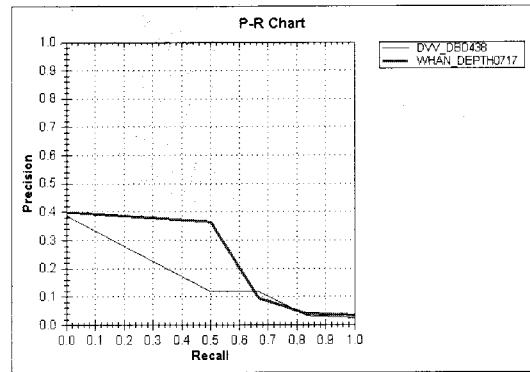
데이터베이스에 저장된 모든 오브젝트의 특징벡터는 이 과정을 반복하여 구하였으며, 구해진 값에서 가장 작은 오브젝트가 우리가 검색하려는 오브젝트로 결정하였다. 즉, 특징벡터 차의 제곱의 합이 작으면 작을수록 두 오브젝트의 모양은 유사하다고 판단하였다. 아래 식 (5)는 2개의 특징벡터의 차이의 합을 구하는 함수를 나타낸다.

$$M = \sum_{n=0}^{N-1} (FV_n - fv_n)^2 \quad (5)$$

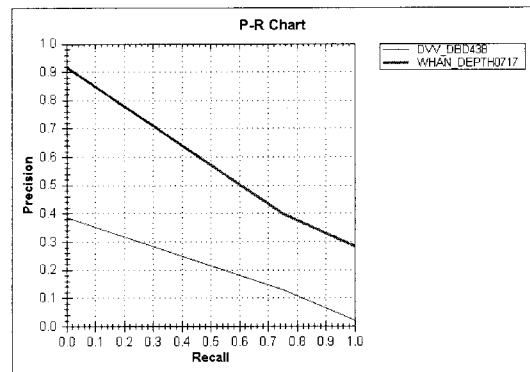
위 식에서 FV_n 은 검색하는 오브젝트의 특징벡터이고, fv_n 은 데이터베이스에 저장된 임의의 레코드의 특징벡터이며, N 은 특징벡터의 크기를 의미한다.

또한 제안한 알고리즘의 성능을 평가하기 위하여 Vranic이 연구한 깊이버퍼 기반 특징벡터(the depth-buffer based feature vector) 알고리즘과 비교하였다^[10]. Vranic은 오브젝트에 6면체의 바운딩박스(bounding box)를 씌우고, 6방향의 각 면에서 깊이버퍼를 이용하여 검색하였다.

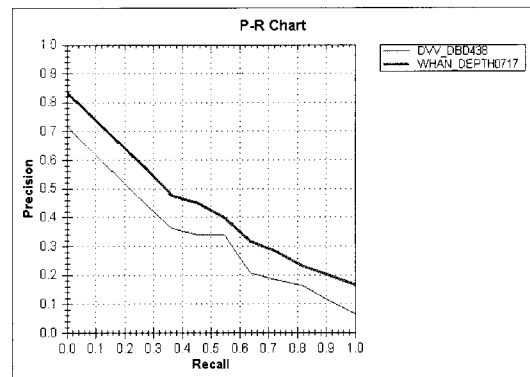
본 논문에서 제안한 대칭면에 대한 깊이 버퍼를 이용



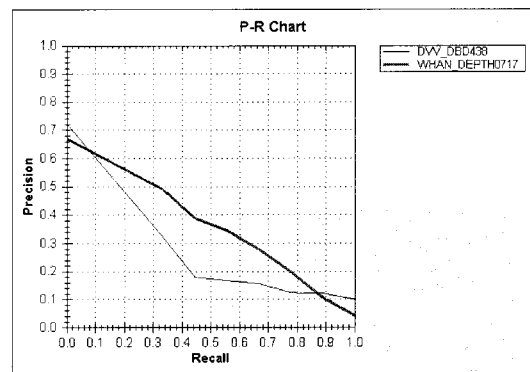
(a) 01_ant



(b) 09_can



(c) 36_couch



(c) 17_helicopter

그림 8. P-R charts
Fig. 8. P-R charts.



그림 9. 오브젝트 검색 예
Fig. 9. Example retrieval.

한 알고리즘과 Vranic의 알고리즘과 제안한 알고리즘결과를 아래의 precision-recall chart로 비교하였다.

<그림 8>은 472개의 데이터에서 완전 대칭면을 갖는 몇 개의 데이터의 성능을 비교하였다. 굵은 선으로 그려진 그래프는 본 논문에서 제안한 방법이며, 나머지는 Vranic이 제안한 방법이다. 완전대칭을 갖는 오브젝트에서는 본 논문에서 제안한 방법이 성능비교에서 우위를 알 수 있었다. <그림 8>은 오브젝트 0번 (k0000.off; ant; <그림 8(a)>)을 본 논문에서 제안한 방법으로 검색할 때 ant라고 판단되는 오브젝트들을 가장 높은 확률을 가진 것부터 상위 35개를 순서대로 나타내었다. <그림 9>는 쿼리 오브젝트에 의해 유사한 오브젝트를 검색하는 과정을 그림으로 나타내었다.

IV. 결 론

본 논문은 3차원 오브젝트의 검색에 대한 알고리즘을 제안하였다. 반사대칭에 의해 대칭면을 구하고, 대칭면에 대한 깊이버퍼를 구하였으며, 깊이버퍼에 푸리에 변환을 적용하여 오브젝트에 대한 특징벡터를 생성하여 검색하였다.

오브젝트의 대부분은 대칭성이 있는 물체로 본 논문에서 제안한 방법은 대칭면을 갖는 오브젝트를 검색하는 기법으로 precision과 recall 값이 높게 나타남을 구현을 통해 알 수 있었다.

또한 본 논문의 성능을 비교하기위해 Vranic이 연구한 깊이버퍼 알고리즘과 비교해 본 결과 대칭면이 존재

하는 3차원 오브젝트 데이터인 경우 본 논문의 결과가 Vranic의 알고리즘보다 검색 결과가 우수함을 알 수 있었다.

향후에는 대칭면을 찾는 시간을 단축하면서도 더 효율적인 알고리즘을 개발하는 연구를 할 계획이다.

참 고 문 헌

- [1] B. Bustos, D. A. Keim, D. Saupe, T. Schreck, and D. V. Vranic, "Feature-Based Similarity Search in 3D Object Databases," ACM Computing Surveys, Vol. 37, no. 4, pp. 345-387, December 2005.
- [2] J. D. Wolter, T. C. Woo, and R. A. Volz, "Optimal Algorithms for Symmetry Detection in Two and Three Dimensions," The Visual Computer, Vol. 1, no. 1, pp.37-48, 1985.
- [3] X. Y. Jiang and H. Bunke, "Determination of the Symmetries of Polyhedra and an Application to Object Recognition", Lecture Note in Computer Science, Vol. 553, pp. 113-121, 1991.
- [4] C. Sun and J. Sherrah, "3D Symmetry Detection Using the Extended Gaussian Image", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, no. 2, pp. 164-168, 1997.
- [5] C. Sun and D. Si, "Fast Reflective Symmetry Detection Using Orientation Histograms," Real-Time Imaging, Vol. 5, pp. 63 - 74, 1999.
- [6] P. Minovic, S. Ishikawa, S., and K. Kato, "Symmetry Identification of a 3-D Object Represented by Octree," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, no. 5, pp.507-514, 1993.
- [7] M. Kazhdan, B. Chazelle, D. Dobkin, A. Finkelstein, and T. Funkhouser, "A Reflective Symmetry Descriptor," Algorithmica, Vol. 38, no. 1, pp. 201-225, 2003.
- [8] J. Song and O. Gwun, "Reflective Symmetry Detection Based on Parallel Projeciton," Lecture Note in Computer Science, Vol. 4432, pp. 590-598, 2007.
- [9] D. Zhang and G. Lu, "A comparative study of curvature scale space and Fourier descriptors for shape-based image retrieval," Journal of Visual Communications and Image Representation, Vol. 14, no. 1, pp. 41-60, 2003.
- [10] D. V. Vranic, "3D Model Retrieval," Ph. D. Thesis, University of Leipzig, Germany, 2004.

저 자 소 개



송 주 환(정회원)

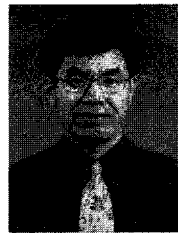
1995년 전주대학교 전자계산학과
학사 졸업.

1997년 전북대학교 전산통계학과
석사 졸업,

2003년 전북대학교 전산통계학과
박사 졸업,

2001년~현재 전주대학교 교양학부 교수

<주관심분야 : 3D컴퓨터그래픽스, 멀티미디어, 멀티
미디어 검색>



권 오 봉(정회원)

1980년 고려대학교 전기공학과
학사 졸업

1983년 고려대학교 전기공학과
석사 졸업

1993년 일본구주대학교 총합
이공학연구과 박사 졸업

1992년~1993년 일본구주대학교 정보공학과 조수

1993년~현재 전북대학교 전자정보공학부 교수

<주관심분야 : 컴퓨터그래픽스, 사이언티픽 비주
얼라이제이션>



최 성 희(정회원)

1982년 전북대학교 통계학과
학사 졸업.

1984년 중앙대학교 전자계산학과
석사 졸업,

2005년 전북대학교 전산통계학과
박사 졸업

1989년~현재 전주비전대학 유아교육과 교수

<주관심분야 : 컴퓨터그래픽스, 멀티미디어, 멀티
미디어 검색>