博 士 學 位 論 文

3 次元 形狀 檢索을 위한 不變 特徵벡터 抽出에 관한 硏究

A Study on Exterior Characteristic Vector Extractor for Retrieving 3-Dimensional Shape

2004년

漢 城 大 學 校 大 學 院

産 業 工 學 科

李 燦 鎬

博 士 學 位 論 文
指導敎授 洪 允 基

3 次元 形狀 檢索을 위한 不變 特徵벡터 抽出에 관한 硏究

A Study on Exterior Characteristic Vector Extractor for Retrieving 3-Dimensional Shape

위 論文을 工學 博士學位 論文으로 提出함

2003년 12월

漢城大學校 大學院

産 業 工 學 科

李 燦 鎬

3 次元 形狀 檢索을 위한
不變 特徵벡터 抽出에 관한 硏究

위 論文을 李燦鎬의 工學博士 學位論文으로 認定함

2003년 12월

審査 委員長 이재득

審 査 委 員 홍 윤 기

審 査 委 員 위 남 숙 印

審 査 委 員 徐 元 是

審 査 委 員 朴 건

A Study on Exterior Characteristic Vector Extractor for Retrieving 3-Dimensional Shape

by

Chan-Ho Lee

Department of Industrial Engineering
The Graduate School
Hansung University

ABSTRACT

Technology for audio and 2-dimensional image referencing has been developed in and out of the country, and is somewhat in the early stage of common usage. 3-D can be referred to as the basis of virtual reality, yet there is still much research to be done on 3-dimensional image referencing worldwide. 3-D configuration-matching technology is essential not only for restoration, but for effective future management of other 3-D digital functions.

The three dimensional exterior information extractor retrieves information on the shape of the surface of an object at a specific

point. The standard file format for 3-D is DFX. DFX uses a commercially-used data format, which is translated by the Characteristic Vector Extractor. After the information of the 3-dimensional shape of an object is translated, the Exterior Information Extractor provides details on the 3-dimensional form of the surface of any specific area.

Regardless of the 3-D object's size or direction, the Characteristic Vector Extractor extracts only the significant characteristic details the object's form is composed of, using a set value.

Out of a number of vectors, one particular vector is selected. All the vectors are compared to find the similarities between them. The Vector-Matching Engine then pinpoints the vector with the greatest resemblance to the previously selected vector.

By expressing 3-dimensional elements in a 2-dimensional form, management of data and various calculating is made easier. Out of each frequency, the elements other than the principal element that expresses the object were changed into wavelets, thus improving on current technology.

We tested the four different technologies, which we developed, to estimate the extraction duration time and the validity of typical image with 3-Dimensional object, and it was concluded that the method of 'fuzzy k-mean' is generally effective.

It could be used not only 3-dimensional matching engine but also multimedia contents-based indexing engine. So it is

necessary to drive the further development to make a commercialized product which would make it possible to contribute to the development of 3-D contents industry.

目 次

圖 目 次

表 目 次

제 1 장 서 론

1.1 연구의 배경 및 목적

정보의 종류가 문자뿐만 아니라 음성, 2차원 이미지, 3차원 컨텐츠 등으로 다양화 되고 그 정보량들이 급속히 폭증함으로써 그 들을 효율적으로 색인, 저장, 검색 처리할 장치 와 기술의 확보가 시급한 상황이다. 특히 디지털 컨텐츠, 디지탈 박물관, 가상현실, 게임, e-business의 정보에 대한 색인 및 검색기술에 연구 개발이 집중되고 있다.[Nadler1993]

일반적으로 이미지 파일내용검색은 text에 대한 전문검색 과는 달리 키워드 또는 주석을 이용한 색인 검색 기술을 이용해 왔는데 색인 관리자의 주관적, 기능적, 문화적 차이에 따라 동일 이미지에 다른 키워드로 처리한다면 다른 이용자는 이미지 컨텐츠가 있어도 검색이 불가능하다. 또한 키워드로 영상을 분석함에 있어서 그 영상이 확실히 구분 될 수 있는 내용으로 서술해야 한다는 것이다. 이러한 문제점은 관리자가 입력한 영상의 내용과 사용자가 관심을 가지고 찾고 싶어 하는 영상에 대해 일치점을 찾지 못한다면 사용자는 데이터베이스에서 자신이 관심을 가지고 있는 영상을 찾아내지 못하는 결과를 얻게 될 것이다.

사용자와 색인자가 같은 영상에 관해 다른 키워드를 입력한다면 사용자가 관심을 가지고 있는 영상을 찾을 수 없게 된다는 것이다. 또한 언어적 장벽으로 인한 문제점도 생각 할 수 있다. 현재 인터넷이 데이터를 국가간에 공유한다는 관점에서 특정언어로 색인된 데이터베이스는 제한적일 수 밖에 없다는 의미이다. 형태, 색상, 질감 등의 정보는 키워드나 주석으로는 다 색인 할 수 없다.[Kelly1995]

내용 기반 영상 검색을 위해서는 멀티미디어 데이터의 내용을 대표할

수 있는 특징을 추출해야 하며, 이를 기반으로 색인과 검색을 수행해야 하는데 일반적으로 사용되고 있는 영상의 주요 특징으로는 색상, 질감, 형태 및 영상을 구성하고 있는 객체들의 공간적 위치 등이 있다. 수많은 과거의 내용 기반 영상 검색 시스템은 영상의 이러한 특징 값들을 추출하는 방법에 그 초점을 맞추어 왔고, 또한 최근에는 멀티미디어 데이터에 대한 검색과 분석 방법론의 활발한 연구 활동과 더불어 응용 결과물이 실용화 단계로 발전되고 있다.[Berman1997][Faloutsos1995][Niblack1993]

영상에서의 형태는 영상을 표현하는 가장 중요한 특징 중의 하나이다. 흑백영상 또는 이진영상에서는 형태정보를 검색 시 중요하게 사용 될 수 있다. 더구나 색상이나 질감을 가지고는 제작된 많은 인공영상들의 특징을 표현할 수 없기 때문에 형태분석기법이 필요한 것이다.[Gonzales]

영상의 형태특징을 추출하는 많은 기법들은 대부분 영상 내에 존재하는 객체가 배경과 완전히 분리되었다는 가정 하에서 분석을 시도하기 때문에 복잡하고 여러 개의 객체가 혼재해 존재하는 자연영상에서는 기존의 방법을 사용할 수 없다. 따라서 본 연구에서는 사용되는 객체에 대해서 한정한다. 즉 이러한 객체들로서 데이터베이스를 구축하고, 검색을 시행한다. 그러므로 패턴 인식은 다양한 응용분야를 갖고 있지만, 시스템의 구성은 일반적으로 영상 획득 부분, 영상 처리 부분, 특징 추출 부분, 식별 부분으로 이루어지며 이들 각 분야의 연구는 그 속성상 상호 독립적으로 이루어 질 수 있다.[Gudivada1995] 이 중 특징 추출 부분은 변형에 대하여 안정된 물리량을 추출하는 것을 말하며 특징 추출의 목적은 정보의 손실 없이 패턴 벡터의 고유한 특징을 이용하여, 인식에 필요한 시간 및 기억 공간을 줄이는데 있다.

특징 추출은 궁극적으로 최종적인 인식 및 판단 처리를 보다 정확하고 용이하게 하기 위해 입력 영상을 가공하고 필요한 특징 정보를 추출하는 과정이다. 이러한 패턴의 특징 추출은 패턴 인식에 있어서 가장 어렵고 또

한 중요한 문제로 간주되고 있다. 현재까지 2차원 영상에서 패턴의 위치 이동, 크기 변화와 회전 변형에 무관한 특징 추출, 즉 패턴의 기하학적 변형에도 변하지 않는 특성을 유지하는 특징을 추출하기 위해서는, 전체적 특징에 의한 방법, 지역적 특징에 의한 방법, 변환식을 이용한 방법, 신경망에 의한 방법 등 많은 방법들과 기법들이 각각 개발되고 발전되어 왔다.[Weiyu2000][Ericsson1999]

이 논문은 제 2장 3D 특징 벡터 추출에서는 3D 모델 데이터의 형상이 가지고 있는 고유의 특징 정보를 추출하는 것으로 3D 모델의 일정한 방향을 잡기위한 기준 축 선정 및 좌표 변환, 3D 객체의 minimal outer surface 추출기술, Minimal outer surface에 대한 극 좌표계 기반의 distance 추출을 통한 3D 형상 정보의 2차원 정보화를 통해서 Distance 정보에 대한 불변 특징 벡터 추출한다. 또한 제 3장 에서는 형태 서술자의 하나인 Zernike 모멘트[Canterakis1999][Kim1994]를 사용하여 2차원 특징벡터를 추출하고 클러스터링하여 3차원 불변 벡터를 추출하는 방법을 제시하였다. [Chen1999][Paquet2000][Looney2002][Pal1995]

제 4장은 3차원 객체의 형상 정보를 바탕으로 모양, 문양, 패턴 등을 대표하는 형상 특징 정보를 효율적으로 추출하는 3차원 형상 특징벡터 추출 기술과 추출된 두 개 이상의 3차원 특징 벡터간에 가장 유사한 쌍을 찾아주는 벡터 매칭 기술에 대한 성능 분석과 위 기술을 기반으로 3차원 디지털 컨텐츠 원형이 저장된 DB에서 찾고자하는 3차원 또는 2차원 형상에 가장 적합한 유사 원형을 찾아주는 매칭 엔진과 이를 이용하는 데 필요한 API를 개발하였고 프로세스의 설계 및 응용시스템을 개발을 통해서 3차원 형상 검색 시스템을 구현하였다. 마지막 장은 결론과 향후과제를 제시하였다.[Weber1998][Faloutsos1995][Smith1996]

1.2 관련 연구

1.2.1 세계적 기술의 현황

원형복원을 위한 매칭 기술은 미국의 공룡형상 복원을 위한 시스템에서 발굴된 공룡화석 조각들을 DB화하여 이들을 토대로 새로이 발굴된 화석조각을 복원하는 데 사용하고 있다. 이 시스템의 경우 공룡화석에 대한 지식이 내재되어 화석조각의 3차원 정보를 바탕으로 기존의 DB로 구축된 화석 조각의 형태로부터 유사한 원형을 유추하여 복원 작업자의 복원 작업을 돕고 있다.

패턴 매칭에 대한 연구는 최근 대용량 멀티미디어 데이터 처리능력의 발달로 관련 기술의 구현 시도가 활발해지기 시작했다. 그 결과 이 연구의 적용분야는 인공 지능의 일환으로서 여러 분야에 적용이 시도되고 있다. 그 예로서는 문자 인식, 서명 인식, 지문 인식, 비행체 인식, 기계 공구 인식, 얼굴 인식 등 다양한 분야에 대한 연구가 진행 되고 있다.[Sim2000][Hirata1992]

3차원 형상에 관한 image mining에 관한 연구나 기술은 아직 태동 단계이다. 다만, 멀티미디어에 대한 연구는 기초연구단계에 있어서 미주의 몇 몇 대학에 의해서 발표된 논문이 약간 있으며 이는 Concordia University(캐나다)에서 발표된 Multiple query points parallel search algorithm(Comb algorithm) for multimedia database systems.처럼 알고리즘관련분야나 또는 The George Washington University(미국)에서 발표된 Indexing and Searching Schemes for Audio Data in Audio/Multimedia Databases처럼 멀티미디어 초기단계인 오디오 분야에 국한되고 있다. 그러나 2차원 영상에 관한 내용 그대로를 사용해 검색하는 내용기반 검색기술은 현재 기초연구단계를 지나 상용화 단계로 접어들어 다음과 같은 제품들이 나와 있다.

- IBM(미국) : Almaden연구소에서 개발한 정지 및 동영상검색시스템

- VIRAGE(미국) : Virage사에서 개발한 멀티미디어 검색시스템

- Excalibur(미국) : Excalibur사에서 개발한 멀티미디어 검색시스템

그리고 이외에 학계를 중심으로 Columbia 대학의 Advaent, Chicago대학의 WebSeer, MIT의 Photobook등이 있다. 그러나 사용자들의 의견은 아직까지 검색결과와 원하는 내용에 대한 적합도가 낮다는 문제점과 3차원 콘텐츠에 대한 검색이 지원되지 않고 있다는 점이 지적되고 있다.

1.2.2 국내 기술의 현황

벡터 매칭기술에 대한 기술은 크게 이미지 대 이미지(2D image), query type에 의한 이미지 검색이 현재 구현되어 있으며, 이미지 검색방식에는 이미지의 color, texture, shape 등의 feature를 이용해서 유사도 검색을 행할 뿐이며 대부분의 이미지 검색에서 사용되는 특징소들은 사용자의 검색시에 적용되는 모든 판단의 기준, 즉 주관적 또는 객관적 판단의 척도를 적용하기에는 제한적인 특징소들을 사용할 뿐이다.또한 이미지대 이미지 검색도 2D Image에 대한 검색만이 있을 뿐이며 이미지의 검색의 속도나 그 검색의 결과에 신뢰성이 떨어진다. 현재 Yahoo, Empas, Naver등 포털 사이트에서 이미지 검색이 시도되고 있으며 섬유 및 디자인 부분에서 이미지 검색의 필요성이 점차 증가되고 있다.

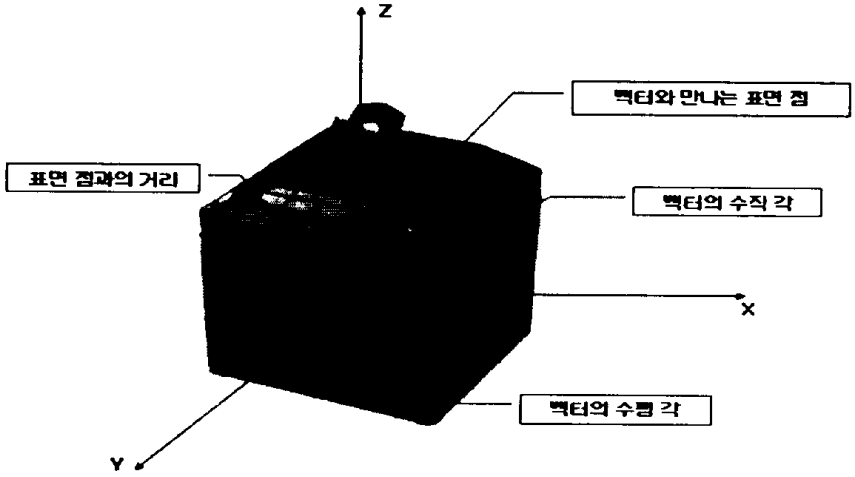
이러한 수요에 의해 국내에서도 2차원 영상에 관련하여 K통신회사에서 '내용기반 정지영상 검색 시스템개발' 연구를 진행하였으나 상용화되지 못하였고, 민간차원에서는 C정보통신에서 텍스트기반의 영상검색시스템을 개발하여 판매하고 있다. 그리고 전자통신연구소에서 상표검색시스템을 개발하여 자체적으로 사용하고 있으나 대부분이 정지영상기반 검색기술에 국한되어 있다.[Gose]

제 2 장 3차원 불변특징벡터 추출 알고리즘

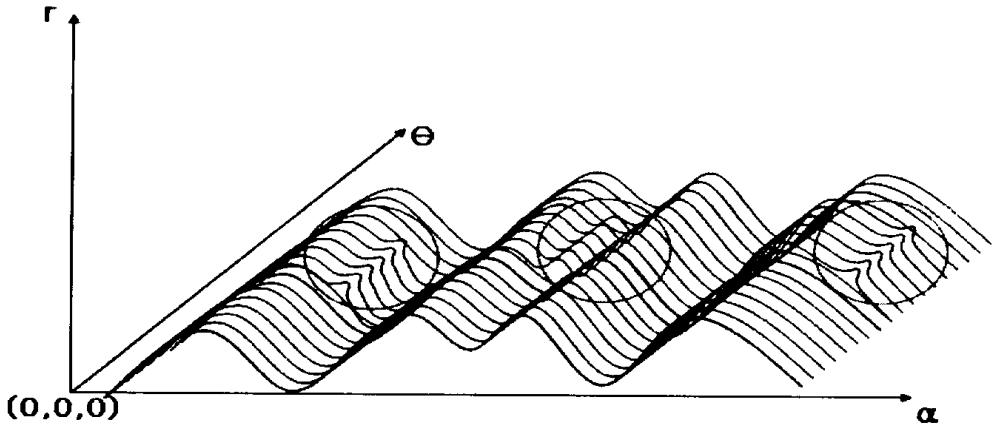
3D 표준파일포맷 DXF, Raw scan data의 3D 객체에는 3차원 좌표 x,y,z 로 표현될 수 있는 수많은 임의의 점들로 구성되어 있다.

아래의 그림<2.1>과 같은 3차원 객체가 존재할 때 3D 표준파일포맷, Raw scan data에서 3차원 형상의 임의의 점 (x, y, z)좌표 상에 최적의 중심점(0, 0, 0)과 각 좌표축을 기준으로 하는 회전각과 물체와 중심점과의 거리를 구할 수 있으며 이를 바탕으로 하나의 영상을 만들고 이 영상을 주파수 영역으로 변환이 가능하다. 즉 물체의 x, y, z 좌표 상에서 z축에 대한 회전각을 θ, y축에 대한 회전각을 α, 중심점으로부터의 임의의 벡터 v와 3차원 형상의 표면과 만나는 표면 점을 가정한다면 중심점의 위치는 최적의 임의의 값을 정할 수 있으며 θ, α (0 =< α, θ =<360)가 변함에 따라 만나게 되는 표면 점들의 좌표 값도 정할 수 있어 이를 바탕으로 중심점과의 거리를 구할 수 있게 된다. 따라서 α, θ,거리를 축으로 하는 3차원 그래프를 구할 수 있다.

각의 변화에 따라 접하게 되는 3차원 형상의 중심점으로부터 표면 점까지의 거리를 통해 주기성이 존재하는 주파수를 찾아낸다. 이렇게 추출된 주파수 성분을 가지고 3차원 형상의 특징벡터를 추출하게 된다. 직접 3차원 좌표 상에 r, θ, α 의 형식으로 표면 점을 표현하면 그림 <2.2>와 같은 형태의 파형이 형성된다.



그림<2.1> 차원 형상 좌표에서의 벡터추출 모형



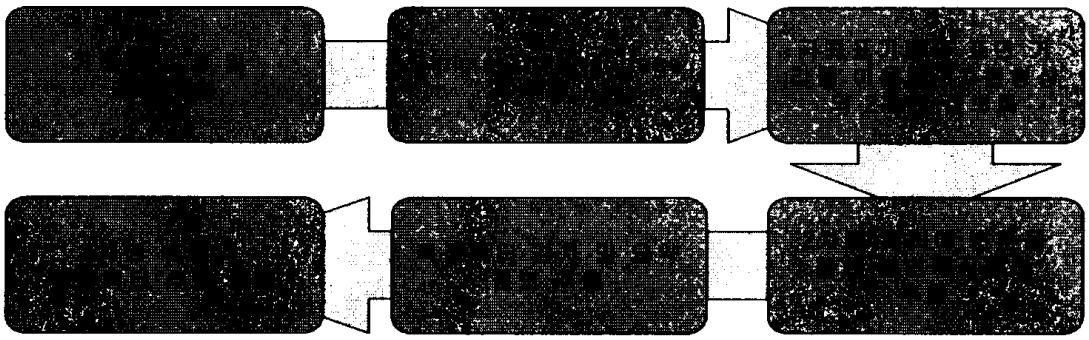
그림<2.2> 그림<2.1>의 각 좌표계로 변환된 표면 파형

이를 바탕으로 거리 축에 평행한 영상정보를 획득할 수 있게 된다. 즉, 영상정보를 획득하기 위해서 영상정보를 표시할 수 있도록 표준화를 선 처리해야 한다.

따라서 특징벡터인 주파수영역으로의 변환이 가능하게 되므로 이 주파수를 활용하여 검색이 가능하게 된다. 여기서, 추출된 주파수들 중에서 주요 주파수 성분을 추출한다. 이들은 2차원 주파수 Domain에 속하게 된다.

다.[Arbter1990]

3차원 객체 형상 불변특징벡터 추출 기술의 목적은 3D 모델 데이터의 형상이 가지고 있는 고유의 특징정보를 추출하는 것으로 개발된 기술의 주요내용은 3D 모델의 일정한 방향을 잡기위한 기준 축 선정 및 좌표변환, 3D 객체의 minimal outer surface 추출기술, Minimal outer surface에 대한 극 좌표계 기반의 distance 추출을 통한 3D형상정보의 2차원 정보화 기술, 마지막으로 Distance 정보에 대한 불변특징벡터 추출 과정으로 그림 <2.3>과 같다.[Weiyu2000]



그림<2.3> 3차원 고유 형상 특징불변벡터 추출의 구성

2.1 3D 모델의 기준축 선정 및 좌표변환

3차원 객체를 표현하는 방법 중 많이 사용되는 매쉬 모델은 다음 식과 같이 삼각형의 집합으로 표현된다.

$$\begin{aligned} object &= \{m_i\}\ i = 1,,,n \\ m_i &= (p_1, p_2, p_3) \\ p_j &= (x_j, y_j, z_j) \end{aligned} \tag{2.1.1}$$

위 식에서 p_j는 각 매쉬를 구성하는 삼각형의 꼭지점이다.

동일한 객체에 대해서 매쉬를 표현하는 좌표 값은 축을 달리할 경우

다른 값을 가지게 되어 3차원 객체의 경우 축의 방향에 무관한 불변 특징 벡터를 추출하기위해 축을 고정할 필요성을 가진다. 어느 한 객체에 대해 동일한 축을 찾아내어 이를 기준으로 좌표계를 고정 시키는 것은 3차원 객체를 비교하는 데 중요한 작업이다. 이를 위해 본 과제에서는 하나의 3차원 직선에 대해 모든 매쉬의 직교거리의 합이 최소가 되는 평면방정식을 구하고 그 평면을 xy평면으로 고정하는 방법을 사용하였다.

하나의 평면 $ax+by+cz+d$에 대한 모든 매쉬의 직교거리의 자승 합은 다음 식으로 표현된다.

$$D(a,b,c,d) = \sum_{i=1}^{n} d(m_i)$$

$$d(m_i) = A_i^2 \frac{(ax_{ci} - by_{ci} - cz_{ci} + d)^2}{a^2 + b^2 + c^2} \qquad (2.1.2)$$

A_i : *Area of mesh*

(x_{ci}, y_{ci}, z_{ci}) : *COG of mesh*

위 함수 $D(a,b,c,d)$를 a, b, c, d에 대해 각각 편미분하고 해 a^*, b^*, c^*, d^*를 구하면 모든 매쉬에 대해 직교거리의 합이 최소인 3차원 평면 방정식을 얻게 된다. 이 평면식은 회전 대칭이 아닌 3차원 물체에 대해 하나의 해를 가지게 된다. 이 평면을 xy평면으로 하고 물체의 무게중심을 원점으로 변환하면 하나의 3차원 물체는 동일한 z축을 가지며 가질 수 있는 3차원 좌표는 z축에 대해 회전된 정보를 가지게 되어 임의의 3차원 객체는 기준 축 적용 전에는 3개축에 대해 회전 자유도를 가지나 기준 축을 적용함으로써 z축을 중심으로 하는 회전 자유도만을 가지게 된다.

이러한 결과는 다음에 나올 극 좌표계 거리변환을 통해 하나의 3차원 객체가 2차원 정보로 표현될 때 축 회전 변환이 이동변환으로 나타나는 특성을 가지는 형태가 되어 불변벡터 추출이 용이하게 된다.

2.2 객체 최소폐곡면 추출기술

객체는 특정한 응용 목적에 맞도록 smoothing, sharpening 등의 전처리 과정을 거쳐 폐곡면을 추출하게 된다. 3차원 객체에서 폐곡면을 추출하는 이유는 변환된 3차원 객체 자체를 모두 표현하는 데에는 많은 데이터가 필요하게 되므로 일정 간격으로 표본을 추출하여 객체 정보를 효과적으로 표현하고자 함이며 이러한 과정을 통해 폐곡면 정보를 획득, 표현하게 된다.

폐곡면 정보 추출 및 표현은 표본 결정 및 추출, 표현 방법에 따라 여러 가지 방법이 있다 본 과제에서는 3차원 객체의 형상을 최대한으로 유지하면서 3차원 좌표 정보를 2차원 정보체계로 변환하기 용이한 형태의 폐곡면을 추출하는 방법을 연구하였다.

본 과제에서 사용한 최소 폐곡면은 3차원 객체의 COG(Center of Gravity)를 원점으로 xy평면에 대한 각 α와 xz평면에 대한 각 β를 가지는 임의의 방향벡터와 교차하는 객체의 표면점 중 가장 바깥에 있는 표면점의 집합으로 이루어진 곡면을 최소 폐곡면이라 정의하였다. 이 폐곡면은 3차원 객체의 무게중심을 기준으로 외양을 둘러쌓고 있는 최소크기의 폐곡면을 형성하여 외양을 최대한으로 표현하는 특성을 가지고 있다.

3차원극좌표에서 임의의 점은 원점으로부터의 거리(r)와 두 개의 각도 정보(α, β)로 표현되는데 원점을 폐곡면의 중점에 위치시키면, 폐곡면 정보는 중점을 중심으로 객체의 $P(r, \alpha, \beta)$을 구해 재구성함으로써 획득할 수 있다. α와 β가 각각 일정 간격의 각도로 변화하고 폐곡면은 다음과 같이 각각의 점P의 집합으로 표현할 수 있다.

$$S = \{ P_{\alpha,\beta} \mid -\frac{\pi}{2} =< \alpha =< \frac{\pi}{2}, 0 =< \beta =< 2\pi \} \qquad (2.2.1)$$

2.3 최소 폐곡면에 대한 극좌표계 기반 2차원 정보추출

객체의 내측거리 값을 $r_{\alpha,\beta}$ 이라 하고 이를 3차원극좌표상에 도식하면 폐곡면을 얻을 수 있지만 특징 벡터의 크기를 줄이고 표현을 효율적이고 용이하게 하기 위해서 2차원 정보로 표현하고자 X축은 각도 α값을, Y축은 각도 β값을 나타내고 각도의 변화크기를 $\Delta\alpha$와 $\Delta\beta$라 하면 크기가 $M \times N$이고 원소가 $r_{\alpha,\beta}$인 행렬로 나타낼 수 있다. 식(2.3.1)는 각도의 간격을 각각 $\Delta\alpha$, $\Delta\beta$로 했을 때 내측거리 값을 2차원 행렬로 표현한 것이다. 이 행렬은 행의 크기 $N = \pi/\Delta\alpha$, 열의 크기 $M = 2\pi/\Delta\beta$가 된다.

$$R = \begin{vmatrix} r_{-\pi/2,0} & r_{-\pi/2,\Delta\beta} & \cdots & r_{-\pi/2,2\pi-\Delta\beta} \\ r_{-\pi/2+\Delta\alpha,0} & r_{-\pi/2+\Delta\alpha,\Delta\beta} & \cdots & r_{-\pi/2+\Delta\alpha,2\pi-\Delta\beta} \\ & & & \\ r_{\pi/2,0} & r_{\pi/2,\Delta\beta} & \cdots & r_{\pi/2,2\pi-\Delta\beta} \end{vmatrix} \qquad (2.3.1)$$

행렬 R은 동일한 3차원 객체에 대해서도 그 크기에 따라 다른 값을 가지게 된다. 이를 행렬인자 중 최대값 $r^* = max(r_{i,j})$을 찾아 행렬에 대해 정규화를 하면 식(2.3.2)와 같은 행렬식을 얻게 되며 이 행렬은 동일한 객체에 대해 크기에 무관한 특성을 지니게 되며 그 값은 [0,1]의 범위를 가진다.

$$R^* = \frac{1}{r^*} \begin{vmatrix} r_{-\pi/2,0} & r_{-\pi/2,\Delta\beta} & \cdots & r_{-\pi/2,2\pi-\Delta\beta} \\ r_{-\pi/2+\Delta\alpha,0} & r_{-\pi/2+\Delta\alpha,\Delta\beta} & \cdots & r_{-\pi/2+\Delta\alpha,2\pi-\Delta\beta} \\ & & & \\ r_{\pi/2,0} & r_{\pi/2,\Delta\beta} & \cdots & r_{\pi/2,2\pi-\Delta\beta} \end{vmatrix} \qquad (2.3.2)$$

2.4 불변 특징 벡터 추출

2.4.1 변형 불변 정의

위치 이동의 경우 패턴 구성 요소 점 $P(r_{\alpha,\beta}, \alpha, \beta)$를 이동 연산자 T에 의해 $(dr_{\alpha,\beta}, d\alpha, d\beta)$만큼 위치 이동시킨 점 $P'(r_{\alpha,\beta}', \alpha', \beta')$을 행렬식으로 표현하고 연산자와의 관계를 나타내면 다음 식(2.4.1)과 같다.

$$\begin{bmatrix} r_{\alpha,\beta}' \\ \alpha' \\ \beta' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & dr_{\alpha,\beta} \\ 0 & 1 & 0 & d\alpha \\ 0 & 0 & 1 & d\beta \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{\alpha,\beta} \\ \alpha \\ \beta \\ 1 \end{bmatrix} \tag{2.4.1}$$

$$P' = T_d \cdot P$$

크기 변화의 경우는 패턴을 구성하는 점 $P(r_{\alpha,\beta}, \alpha, \beta)$를 크기 연산자 S에 의해 a만큼 확대 또는 축소된 점 $P'(r_{\alpha,\beta}', \alpha', \beta')$의 행렬식과 연산 관계는 다음 식과 같이 표현된다.

$$\begin{bmatrix} r_{\alpha,\beta}' \\ \alpha' \\ \beta' \\ 1 \end{bmatrix} = \begin{bmatrix} S_a & 0 & 0 & 0 \\ 0 & S_a & 0 & 0 \\ 0 & 0 & S_a & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{\alpha,\beta} \\ \alpha \\ \beta \\ 1 \end{bmatrix} \tag{2.4.2}$$

$$P' = S_a \cdot P$$

회전 변화의 경우도 패턴을 구성하는 점 $P(r_{\alpha,\beta}, \alpha, \beta)$ 와 회전 연산자 R이 주어지면 점 P가 R에 의해 X, Y, Z축 기준으로 각각 θ만큼 회전한 점 $P'(r_{\alpha,\beta}', \alpha', \beta')$는 각각 다음 식(2.4.3)과 같다.

$$\begin{bmatrix} r_{\alpha,\beta}' \\ \alpha' \\ \beta' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{\alpha,\beta} \\ \alpha \\ \beta \\ 1 \end{bmatrix} \quad \text{(a)}$$

$$\begin{bmatrix} r_{\alpha,\beta}' \\ \alpha' \\ \beta' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{\alpha,\beta} \\ \alpha \\ \beta \\ 1 \end{bmatrix} \text{(b)} \qquad (2.4.3)$$

$$\begin{bmatrix} r_{\alpha,\beta}' \\ \alpha' \\ \beta' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{\alpha,\beta} \\ \alpha \\ \beta \\ 1 \end{bmatrix} \text{(c)}$$

$P' = R_\theta \cdot P$

따라서 세 가지 연산자를 이용하여 위치 이동과 크기 변형, 회전 변형이 동시에 가해진 패턴은 식(2.4.4)와 같이 표현되므로,

$$P' = R_\theta \cdot S_a \cdot T_a \cdot P \qquad (2.4.4)$$

인식시스템을 ψ, 변형이 가해지기 전의 인식대상을 X, 변형이 가해진 후의 인식대상을 X'라 하면 위치 이동 및 크기 변화, 회전 변화에 불변인 문제는 식(2.4.5)과 같이 표현된다.

$$\psi(X) = \psi(R_\theta \cdot S_a \cdot T_d \cdot X) = \psi(X') \qquad (2.4.5)$$

즉 위치 이동, 크기 변화, 회전 변화에 불변인 문제는 변형이 가해지기 전의 인식 결과와 변형이 가해진 후의 인식 결과와 같아야 한다는 것을 의미한다.

2.4.2 위치 이동에 불변

특징 벡터가 위치 이동에 불변임을 보이기 위하여 $r_{\alpha,\beta}$를 이용한다. 내측거리 값들의 집합을 내측거리벡터 $R_{\alpha,\beta}$라고 하면 식 (2.2.4)와 (2.2.5)에 따라 $R_{\alpha,\beta} = R_{x,y}$라고 정의할 수 있다. 따라서 $R_{\alpha,\beta}$는 식(2.4.6)과 같이 정의할

수 있고,

$$
\begin{aligned}
&R_{\alpha,\beta} = \\
&\{r_{\alpha,\beta} \mid \alpha = 0, \Delta\alpha, \cdots, M-\Delta\alpha,\ \beta = 0, \Delta\beta, \cdots, N-\Delta\beta\} \\
&= \{r_{x,y} \mid x = 0, 1, \cdots, M-1,\ \ y = 0, 1, \cdots, N-1\} \\
&= R_{x,y}
\end{aligned}
\tag{2.4.6}
$$

$R_{\alpha,\beta} = R_{x,y}$의 구성 원소인 내측거리 값 $r_{\alpha,\beta}$는 식(2.4.7)와 같이 정의할 수 있다.

$$
r_{\alpha,\beta} = [(x_i - x_c)^2 + (y_i - y_c)^2 + (z_i - z_c)^2]^{\frac{1}{2}} \tag{2.4.7}
$$

$$
\alpha = 0, \Delta\alpha, \cdots, M - \Delta\alpha
$$
$$
\beta = 0, \Delta\beta, \cdots, N - \Delta\beta
$$

$$
x_c = \frac{1}{D}\sum_{i=0}^{D-1} x_i,\ \ y_c = \frac{1}{D}\sum_{i=0}^{D-1} y_i,\ \ z_c = \frac{1}{D}\sum_{i=0}^{D-1} z_i
$$

여기서 좌표축이 각각 dx, dy, dz 만큼 이동하게 되면, 식(2.4.8)과 같이 중점도 각각 dx, dy, dz 만큼 이동하게 되고, 결국에는 폐곡면도 함께 이동하게 되어 $r_{\alpha,\beta}$는 불변이 되게 된다. 즉 좌표축 중심과 폐곡면상의 좌표 점과의 관계로 인해 중심과 폐곡면 각 화소의 좌표 점은 같은 방향, 같은 크기로 이동하므로 결국 위치 이동에 불변임을 알 수 있다.

$$
\frac{1}{D}\sum_{i=0}^{D-1}(x_i + dx) = \frac{1}{D}\sum_{i=0}^{D-1} x_i + \frac{1}{D}\sum_{i=0}^{D-1} dx = x_c + dx
$$

$$
\frac{1}{D}\sum_{i=0}^{D-1}(y_i + dy) = \frac{1}{D}\sum_{i=0}^{D-1} y_i + \frac{1}{D}\sum_{i=0}^{D-1} dy = y_c + dy \tag{2.4.8}
$$

$$
\begin{aligned}
&\therefore [(x_i + dx - x_c - dx)^2 + (y_i + dy - y_c - dy)^2 + (z_i + dz - z_c - dz)^2]^{\frac{1}{2}} \\
&= [(x_i - x_c)^2 + (y_i - y_c)^2 + (z_i - z_c)^2]^{\frac{1}{2}} \\
&= r_{\alpha,\beta}
\end{aligned}
$$

2.4.3 크기 변화에 불변

입력 패턴의 크기 변화에 따라 폐곡면상의 화소수의 변화와 내측거리 값의 변화가 발생한다. 따라서 크기 변형에 불변인 특징을 추출하기 위해서는 내측거리 값의 크기를 정규화 한다. 정규화된 내측거리벡터를 ℝ이라고 하면 식(2.2.11)에 의해 $R_{0,0} = R_{x,y}$이므로 정규내측거리벡터 ℝ은 식(2.4.9)와 같다.

$$\mathbb{R} = \{ \frac{R_{x,y}}{|r|} \mid x = 0, 1, \cdots, M-1 \qquad y = 0, 1, \cdots, N-1 \} \tag{2.4.9}$$

단 $|r| = (\sum_{y=0}^{N-1} \sum_{x=0}^{M-1} r_{x,y}^2)^{\frac{1}{2}}$

따라서 객체가 a만큼 확대 또는 축소되더라도 식(2.4.10)와 같이 정규내측거리벡터 ℝ은 불변이 되게 된다. 즉 정규화된 내측거리 값들은, 물체가 확대 또는 축소되더라도 크기 변화에 불변임을 알 수 있다.

$$(\sum_{y=0}^{N-1} \sum_{x=0}^{M-1} a^2 r_{x,y}^2)^{\frac{1}{2}} = a(\sum_{y=0}^{N-1} \sum_{x=0}^{M-1} r_{x,y}^2)^{\frac{1}{2}} = a|r| \tag{2.4.10}$$

$$\therefore \frac{aR_{x,y}}{a|r|} = \frac{R_{x,y}}{|r|} = \mathbb{R}$$

2.4.4 회전 변화에 불변

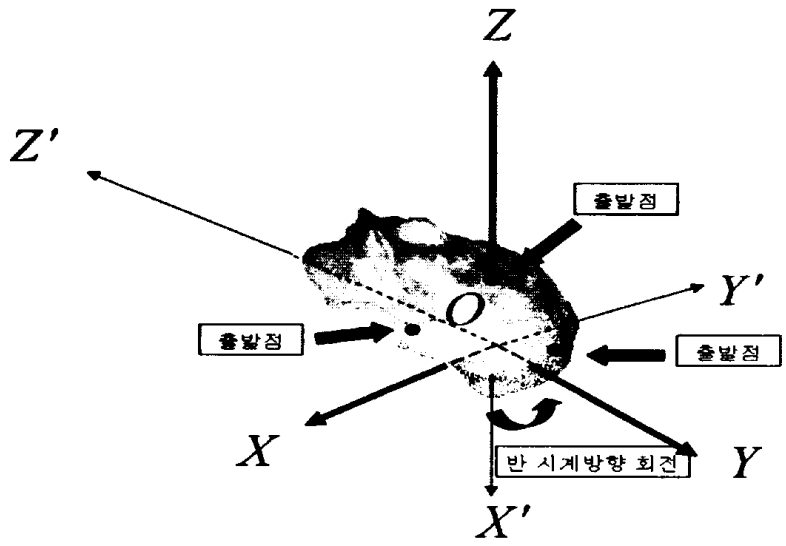
정규내측거리벡터의 제곱합은 파시벌정리(Parseval's Theorem)에 의해 다음 식(2.4.11)과 같이 표현된다.

$$\mathbb{R}^2 = \sum_{y=0}^{N-1}\sum_{x=0}^{M-1}\frac{R_{x,y}^2}{|r|^2} = 1 = \sum_{v=0}^{N-1}\sum_{u=0}^{M-1}P(u,v) \tag{2.4.11}$$

따라서 정규내측거리벡터 $\mathbb{R}$ 즉 정규내측거리 값들은, 파시벌의 정리를 적용하여 주파수처럼 각각의 성분으로 분해가 가능하므로 다음 식(2.4.12)과 같이 퓨리에 변환을 이용하여 이 벡터의 Fourier 계수를 구할 수 있다.

$$\begin{aligned} F(u,v) &= \frac{1}{AB}\sum_{a=0}^{A-1}\sum_{b=0}^{B-1}f(x,y)\exp\left[-j2\pi\left(\frac{ax}{A}+\frac{by}{B}\right)\right] \\ &= \frac{1}{MN}\sum_{y=0}^{N-1}\sum_{x=0}^{M-1}\vec{r}\exp\left[-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right] \\ u &= 0,1,\cdots,M-1 \quad v = 0,1,\cdots,N-1 \end{aligned} \tag{2.4.12}$$

3차원 객체의 회전은 그림<2.4> 에서 보는 것처럼 폐곡면 추출의 출발점 변화로 볼 수 있다.



그림<2.4> 객체 회전에 따른 출발점의 변화

그러므로 객체의 회전은 폐곡면의 2차원 영상의 Fourier 계수를 X축으로 dx화소, Y축으로 dy화소만큼 이동한 것으로 볼 수 있어 식(2.4.1)과 같이 위상 변화만 생길 뿐 진폭 변화는 발생하지 않는다.

$$
\begin{aligned}
&F(u,v)\cdot \exp\left[-j2\pi\left(\frac{u\cdot dx}{M}+\frac{v\cdot dy}{N}\right)\right] \\
&=f(x,y)\cdot \exp\left[-j2\pi\left(\frac{u(x+dx)}{M}+\frac{v(y+dy)}{N}\right)\right] \\
&=F_{dx,dy}(u,v) \\
&u=0,1,\cdots,M-1 \quad v=0,1,\cdots,N-1
\end{aligned}
\tag{2.4.13}
$$

따라서 다음 식(2.4.14)과 같이 Power Spectrum에 적용하면 객체가 좌표축을 기준으로 회전한다 하더라도 결국 회전 변형에 불변임을 알 수 있다.

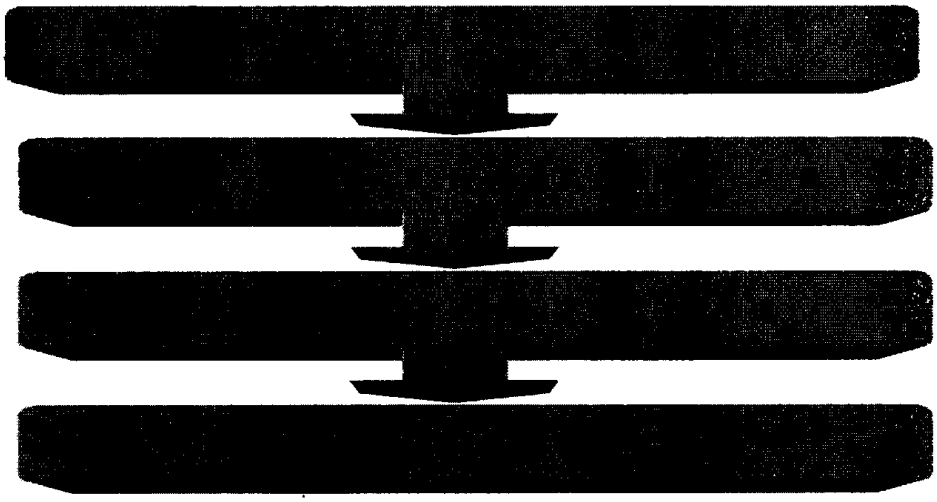
$$
\begin{aligned}
&P_{dx,dy}(u,v)=|F_{dx,dy}(u,v)|^2=|F(u,v)|^2=P(u,v) \\
&u=0,1,\cdots,M-1 \quad v=0,1,\cdots,N-1
\end{aligned}
\tag{2.4.14}
$$

제3장 2차원 매칭을 위한 불변특징벡터 추출 알고리즘

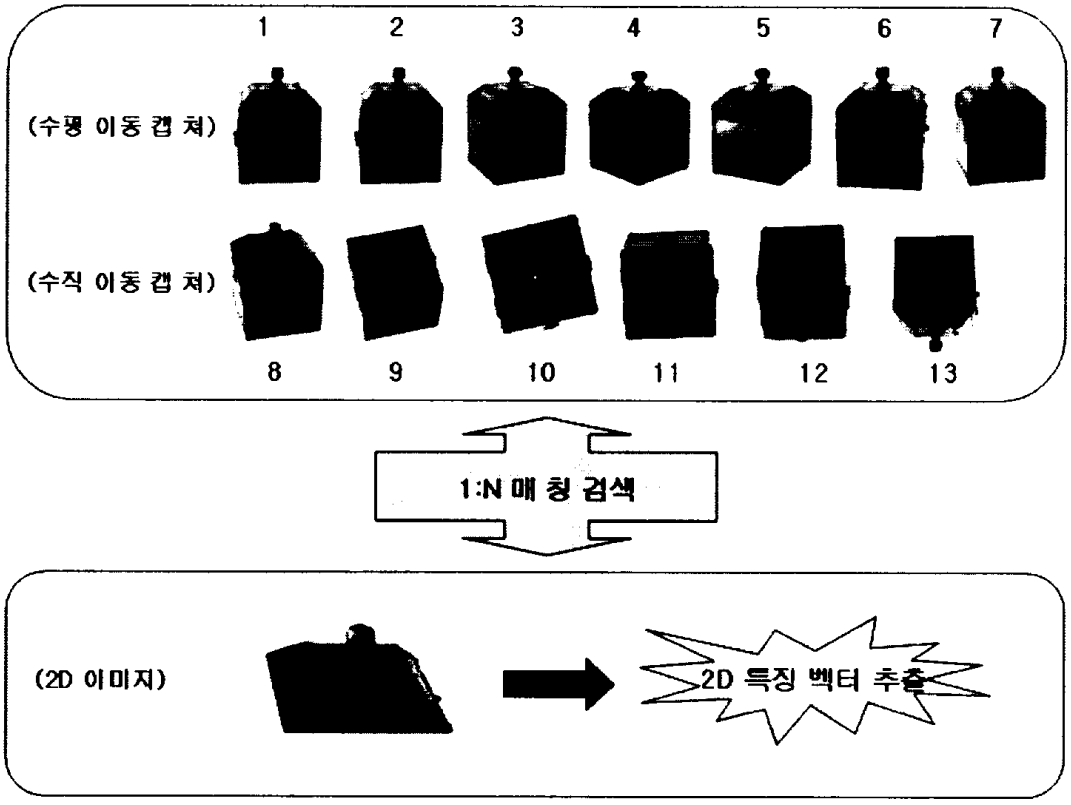
임의의 3차원 객체를 일정한 회전량으로 구분된 여러 각도로 객체의 전후 상하 좌우의 영상을 획득한 후 획득한 영상들을 클러스터링하면 유사한 영상별로 그룹 화되고 그룹화 된 영상 중 하나의 대표영상의 특징벡터 값을 추출하고 이를 이용하면 구하고자 하는 2차원영상이 주어졌을 때 주어진 영상의 특징벡터 값과 3차원객체의 추출된 2차원영상 벡터 값에 적용할 수 있다.

그림<3.2>와 같이 3차원 모델에서 추출된 이미지에서 특징 벡터를 추출한 후 검색을 위해 입력된 2차원 이미지에서의 특징 벡터와 매칭이 이루어진다. 2차원 이미지 위치, 크기, 회전에 불변한 특징벡터의 추출은 기존 연구들이 다수 이루어져 있다.

간단한 개념도를 살펴보면 그림<3.1>과 같다.



그림<3.1> 3D-2D 매칭 개념도



그림<3.2> 3차원 모델에서의 회전각에 따른 이미지 추출 및 매칭

3.1 Zernike 모멘트를 이용한 2D 특징 벡터 추출

특징 벡터의 추출을 위해서 3차원 객체를 일정한 각도로 x, y, z축에 대해서 30도씩 회전시키고 이때 얻어진 1728($12\times12\times12$)개의 2차원 이미지를 각각 이진화 및 영역의 정규화를 시켜, 실용적인 패턴 인식 시스템을 구축하기 위한 특징벡터를 추출하기 위해서 영역 기반 형상 서술자의 하나인 Zernike 모멘트 연산자로 각각의 이미지에 대한 특징 벡터를 10차원까지 추출한다. Zernike 모멘트는 MPEG-7에서 영역 기반 형상 서술자의 하나로 실험중인 모델이다. 이 서술자는 물체의 형태를 표현할 때 외곽경계정보뿐만 아니라 물체내의 정보도 이용할 수 있다는 장점이 있고, 회전에 불변하고 노이즈에 강하며, 모양의 왜곡에도 다소 강한 특징을 가지고

있다. 그리고 다진 영상으로 나타난 물체를 이진화 하는데 필요한 임계치의 변화에도 다른 시스템보다도 덜 민감하다. 또한 상이한 크기와 임의의 회전방향에 있는 객체들을 인식하는데 유리한 점들이 있기 때문에 사용되었다.

3.1.1 Zernike 모멘트의 정의

Zernike는 단위원(unit circle)내에서 정의 되어지는 직교 집합을 형성하는 복소수 다항식을 소개한다. 디지털 영상에 대한 $(m+n)$차의 Zernike 모멘트는 (ρ,θ)의 극 좌표체계에서 아래의 식 (3.1.1)과 같이 정의된다.

$$\begin{aligned} A_{nm} &= \frac{n+1}{\pi}\sum_{x}\sum_{y} f(x,y)\,V^{*}_{nm}(x,y),\ x^2+y^2\le 1 \\ &= \frac{n+1}{\pi}\sum_{\rho}\sum_{\theta} f(\rho,\theta)\,V^{*}_{nm}(\rho,\theta),\ \rho\le 1 \end{aligned} \tag{3.1.1}$$

여기서, ρ는 원점과 (x,y)픽셀간의 벡터의 길이이고, θ는 ρ와 x축사이의 시계반대 방향으로의 각도이다. 그리고 벡터 $V_{nm}(\rho,\theta)$는 Zernike basis polynomial로, 다음의 식 (3.1.2)로 정의되어지며, $V^{*}_{nm}(\rho,\theta)$는 $V_{nm}(\rho,\theta)$의 공액 복소수이다.

$$V_{nm}(x,y) = V_{nm}(\rho,\theta) = R_{nm}(\rho)\,exp(jm\theta) \tag{3.1.2}$$

또한, radial polynomial $R_{nm}(\rho)$는 아래의 식 (3.1.3)으로 정의되어진다.

$$R_{nm}(\rho) = \sum_{s=0}^{\frac{n-|m|}{2}} (-1)^s \times \frac{(n-s)!}{s!\left(\frac{n+|m|}{2}-s\right)!\left(\frac{n-|m|}{2}-s\right)!}\rho^{n-2s} \tag{3.1.3}$$

여기서, $n=0,1,2\cdots,\infty$, $n+|m|$ 과 $n-|m|$ 는 짝수이다. 그리고 이 다

항식(polynomials)들이 직교(orthogonal)한다. Zernike 모멘트는 이미지 함수를 이러한 직교 기본 함수(orthogonal basis function) 위에 투영한 것이다. 주어진 이미지의 Zernike 모멘트를 계산하기 위하여 이미지의 중심이 원점으로 취해지고 픽셀 좌표들은 단위원의 범위($x^2+y^2 \leq 1$)로 사상되어진다. 단위 원 바깥으로 떨어지는 픽셀들은 계산에서 사용되어지지 않는다. 그리고 $A^*_{nm} = A_{n,-m}$이다. n = 10차까지의 Zernike 모멘트 개수는 36개로 다음의 표 3.2.1과 같이 나타낼 수 있다.

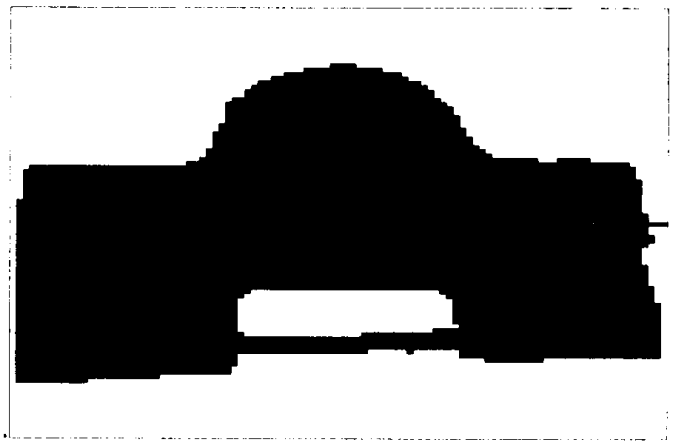
표 3.1.1 10차까지의 해당 차원의 모멘트 수

n	모 멘 트	모 멘 트 개 수
0	$A_{0,0}$	1
1	$A_{1,1}$	1
2	$A_{2,0}$ $A_{2,2}$	2
3	$A_{3,1}$ $A_{3,3}$	2
4	$A_{4,0}$ $A_{4,2}$ $A_{4,4}$	3
5	$A_{5,1}$ $A_{5,3}$ $A_{5,5}$	3
6	$A_{6,0}$ $A_{6,2}$ $A_{6,4}$ $A_{6,6}$	4
7	$A_{7,1}$ $A_{7,3}$ $A_{7,5}$ $A_{7,7}$	4
8	$A_{8,0}$ $A_{8,2}$ $A_{8,4}$ $A_{8,6}$ $A_{8,8}$	5
9	$A_{9,1}$ $A_{9,3}$ $A_{9,5}$ $A_{9,7}$ $A_{9,9}$	5
10	$A_{10,0}$ $A_{10,2}$ $A_{10,4}$ $A_{10,6}$ $A_{10,8}$ $A_{10,10}$	6

3.1.2 Zernike 모멘트의 계산과 특성

3차원 이미지의 회전을 통해서 얻어진 2차원 이미지들 각각에 대한 특징 벡터를 추출하기 위해서, 먼저 2차원 이미지들을 이진화 시킨다. Zernike 모멘트는 단위원에 대해서 정의되어지므로, 그림<3.1.1>과 같이 물체의 중심으로부터 물체의 가장 밖에 있는 픽셀까지 완전히 포함할 수 있는 반경 R인 원을 결정하고, 이미지의 크기가 2R × 2R 크기로 되도록 이미지의 크기를 정규화 한다. 정규화된 이미지에 대해서 n = 10차에 대

한 36개의 Zernike 모멘트를 계산한다



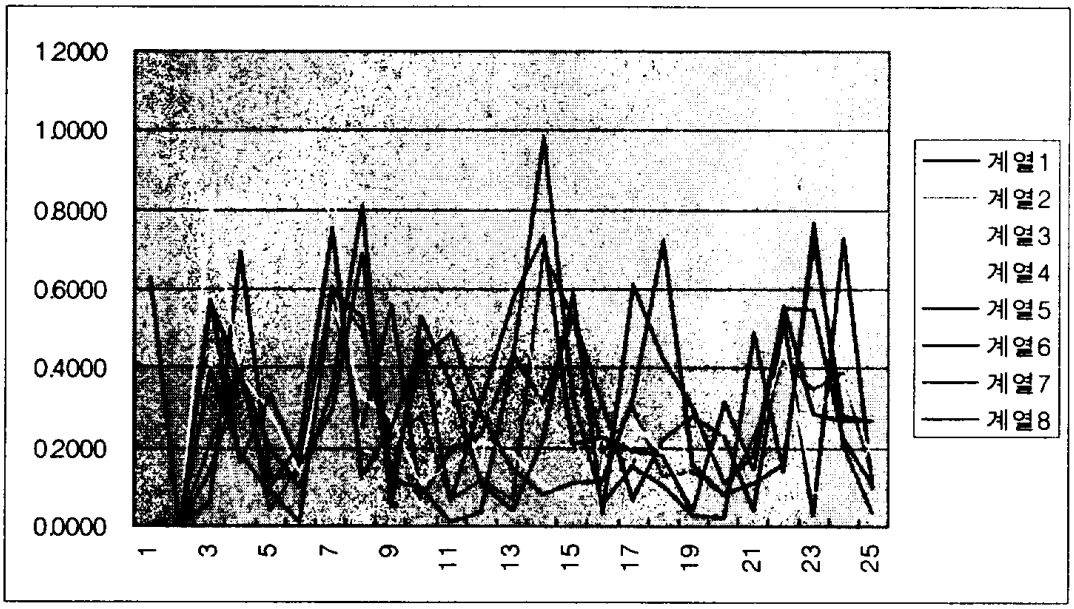
그림< 3.1.1> R=12로 이미지 정규화

Zernike모멘트 계산에서 radial polynomial $R_{nm}(\rho)$함수는 오프라인에서 계산하여 룩업 표를 작성 메모리에 저장하고 모멘트를 구할 때는 저장된 룩업 표를 이용하면 계산의 속도를 향상 시킬 수 있다.

표 3.1.2 도자기 객체에 대해서 추출된 특징 벡터의 예

RX	RY	RZ	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	0	0	2486	0	237.704	51.171	0.967000	26.464	365.474	74.538	10.87	3.568
0	0	30	2504.462	0	470.517	42.494	2.802	45.952	675.564	52.472	73.469	9.534
0	0	60	2443.346	0	313.881	54.684	2.678	68.04601	467.71	75.804	50.791	9.461
0	0	90	2493.957	0	218.778	47.804	1.282	68.75	335.532	69.367	28.968	4.708
0	0	120	2562.394	0	410.743	33.765	0.448	13.746	601.854	43.089	102.902	1.513
0	0	150	2502.87	0	360.869	34.692	2.444	47.735	534.4821	48.085	64.493	8.734
0	0	180	2451.622	0	294.938	48.645	1.538	66.343	445.914	68.204	20.86	5.631
0	0	210	2500.96	0	299.143	46.746	1.528	58.585	448.385	64.187	67.722	5.457
0	0	240	2450.349	0	379.635	50.088	1.345	26.275	559.475	68.11	46.777	4.839
0	0	270	2491.093	0	302.173	49.961	1.039	34.683	456.132	69.374	18.674	3.692
0	0	300	2555.391	0	303.498	42.201	1.691	71.908	453.737	58.368	93.52801	6.148
0	0	330	2506.372	0	378.249	49.169	1.034	54.238	555.563	63.998	67.953	3.613
0	30	0	1881.648	0	1978.789	239.212	142.482	227.513	718.71	15.375	231.85	183.054
0	30	30	1925.774	0	1953.91	248.016	146	253.874	778.481	24.556	237.444	199.321
0	30	60	2487.91	0	2554.501	307.797	178.62	312.346	1010.481	18.144	308.614	234.305
0	30	90	1964.926	0	1971.123	230.172	138.887	240.953	831.73	34.155	235.089	193.145
0	30	120	2023.177	0	2025.666	248.143	145.269	250.596	851.437	34.165	244.881	201.836
0	30	150	2408.014	0	2543.5	300.946	172.399	304.978	932.822	23.052	295.647	209.493
0	30	180	1913.679	0	1967.054	235.184	139.541	253.179	771.639	18.171	225.649	186.223
0	30	210	1963.017	0	2037.278	254.038	151.428	248.793	759.997	18.664	244.789	199.808
0	30	240	2414.38	0	2555.952	323.666	188.976	298.1	896.204	12.15	320.278	234.504
0	30	270	1946.783	0	2088.688	237.526	156.557	256.31	703.326	13.902	235.033	188.605
0	30	300	2009.49	0	2114.804	261.671	161.952	274.887	746.3911	17.562	249.345	203.764

위의 표 3.1.2는 3차원 도자기를 일정한 각도로 x, y, z축에 대해서 30



그림<3.1.3> 25차원에서 다른 부류의 객체를 나타낸 경우

도씩 회전시키고 이때 얻어진 1728($12\times12\times12$)개의 2차원 이미지에 대해서 Zernike 모멘트를 계산하여 얻어진 특징 벡터의 일부를 나타낸 것이다.

많은 패턴 인식 시스템에서 같은 부류의 패턴을 인식할 때는 패턴의 조그만 차이에는 둔감해야하지만 다른 부류의 패턴일 경우에는 패턴사이의 차이에 민감해야 한다. 아래 그림들은 화소 개수인 첫 번째 필드의 값을 $2\pi/10$으로 고정하고 다른 필드의 값을 첫 번째 필드의 값을 기준으로 하여 변환시켰다. 이렇게 하여 필드간의 크기의 비율은 유지하면서 모든 필드의 값을 거의 0에서 1까지의 값으로 나타냈다. 그림<3.1.3>은 다른 부류의 객체에 속한 경우에 특징벡터들이 어떻게 나타나는지 객체들의 25차원의 값들을 보여준다. 그 결과 다른 부류의 객체들에 대한 특징 벡터들 간의 차이는 뚜렷하게 구분되어지고 있음을 보여주고 있다. 또한 그림<3.1.2>는 같은 객체의 특징 벡터들 간의 차이가 둔감함을 보여주고 있으

나, 아주 비슷하지는 않음을 알 수 있다. 그러므로 Zernike 모멘트로 표현하는 것이 적절하다는 것을 알 수 있다.

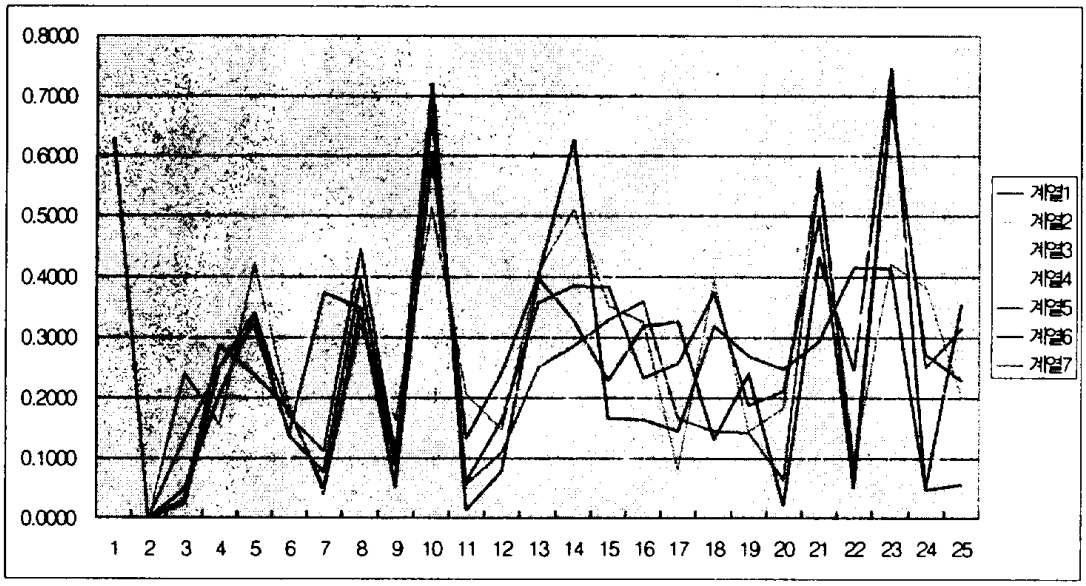
일반적으로 Zernike 모멘트는 다음과 같은 특성이 있다.

① Zernike 모멘트는 간단한 회전 변환 특성을 가지고 있다. 즉 α각도만큼 회전한 경우 $A'_{nm} = A_{nm}exp(-jm\alpha)$로 나타나기 때문에 회전에 대한 phase shift만 얻게 된다. 그러므로 회전된 Zernike 모멘트의 크기는 회전하기 전과 동일하게 된다.

② 기저(base polynomial)들이 직교하기 때문에 정보 중복이 없다. 그러므로 다른 모멘트보다 적은 수의 Zernike 모멘트에 의해서 이미지를 더 잘 묘사할 수 있다.

③ 형태의 조그만 왜곡과 잡음에 강하다.

④ 낮은 차수의 모멘트는 패턴의 전체 형상을 나타내고, 고차원의 모멘트는 세밀한 부분을 묘사하기 때문에 차원에 따른 계층적인 특성을 가진다.



그림<3.1.2> 25차원에서 같은 부류의 객체를 표시한 경우

다. Zernike 모멘트를 이용한 이미지 복원 이미지 $f(x, y)$의 $n_{\max}$차까지의 모든 Zernike 모멘트를 알고 있다면, 복원 이미지 $\hat{f}(x, y)$는 아래의 식 (3.1.4), 식 (3.1.5)와 식 (3.1.6)을 통해서 단계적으로 복원되어질 수 있다.

$$\begin{aligned} \hat{f}(x, y) &= \sum A_{nm} V_{nm}(\rho, \theta) \\ &= \sum_{n} \sum_{m>0} (C_{nm} \cos m\theta + S_{nm} \sin m\theta) R_{nm}(\rho) + \frac{C_{n0}}{2} R_{n0}(\rho) \end{aligned} \quad \forall x, y \qquad (3.1.4)$$

여기서,

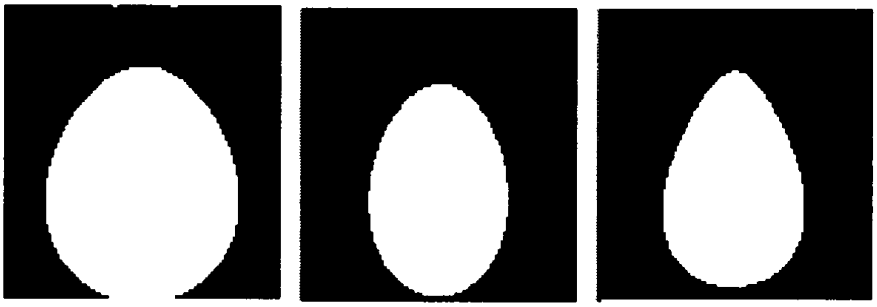
$$\begin{aligned} C_{nm} &= 2\, Real(A_{nm}) \\ &= \frac{2n+2}{\pi} \iint_{x^2+y^2 \le 1} f(x, y) R_{nm}(\rho) \cos m\theta \; dx \; dy \end{aligned} \qquad (3.1.5)$$

$$\begin{aligned} S_{nm} &= -2\, Imaginary\,(A_{nm}) \\ &= \frac{-2n-2}{\pi} \iint_{x^2+y^2 \le 1} f(x, y) R_{nm}(\rho) \sin m\theta \; dx \, dy \end{aligned} \qquad (3.1.6)$$

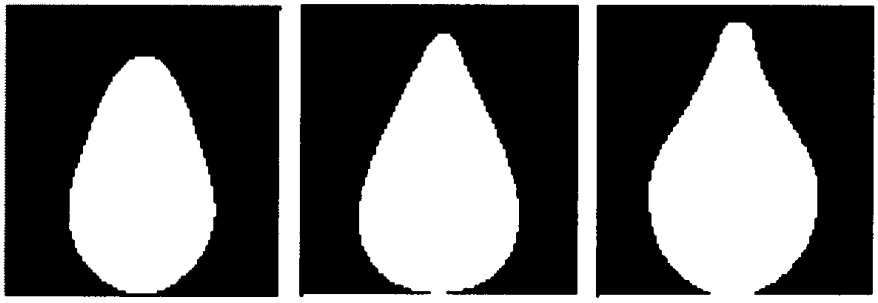
이다.

아래의 그림<3.2.5>는 식(3.1.4), 식(3.1.5), 식(3.1.6)을 이용하여 각 차원 별로 복원된 이미지를 나타낸 것이다. 복원된 이미지를 보면 n이 몇 차까지 이용되어져야 하는가의 질문에 대답해야 한다. 다른 말로 하면, 주어진 데이터베이스에 좋은 분류를 위해서 몇 차 모멘트까지 필요한지 결정해야한다. 사실상 이미지 표현을 위한 특징들을 선택하는 데에 큰 단점은 몇 차 모멘트까지 사용하는가에 대한 자동 선택을 위한 체계적인 방법이 없다. 좋은 특징들의 집합이라는 것은 이미지의 특성을 잘 나타내고 기술할 수 있는 것이다. 원래 이미지와 한정된 모멘트로부터 복원된 이미지와의 차이는 모멘트의 이미지 표현 능력에 대한 좋은 척도이다.

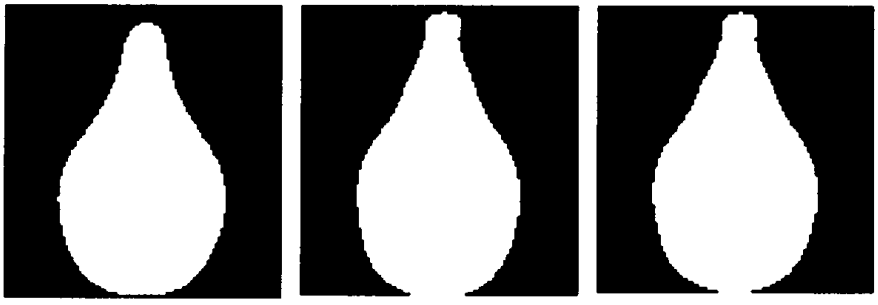
3D 객체 원본 2D이미지 3차 복원 이미지



4차 복원 이미지 5차 복원 이미지 6차 복원 이미지



7차 복원 이미지 8차 복원 이미지 9차 복원 이미지



10차 복원 이미지 11차 복원 이미지 12차 복원 이미지

그림<3.1.4> 2차원 특징벡터를 이용하여 복원된 도자기 이미지

Zernike 모멘트로부터 이미지 복원의 편이성은 특징 선택 과정을 그러한 척도에 대한 기초를 두는 것을 실질적인 것으로 만든다는 것이다. 즉 어떤 정의된 임계값의 관점에서 원본 이미지와 비슷한 복원된 이미지를 만들어 낼 수 있는 모멘트 차수를 결정하는 것이다.

0차에서부터 i 차까지의 모멘트를 이용하여 복원된 이진 이미지를 Ei로 표시하고, 원래 이미지픽셀과 복원된 이미지 픽셀을 비교하였을 때, 일치하지 않는 픽셀이 $H(E_i, E)$개 존재한다면 i 차원의 이미지 복원 기여도는 다음의 식 (3.1.7)과 같이 표현될 수 있다.

$$C(i) = H(E_{i-1}, E) - H(E_i, E) \tag{3.1.7}$$

여기서 E는 원본 이미지이고, E_i는 i번째 모멘트까지 사용하여 복원된 이미지를 말한다. 따라서 각 차원의 가중치 Wi는 식 (3.1.8)과 식 (3.1.9)을 통해 구할 수 있다. 만약 $C(i)$가 음수이면 가중치는 0으로 정의한다.

$$W_i = \frac{C(i)}{D(n^*)} \qquad i = 1, 2, \cdots, n^* \tag{3.1.8}$$

$$D(n^*) = \sum_{i=1}^{n^*} C(i),\ C(i) \geq 1 \tag{3.1.9}$$

여기서 n^*는 복원된 이미지를 만들어 내는데 필요한 최대한의 모멘트 차수를 의미한다. 이렇게 식에 의해서 그 값을 정할 수도 있으나, 복원된 이미지를 보면 8차나 9차에서 이미지 변화가 두드러지지 않음을 알 수가 있다.

3.2 3D 객체의 대표 2D 영상 클러스터링

다음은 앞의 방법에서 얻어진 객체 당 1728($12\times12\times12$)개의 2차원 특징 벡터들을 클러스터링 하여 3차원 객체의 불변 특징 벡터들을 구하는 방법들이다. $\{x^{(q)} \mid q=1,2,\cdots,Q\}$를 특징 벡터의 집합이라고 하고, $\{S^{(k)} \mid k=1,2,\cdots,K\}$를 클러스터의 집합이라 하자. 여기에서 Q개의 특징 벡터 $x^{(q)}=(x_1^{(q)}, x_2^{(q)}, \cdots, x_d^{(q)})$와 K개의 클러스터의 중심점 $c^{(k)}=(c_1^{(k)}, c_2^{(k)}, \cdots, c_d^{(k)})$이 각각 d차원으로 구성되어 있다면, 클러스터링 과정은 Q개의 특징 벡터들을 최소 거리 할당 원칙에 따라서 K개의 클러스터들로 할당하는 것이다. 첫 번째 방법은 보편적으로 클러스터링을 할 경우 사용하는 방법이고, 두 번째 방법은 특징벡터공간을 셀에 의해서 나누었을 때 셀 밀도를 계산해 낼 수 있으므로, 초기해를 어느 임계값 이상인 셀 밀도에 의해서 정하는 경우이고, 세 번째 방법은 클러스터의 평균값이 특이한 값(outlier)에 의해서 영향을 받으므로, 그 영향을 줄이기 위해서 가중치를 사용하는 방법을 보인다.

3.2.1 k-means 알고리듬

클러스터링은 특징벡터들을 클러스터들로 나누어주는 과정이다. 이 중에서도 $k-$means 클러스터링 알고리듬은 수많은 곳에서 적용되고 있는 아주 보편적이고 아주 간단한 방법이다. 기본적인 알고리듬은 다음과 같다.

① K개의 클러스터들의 중심점 $c^{(1)}(l), c^{(2)}(l), \cdots, c^{(K)}(l)$을 선택한다. 일반적으로, 주어진 표본 집합의 처음 K개의 표본을 임의로 선택한다. $c^{(j)}(l)$은 l번째 반복에서 j번째 클러스터 중심점의 값이다.

② l번째 반복단계에서 $i=1,2,\cdots,K, i\neq j$ 에 대하여, 다음의 관계를 이용하여 모든 샘플벡터 $x^{(q)}$를 K개의 클러스터에 분배한다.

$\|x^{(q)}-c^{(j)}(l)\|<\|x^{(q)}-c^{(i)}(l)\|$ 이면

$$x^{(q)}\in S^{(j)}(l) \tag{3.2.1}$$

여기서 $S^{(j)}(l)$는 l번째 반복에서 클러스터 j의 집합이다. 그리고 식 (3.2.1)을 만족하는 클러스터가 여러 개일 경우 임의로 정한다.

③ ②의 결과로부터 모든 $j=1,2,\cdots,K$에 대하여 $S^{(j)}(l)$에 포함된 모든 점들로부터 클러스터 중심까지의 거리의 제곱의 합을 최소화하는 새로운 클러스터 중심 $c^{(j)}(l+1)$을 계산한다. 즉, 다음과 같은 성능지표를 최소화하도록 새로운$c^{(j)}(l+1)$을 계산한다.

$$J_j=\sum_{x^{(q)}\in S_j(l)}\|x^{(q)}-c^{(j)}(l+1)\|^2, j=1,2,\cdots,K \tag{3.2.2}$$

이 성능지표를 최소화하는 $c^{(j)}(l+1)$은 단순히 $S^{(j)}(l)$의 평균이다. 그러므로 N_j가 $S^{(j)}(l)$에 속한 표본의 개수일 때, 새로운 클러스터 중심은 다음과 같이 주어진다.

$$c^{(j)}(l+1)=\frac{1}{N_j}\sum_{x^{(q)}\in S^{(j)}(l)}x^{(q)} \tag{3.2.3}$$

④ 모든 $j=1,2,\cdots,K$에 대하여 $c^{(j)}(l+1)=c^{(j)}(l)$를 만족하면, 알고리듬은 수렴하여 종료한다. 그렇지 않으면 ②로 간다.

$k-$means 클러스터링 알고리듬의 결과는 명시된 클러스터 중심의 개수, 초기 클러스터 중심의 선택, 표본(특징벡터)의 처리순서, 그리고 데이터의 기하학적 특성 등에 영향을 받는다. 이 알고리듬의 수렴성에 대한

증명은 없지만, 데이터가 서로 비교적 멀리 떨어진 특성을 나타낼 때 적당한 결과를 만들어 낸다. 대부분의 실용적인 경우에 이 알고리듬을 적용할 때는, 서로 다른 초기 클러스터 중심 뿐 만 아니라 다양한 K값에 대하여 실험해야 한다. 이러한 이유는 국부 최적해(local optima)에 빠지게 되기 때문에 전체 최적해(global optima)가 아니라는 말이다. 이를 해결하기 위해서는 메타휴리스틱 방법처럼 국부 최적해를 빠져나오기 위한 방법을 고안하여 사용할 수도 있을 것이다.

또 하나의 결정적인 단점은 알고리듬 처리시간이 아주 오래 걸린다는 것이다. 이것에 대한 이유는 두 가지가 있다. 그 하나는 이 알고리듬이 중차원 혹은 고차원의 공간에 적용되기 때문에, 가장 가까운 중심점을 찾는 계산시간이 아주 많이 걸린다는 것이다.(non-trivial problem) 그리고, 두 번째 이유로는 알고리듬의 많은 반복이 종료 조건을 만족할 때 까지 필요하다는 것이다.

3.2.2 셀 밀도를 이용한 개선된 k-means 알고리듬

알고리듬 처리시간을 줄이고, 미리 특징벡터들이 공간상에 분포되어져 있는 상태를 알아보고 초기해를 의미 있게 주기 위하여 특징벡터들의 각 차원의 최대 최소 값을 기준으로 특정의 수로 나누어서, 특징공간을 미리 분할하여 놓는다. 즉 특징공간을 셀, B_j로 나누어 놓고, 셀 안의 특징벡터의 수, $Count(B_j)$, 셀 안의 특징벡터들의 평균, $Ave(B_j)$,등을 미리 계산하여 전체 특징벡터들의 수를 셀 중심점의 수로 변환시켜서 클러스터링을 한다. 특징벡터의 수, Q에 의해서 바운드되어지고, 빈 셀들을 제외시키고 중심점들로만 클러스터링을 하면 그 계산량이 현저히 줄어든다. (중심점들의 수$\ll Q$) 즉 미리 클러스터링 하고자 하는 특징벡터들을 이렇게 전처리(preprocessing) 하는 것이 필요하다. 이는 반복할 때 마다 특징벡터 공간

상에서 특징벡터들이 변하는 것이 아니기 때문에 한번만 이루어진다. 또한 적당하지 않은 초기 클러스터 중심들의 선택에 기인한 좋지 않은 클러스터링을 막는 가장 간단한 방법은 그럴듯한 초기해를 주는 것이다. 이는 $Count(B_j) \geq \eta$ 인 경우, 그 셀의 중심점을 클러스터의 중심점으로 한다. 그리고 좋은 클러스터링을 하는 데에 영향을 미치고 있고, 적절한 클러스터의 개수 K의 선택이 반복에 의해서 찾아지기 때문에 적절한 초기 개수가 필요하다. 그러므로 적당한 η을 선택하여 조정한다.

좋은 클러스터링을 나타내어줄 기준이 필요하게 되는데 이를 유효성 측정(validity measure)이라 한다. 클러스터링은 클러스터들이 상대적으로 클러스터 중심을 기준으로 가까이 모여져 있고(compact), 클러스터 중심 간에는 적절히 떨어져 있는(well separated) 것이 좋다. K개의 클러스터 분산을 각각 σ_k^2로 나타내고, 이 값이 모든 클러스터들에 대하여 작다면 바람직하다고 말할 수 있고, 또한 $D_{\min}$은 모든 클러스터 중심 간의 거리 중에서 가장 적은 것을 나타낸다고 하면, 이는 적은 것보다는 큰 것이 좋다고 말할 수 있다. 클러스터링 유효성(Validity of Cluster)은 아래의 식 (3.2.4)을 통해서 얻을 수 있으며, 적은 값을 가질수록 좋은 특성을 갖는 클러스터들이라고 할 수 있다.

$$Validity\ of\ cluster = \sum_{k=1}^{K} \sigma_k^2 / D_{\min} \tag{3.2.4}$$

개선된 k-means 알고리듬은 다음과 같다

① K개의 클러스터들의 중심점 $c^{(1)}(l), c^{(2)}(l), \cdots, c^{(K)}(l)$을 선택한다. 여기에서 셀 밀도가 η보다 큰 셀들의 개수를 K라고 하고, 셀 내의 특징벡터들의 평균값인 셀의 중심 값을 클러스터들의 중심점으로 한다.

$$Count(B_j) \geq \eta \tag{3.2.5}$$

② K개의 클러스터들의 중심점 $c^{(1)}(l), c^{(2)}(l), \cdots, c^{(K)}(l)$들이 다음과 같이 너무 가까우면 2개의 클러스터들을 하나로 합친다. 두 개의 중심점 중에서 셀 밀도가 더 높은 쪽을 중심점으로 선택한다. $(K \leftarrow K-1)$

$$d(c^{(i)}(l), c^{(j)}(l)) < \epsilon \tag{3.2.6}$$

③ 최소 거리 할당 원칙에 의하여 모든 중심점, $Ave(B_j)$를 K개의 클러스터에 분배한다.

④ ③의 결과로부터 모든 $j = 1, 2, \cdots, K$에 대하여 $S^{(j)}(l)$에 포함된 모든 점들로부터 클러스터 중심까지의 거리의 제곱의 합을 최소화하는 새로운 클러스터 중심 $c^{(j)}(l+1)$을 계산한다. 계산 할 때 셀 밀도를 활용하여 가중치가 있는 중심점들로 만들어서 평균을 구한다.

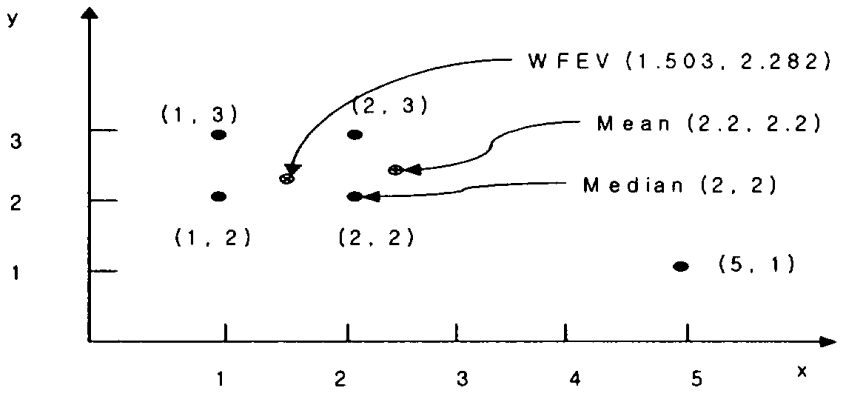
⑤ 모든 $j = 1, 2, \cdots, K$에 대하여 다음의 식 (3.2.7)이나 식 (3.2.8)을 만족하거나 식 (3.2.4)의 유효성(validity of cluster)이 증가하면 알고리듬은 수렴하여 종료한다. 그렇지 않으면 가장 가까운 2개의 클러스터 중심점을 하나로 합치고, ③으로 간다.

$$c^{(j)}(l+1) = c^{(j)}(l) \tag{3.2.7}$$

$$\left| c^{(j)}(l+1) - c^{(j)}(l) \right| \leq \rho \tag{3.2.8}$$

3.2.3 퍼지 k-means 알고리듬

클러스터링을 하는 데에 또 다른 문제는 중심에서 멀리 떨어져서, 경계에 위치한 특이한 값(outlier)에 의해서 심하게 영향을 받는다는 것이다. 이 특이한 값에 의해서 영향을 받는 평균보다는 가우시안을 이용한 식 (3.3.3)에 의한 가중치를 적용한 특징 벡터들의 가중 평균을 계산한다. 아



그림<3.2.1> 가중 퍼지평균, 평균과 중간 값의 비교

래의 그림<3.2.1>은 가중 퍼지 평균과 평균, 중간 값(median)의 관계를 나타낸 것이다. 가중 퍼지 평균은 평균, 중간 값(median) 또는 퍼지 평균 보다는 집합을 대표할 수 있는 값이다. 이때, 가중치의 적용 방법은 이상적인 중심으로부터 먼 곳에 있는 벡터보다는 가까운 벡터에 높은 가중치를 부여하는 것이다. 또한 아래 그림은 아주 간단한 2차원의 예를 가지고 가중 퍼지 평균, 평균과 중간 값의 차이를 보여준다. (1, 2), (2, 2), (1, 3), (2, 3)과 (5, 1)의 벡터들이 있다. 특이한 값(outlier) (5, 1)은 아주 강하게 평균 (2.2, 2.2)과 중간 값 (2, 2)에 영향을 준다. 그러나 가중 퍼지 평균 (1.503, 2.282)는 특이한 값에 의해서 y축 방향으로는 약간 적게 영향을 받고, x축 방향으로는 훨씬 더 적게 영향을 받는다.

가중 퍼지 평균을 구하기 위해서 집합 $\{x_1, x_2, \cdots, x_q\}$의 산술 평균 $\mu^{(0)}$을 초기 중심으로 하고, 초기 분산 $(\sigma^2)^{(0)}$을 현재 고려되고 있는 원형(prototype)의 정규(Gaussian)로 표준화하기 위한 크기 파라미터(spread parameter)로 사용하여 초기 가중치를 구한다. 이후 아래의 식 (3.3.1), 식 (3.3.2), 식 (3.3.3)을 반복적으로 사용하여 새로운 분산과 가중 평균 및 가중치를 구하여 나간다. 결국 $\mu = \mu^{(\infty)}$는 어느 한 값으로 수렴하게 되고 이것을 가중 퍼지 평균이라고 한다.

$$\mu^{(r+1)} = \sum_{(q=1,Q)} \alpha_q^{(r)} x_q \tag{3.3.2}$$

$$(\sigma^2)^{(r+1)} = \sum_{(q=1,Q)} \alpha_q^{(r)} (x_q - \mu^{(r)})^2 \tag{3.3.2}$$

$$\alpha_q^{(r)} = \frac{exp[-(x_q - \mu^{(r)})^2/(2\sigma^2)^{(r)}]}{\sum_{(m=1,Q)} exp[-(x_m - \mu^{(r)})^2/(2\sigma^2)^{(r)}]} \tag{3.3.3}$$

개선된 퍼지 k-means 알고리듬은 다음과 같다.

① K개의 클러스터들의 중심점 $c^{(1)}(l), c^{(2)}(l), \cdots, c^{(K)}(l)$을 선택한다.

② K개의 클러스터들의 중심점 $c^{(1)}(l), c^{(2)}(l), \cdots, c^{(K)}(l)$들이 너무 가까우면 2개의 클러스터들을 하나로 합친다. $(K \leftarrow K-1)$

③ 최소 거리 할당 원칙에 의하여 모든 특징벡터 $x^{(p)}$를 K개의 클러스터에 분배한다.

④ ③의 결과로부터 모든 $j=1, 2, \cdots, K$에 대하여 $S^{(j)}(l)$에 포함된 모든 점들로부터 가중 퍼지 평균을 구하는 식 (3.3.1)을 이용하여 새로운 클러스터 중심 $c^{(j)}(l+1)$을 계산한다.

⑤ 가중 퍼지 평균에 대해서 최소 거리 할당의 원칙에 의하여 $x^{(p)}$를 K개의 클러스터에 분배한다.

⑥ 첫 번째 패스이거나 어떤 가중 퍼지 평균이 변하면 ④로 간다.

⑦ 첫 번째 패스가 아니고 유효성이 증가하면 알고리듬은 종료한다. 그

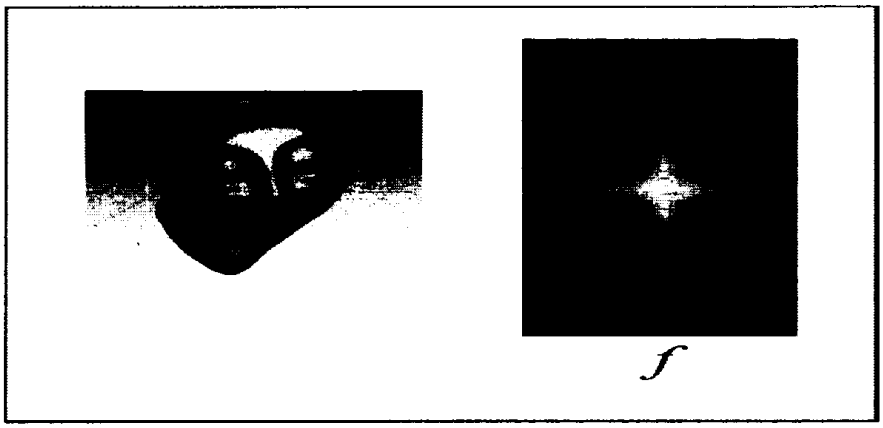
렇지 않으면 가장 가까운 2개의 클러스터 중심점을 하나로 합치고, ④로 간다.

K개의 초기 중심점들을 설정한 다음, 가장 근접한 두 중심의 거리를 나타내는 식 (3.2.6)의 $\epsilon = 1/(K)^{1/N}$로 구하여 ϵ보다 작은 경우 두 중심을 평균값으로 하여 중심을 정하고, 특징 벡터들을 최소 거리 할당 원칙에 따라서 가장 가까운 중심에 할당한다. 클러스터에 할당된 벡터들의 산술 평균값을 새로운 중심점으로 하고, 다시 특징 벡터들을 새로운 중심점에 할당한다. 이렇게 몇 번의 반복 과정을 수행한다. 반복 과정을 통해 얻어진 클러스터의 벡터들에 대해서 산술평균 $\mu^{(r)}$을 구하여, 식 (3.3.2)와 식 (3.3.3)에 의해서 분산과 퍼지 가중치를 구하고, 식 (3.3.1)에 의해서 각 클러스터 K의 가중 퍼지 평균을 수정한 후, 특징 벡터를 다시 가중 퍼지 중심 $\{c^{(k)}\}$에 할당한다. 다시 할당된 특징 벡터들에 대해서 식 (3.3.2), 식 (3.3.3)과 식 (3.3.1)을 적용하여 얻어진 새로운 가중 퍼지 중심에 특징 벡터를 할당한다. 이렇게 몇 번의 반복 과정을 실시한다. 이러한 몇 번의 반복 과정은 퍼지 중심을 어느 한 값으로 수렴하게 만든다. 클러스터링 이후, 비교적 근접한 클러스터를 합병시킬 수 있다.

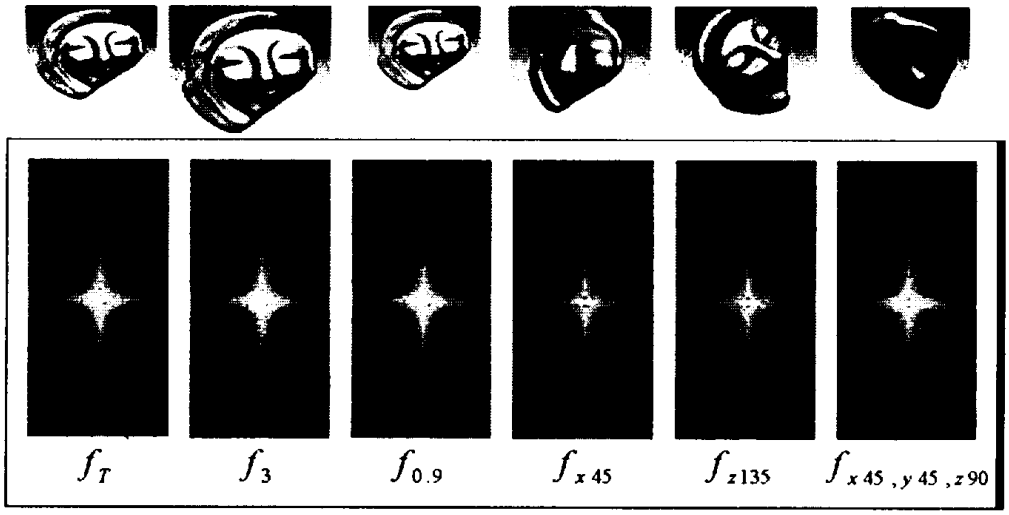
제 4 장 알고리즘에 관한 실험 및 결과

4.1 3차원 불변 특징벡터 추출 알고리즘에 관한 성능

개발된 기술로 하나의 3D 객체에 대하여 3D 불변특징벡터를 추출하였고 72×36영상으로 표현하여 동일 객체에 대한 변형 불변 여부와 상이한 객체들 간의 변별력 유무에 관한 실험을 실시하고 특징벡터간 차이의 잡음대비율인 PSNR(Peak Signal to Noise Ratio)을 사용하여 분석한 결과 동일 객체에 대해서는 원본과 변환된 객체와의 PSNR이 34이상으로 동일한 것으로 나타났으며 상이한 형태의 다른 객체간의 PSNR은 모두 20이하로 다른 형상으로 판단되는 것으로 나타나 개발된 특징벡터가 동일 객체에 대해서는 다양한 변화에 관계없이 유사한 값을 나타내고 이종의 3차원 객체에 대해서는 다른 값을 나타내었다.



그림<4.1.1> 원래의 실험 객체와 특징 벡터



그림<4.1.2> 변형시킨 객체와 특징 벡터

실험결과 동일 객체에 대해서는 원본과 변환된 객체간의 PSNR이 모두 34이상으로 동일한 것으로 나타났다. 위치 이동과 크기 변화 후의 특징 벡터들의 PSNR은 70이상으로 회전 변화 후의 특징 벡터들의 PSNR 34 보다 높게 나타나 개발된 특징벡터가 위치이동과 크기변화에 대한 불변성이 더 강함을 알 수 있다. 이는 위치 이동시에는 객체의 폐곡면과 중심점과의 상호 관계가 유지되었고 크기 변화시에는 내측거리값이 정규화되었기때문이다. 특히 위치 이동시에는 단순한 위치 이동으로 인하여 변형전•후의 특징 벡터가 동일하다는 것을 알 수 있다.

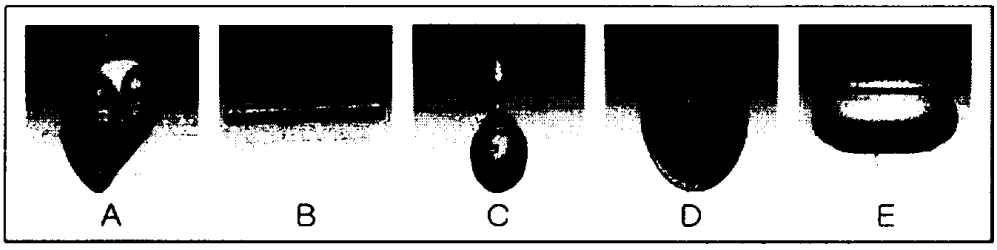
회전 변화시에는 회전 전의 중심축과 회전 후의 중심축이 미세한 차이를 보여 회전 전•후의 내측거리값이 상이했기 때문에 위치 이동과 크기 변화 후의 특징 벡터들에 비해 상대적으로 PSNR수치가 낮게 나타났으나 일반적으로 PSNR이 20이상일 경우 동일한 벡터로 간주하는 기준으로 볼 때 개발된 특징벡터는 위치, 크기, 회전 변환에 불변인 특성을 지닌 것으로 판단된다.

표 4.1.1 동일 객체에 대한 PSNR 수치

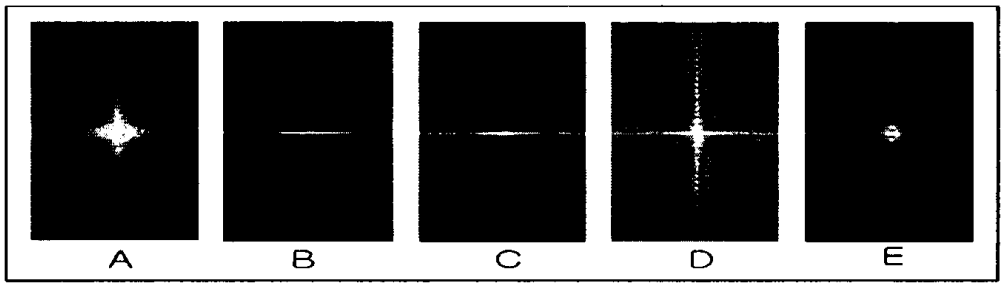
구분	f_T	f_3	$f_{0.9}$	f_{x45}	f_{z135}	$f_{x45,y45,z90}$
f	∞	74.4856	76.2466	34.5754	34.4851	38.2512
f_T	∞	74.4856	76.2466	34.5754	34.4851	38.2512
f_3	·	∞	72.2672	34.5758	34.4853	38.2512
$f_{0.9}$	·	·	∞	34.5745	34.4845	38.2507
f_{x45}	·	·	·	∞	38.6865	35.7919
f_{z135}	·	·	·	·	∞	36.0059

상이한 객체들 간의 변별력그림<4.1.3>과 같은 상이한 객체 5개에 대하여 개발된 기술을 적용하여 그림<4.1.4>와 같은 각각의 특징 벡터를 추출하였고 추출된 특징 벡터들이 각 객체 고유 특징 벡터인지의 여부를 판단하기 위해 PSNR을 계산하여 표 5.1.2와 같은 결과를 얻었다.

직관적으로나 PSNR 수치상으로도 각 객체의 특징 벡터들은 다른 객체의 특징 벡터와 상이함을 보여주고 있어 각기 고유한 특징 벡터를 나타내고 있다는 것을 발견할 수 있다.



그림<4.1.3> 실험 객체들



그림<4.1.4> 실험 객체들의 특징 벡터

표 4.1.2 상이한 객체들간의 PSNR 수치

구분	B	C	D	E
A	11.3796	15.0254	18.4678	13.2018
B	∞	16.8183	10.2049	18.4398
C	·	∞	13.2315	18.8099
D	·	·	∞	11.5524

4.2 2차원기반 불변 특징벡터 추출알고리즘에 관한 성능

2D기반 3D 특징벡터 추출기술에서 중요한 요소는 3D 객체를 대표하는 2D 이미지의 특징벡터를 빠르게 그리고 적절하게 추출하는가에 대한 것이다. 개발된 기술은 3D 객체에서 임의각도로 투영된 2D 이미지에 대해 비슷한 것끼리 군집화하는 클러스터링 방법을 사용한 것이다. 클러스터링에 있어서 중요한 것은 비슷한 개체끼리 어떻게 잘 묶이는 가에 대한 평가이다. 그리고 시스템의 CPU 능력을 고려할 때 수행시간에 대한 평가도 필요하다. 개발된 4가지 클러스터링 방법에 대해 3차원 객체를 가지고 수행시간과 클러스터링의 유효성을 평가한 결과 퍼지 k-mean방법이 유효성과 수행시간이 대체적으로 우수하게 나타나 적합한 클러스터링 방법으로 나타났다.

4.2.1 실험의 구성

본 과제에서 사용된 3차원 데이터는 3차원객체를 3차원 레이저 스캔하여 레피드폼 프로그램을 사용 dxf 파일로 변환된 것들을 사용하였고, 실험은 Zernike 모멘트 연산자를 이용하여 추출된 특징 벡터에 기본적인 k-means 알고리듬, 셀 밀도를 이용한 개선된 k-means 알고리듬과 퍼지 k-means 알고리듬 그리고 셀 밀도를 이용한 개선된 퍼지 k-means 알고리듬의 4가지 클러스터링 방법을 사용하여 그 결과를 수행 시간과 유효성에 대해서 비교하였다.

4.2.2 실험 결과

특징벡터들의 기하학적 특성과 입력 순서에 아주 많은 영향을 받기 때문에 특징벡터에 따라서 다른 결과를 나타낸다. 특히 k-means 알고리듬과 개선된k-means 알고리듬 방법은 결과가 아주 극한적으로 변화하기 때문에 어느 방법이 낫다고 말하기가 힘들다. 그러나 이 방법들과는 달리 퍼지 k-means 알고리듬은 안정적이고 좋은 유효성 결과를 보인다. 개선된 퍼지 k-means 알고리듬은 가장 좋은 유효성을 보이지만 안정적이지 못한 결과를 보인다.

표 4.2.1 클러스터링 방법에 따른 결과 비교

	k-means		개선된k-means		퍼지 k-means		개선된 퍼지 k-means	
	시간	Validity	시간	Validity	시간	Validity	시간	validity
1	18.697	4.2607	0.641	22.1544	4.677	5.2234	20.85	2.6805
2	0.421	10.616	12.298	7.9688	4.707	1.36	0.38	8.6161

3	0.38	37.2643	0.4	47.7003	4.676	5.5988	21.0709	1.1508
4	12.368	23.9333	0.721	25.6669	4.998	3.4337	21.101	2.4028
5	0.461	24.9003	0.161	12.7721	5.538	1.5628	15.893	0.7412
평균	6.4654	20.19492	2.8442	23.2525	4.9192	3.43574	15.859	3.3183

개선된 k-means 알고리듬의 수행시간이 적은 것은 셀의 밀도가 0보다 큰 경우에 그 중심점을 초기 클러스터의 중심점으로 시작했기 때문에 반복 횟수가 몇 번 지난 다음에 수렴했기 때문이다. 그러나 클러스터의 수를 조정하면 다른 결과를 나타내고 있다.

개선된 퍼지 k-means 알고리듬에서 초기해를 셀 밀도를 이용하여 구하고, 똑같이 수행 하였다. 그 결과 유효성이 좋아지고 수행시간은 많이 늘어났다. 수행시간이 늘어났다는 것은 반복 횟수가 많아졌다는 것이고, 클러스터 개수가 몇 개 되지 않는다는 말이다. 또한 유효성이 다른 방법에 비해서 좋게 나왔지만 이는 클러스터의 거리가 많이 떨어졌다는 이야기이다. 다른 말로 하면 유효성이 좋다고 클러스터링이 잘됐다고 절대적으로는 말할 수가 없다. 그 이유는 유효성 계산에서 클러스터 중심 간의 최소 거리가 상황에 따라서 그 의미가 달라질 수 있기 때문이다.

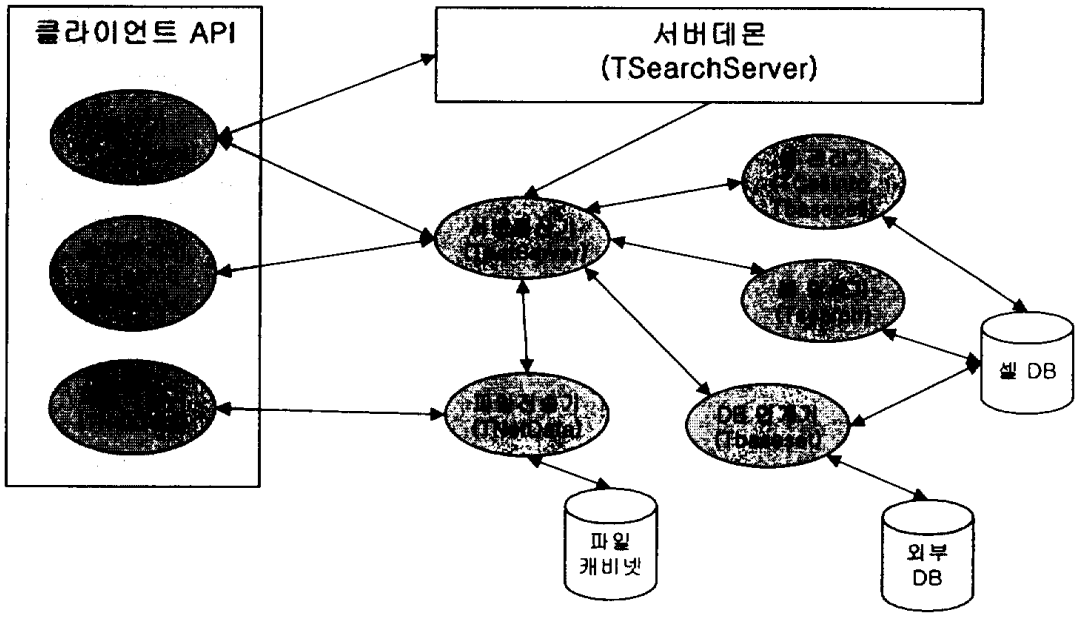
4.3 개발된 알고리즘을 이용한 3차원 형상검색시스템 개발

여러 개의 벡터 중 어떤 하나의 주어진 벡터와 가장 유사한 벡터를 찾아내는 일반적인 방법은 모든 벡터와 주어진 벡터 간의 유사도를 계산해

가장 작은 유사도를 가진 벡터를 찾아내는 방법을 이용한다. 그러나 이 방법은 검색 대상 벡터의 수가 많을 경우 너무 심각한 처리부하를 야기 시켜 검색효율이 나빠진다. 이러한 문제를 해결하기 위해 본 연구에서는 제3장에서 개발된 알고리즘을 이용하여 데이터베이스에 저장된 벡터레코드 수가 증가하더라도 질의 벡터와 유사한 벡터를 찾아 주는 데 필요한 시간이 선형으로 증가하지 않는 벡터 매칭 엔진을 개발하였다. 그리고 개발된 벡터매칭 엔진을 사용하여 3차원 객체 검색을 위한 시범 시스템을 구축하였다.

4.3.1 벡터매칭 엔진기

벡터매칭 엔진은 그림<4.3.1>과 같이 다수의 클라이언트에서 주어진 명령을 해석하여 적절한 기능을 수행하는 서버데몬(TsearchSever), 클라이언트와 서버간의 통신을 처리하고 실질적인 기능을 수행하는 서버통



그림<4.3.1> 백터 매칭엔진 구성도

신기(TNetServer), 클라이언트와 서버간의 원본 자료를 전송하는 파일전송기(TNetData), 연결될 외부 데이터베이스와의 관계를 관리하는 외부

DB연계기(TBaseSet, TRecordSet), 등록 또는 검색될 벡터의 셀 인덱스를 관리하는 셀 관리기 (TCellInfo, TBaseCell), 인접 셀을 찾아주는 셀 탐색기(TSearch), 셀을 관리하고 벡터를 매칭하기 위한 정보가 들어 있는 셀 DB, 등록할 원본데이터를 저장할 원본 캐비넷 그리고 클라이언트에서 매칭엔진 호출에 사용할 클라이언트 API (TnetClient)로 구성되어 있다.

서버데몬(TSearchServer)은 하나의 클라이언트와 접속이 되면 통신과 기능을 수행할 서버통신객체(TNetServer)를 생성하여 클라이언트와의 통신 및 명령처리를 담당하게 한다. 이 방식은 동시에 여러개의 클라이언트에 대한 서비스의 독립적 수행을 가능하게 해준다. TNetServer Instance는 통신소켓의 단절이나 서버데몬의 명령에 의해 소멸된다.

서버통신기(TNetServer)는 클라이언트와 소켓을 통해 통신하며 벡터 등록, 검색, 삭제, 환경설정 등의 명령 수행과 처리결과 통보 그리고 벡터 데이터를 송수신하는 기능을 수행한다. 서버통신기는 실질적으로 명령을 수행하면서 셀관리기, 셀탐색기, 외부 DB 연계기, 파일전송기 등에 명령을 전달하여 주어진 일을 수행하고 그 결과를 클라이언트에 전달한다. 서버통신기는 서버데몬에 의해 생성되며 callback 함수에 의해 각 처리기 로부터 메시지를 전달 받는 구조로 되어 있다. 서버통신기는 클라이언트에서 서버로 접속 시 생성되고 접속이 종료되면 데몬에 의해 소멸된다. 불의의 오류에 의해 클라이언트에서 비정상 종료가 발생시 서버통신기는 연결된 통신 소켓의 닫기 오류에 의해 스스로 소멸되도록 되어 있다.

셀관리기(TCellbase, TCellInfo)는 전달된 벡터 데이터를 개발된 셀 인덱싱 기술에 의해 셀ID로 변환 생성하고 이에 필요한 정보를 관리하는 기능을 수행한다. 셀 ID를 생성하기위해서는 셀의 너비, 각 셀에 존재하는 개체 수, 셀내 개체분포 정보 등이 필요하다. 이러한 정보는 DB에 벡터가 등록 될 때 마다 갱신이 이루어진다. 그리고 벡터의 감차를 위한 정보로 주요인 함수, 고유벡터, 고유치 등이 필요하다. 이런 정보는 벡터의 누적으

로 기존의 감차정보의 효용성이 떨어지면 감차분석을 통해 새로이 갱신이 가능하다. 셀 관리기는 셀 ID를 생성하기위한 기본환경 정보를 관리하는 기능과 셀에 관련된 통계정보를 추출하는 기능 그리고 주어진 벡터에 대한 셀 ID 생성 기능 등을 수행한다.

셀 탐색기(TSearch)는 개발된 인접 셀 탐색 알고리즘을 이용해 질의된 벡터와 인접한 셀을 찾고 그 안에서 유사도를 계산하여 인접한 순서대로 벡터를 찾아내는 기능을 수행한다.

파일전송기(TNetData)는 원본데이터를 서버에 등록하거나 검색된 원본 데이터를 클라이언트에 전송하고 저장하는 기능을 수행한다. 이를 위해 FTP(File Transfer Protocol)을 사용하여 구현하였다.

클라이언트 API의 주요 구성요소는 등록, 검색, 삭제 등의 명령어를 서버에 전달하는 통신기(TnetClient)와 서버의 처리 결과메시지를 받아 결과 처리를 수행하는 결과처리기(TSearchResult) 그리고 파일 전송기이다. 개발된 벡터 매칭 엔진의 주요 클래스와 기능은 표 4.3.1과 같다.

표 4.3.1 벡터 매칭 엔진의 주요 클래스와 기능

Class 명	주요기능
TSearchServer	서버데몬, 서버통신기 생성 및 소멸
TSearch	질의된 벡터와 인접한 셀을 탐색
TFVData	감차를 위한 특성벡터 분석 수행
TCellInfo	셀의 각 종 통계 및 정보 처리
TbaseSet	외부 DB에 접근 기능 수행
TRecordSet	탐색된 셀 내의 레코드를 관리
TbaseReduce	데이터베이스 내의 감차 테이블 관리
TReduceData	감차된 데이터 생성, 변경 등의 정보처리
TbaseCell	셀 ID 생성 및 관리 기능 수행
TCellCount	감차된 셀의 탐색에 관련된 정보관리
TbaseCallback	각 종 콜백을 처리하는 결과처리기
TNetServer	클라이언트와 통신을 처리
TNetClient	서버와의 통신을 처리
TNetData	FTP기반의 파일전송 저장 기능
TSearchResult	서버로부터의 결과 메시지 처리
TSocket	TCP/IP기반의 소켓 통신 담당

4.3.2 시범 시스템

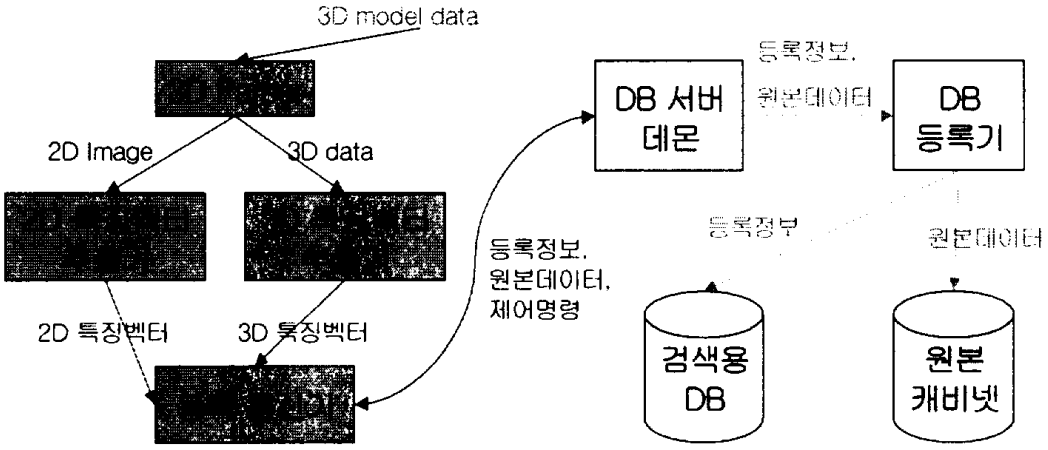
시범 시스템은 CAD 표준 저장 형태인 DXF 파일 구조로 저장된 3차원 객체로부터 3차원 형상불변벡터와 2차원 대표영상 특징벡터를 추출하여 이를 데이터베이스에 저장하고 그로부터 임의의 3차원 객체나 2차원 영상에 대해 검색하는 시스템으로 클라이언트와 개발된 매칭 엔진으로 구성되어 있다.

그림<4.3.2>는 3차원 객체를 등록하는 과정에 관한 시스템의 자료

흐름도로 DXF 형식의 3차원 객체를 분석하여 필요한 2차원 영상정보와 3차원 모델 정보를 추출 변환하여주는 3D Parser, 개발된 3D 형상특징벡터를 추출하는 3D 특징벡터 추출기, 3D 객체의 대표성 있는 2D 영상을 추출하고 그 특징벡터를 추출하는 2D 특징벡터 추출기 그리고 앞에서 개발된 벡터 매칭 엔진으로 구성되어 있다. 개발된 시스템에 의해 3차원 객체가 검색용 데이터베이스에 등록되는 과정은 다음과 같다.

○ 3D PARSER에 의해 입력된 3차원 객체의 매쉬 데이터는 투사각 중심으로 매쉬에 대한 정보의 접근이 용이도록 트리를 이용하여 구조화된다.

○ 2D 특징벡터 추출기는 3D PASER의 도움을 받아 다양한 투사각도의 2차원 영상 이미지를 전달받아 이에 대한 2D 특징벡터를 추출하고 개



그림<4.3.2> 3차원 객체 등록 흐름도

발된 기술을 이용해 이로부터 대표 특징벡터들을 추출한다.

○ 3D 특징벡터 추출기는 3D PASER의 도움을 받아 3차원 객체를 정규화한 후 개발된 기술을 이용해 3차원 형상 불변 특징벡터들을 추출한다.

○ 등록통신기는 추출된 3차원 형상불변 특징벡터와 2차원 대표 특징벡

터와 3차원 객체 데이터를 개발된 매칭엔진의 클라이언트용 API를 이용해 검색용 DB에 등록한다.

그림<4.3.3>은 3차원 객체를 검색하는 과정에 대한 시스템의 자료 흐름도로 DXF 형식의 3차원 객체를 분석하여 3차원 모델 정보를 추출하여주는 3D Parser, 개발된 3D 형상특징벡터를 추출하는 3D 특징벡터 추출기, 3D 객체의 대표성 있는 2D 영상을 추출하고 그 특징벡터를 추출하는 2D 특징벡터 추출기 그리고 벡터 매칭 엔진으로 구성되어 있다. 개발된 시스템에 의해 3차원 객체가 검색용 데이터베이스에 검색되는 과정은 다음과 같다.

○ 3D PARSER에 의해 질의된 3차원 객체의 매쉬 데이터는 투사각 중심으로 매쉬에 대한 정보의 접근이 용이도록 옥트리를 이용하여 구조화된다.

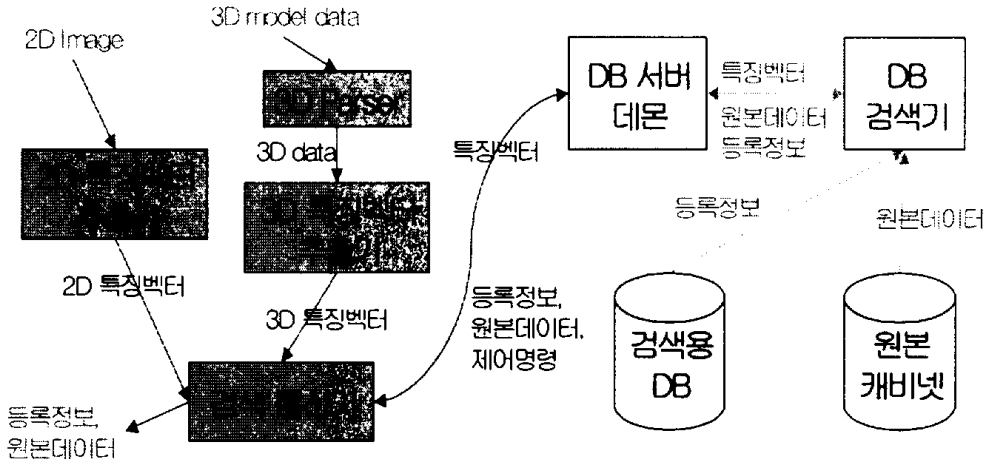
○ 3D 특징벡터 추출기는 3D PASER의 도움을 받아 3차원 객체를 정규화한 후 개발된 기술을 이용해 3차원 형상 불변 특징벡터들을 추출한다.

○ 2D 특징벡터 추출기는 질의된 다양한 형식(JPEG, BMP, TIFF 등)의 2차원 영상 이미지에 대한 2D 특징벡터를 추출한다.

○ 등록통신기는 추출된 3차원 형상불변 특징벡터나 2차원 대표 특징벡터를 개발된 매칭 엔진의 클라이언트용 API를 이용해 벡터 매칭 엔진 서버에 질의하여 그 결과를 보여준다.

그림<4.3.4>는 개발된 시범 시스템의 클라이언트용 사용자화면이다. GUI의 구성은 탭을 사용하여 검색과 등록 화면을 구성하였다. 검색 탭의 상단 우측에 있는 아이콘은 검색수행, 앞검색 결과보기, 다음검색 결과보

기로 구성되어 있으며 검색을 위한 2D 또는 3D 데이터는 파일메뉴의 열기나 파일오픈 아이콘을 사용한다. 등록 탭의 우측 상단에는 파일오픈 아

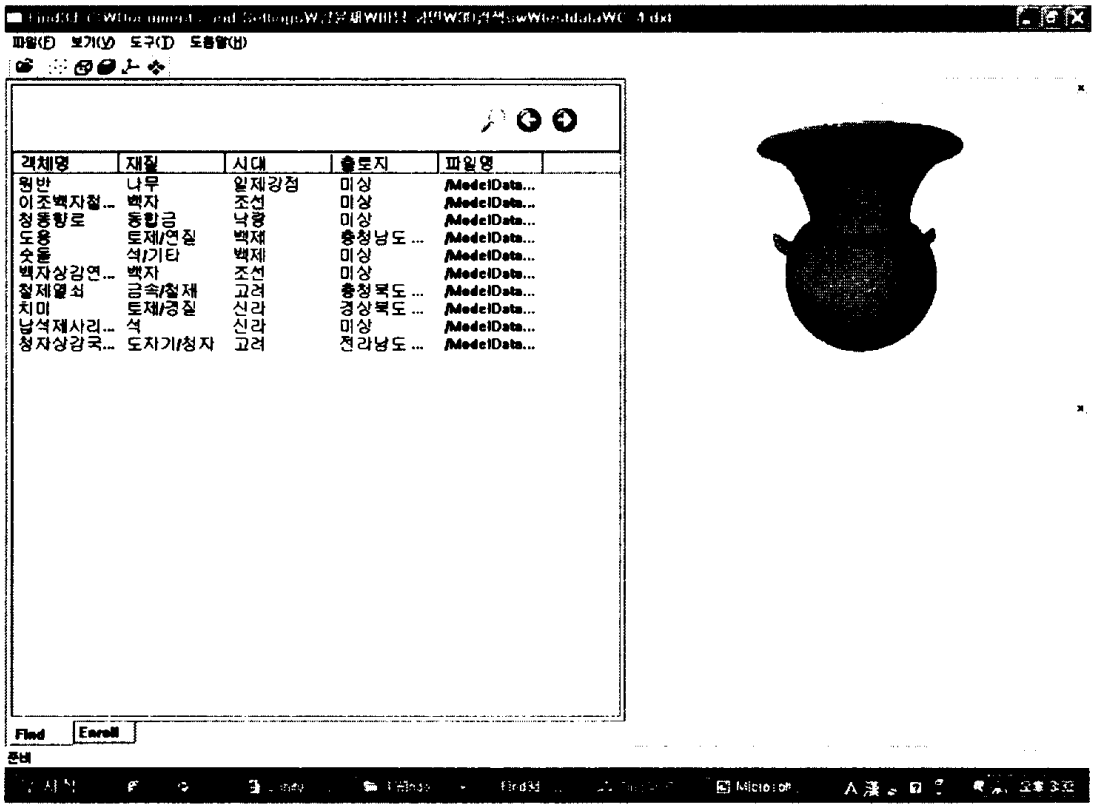


그림<4.3.3> 3차원 객체 검색 흐름도

이콘과 등록 아이콘으로 구성되어 있다.

화면의 우측 상단의 영상은 등록 대상 또는 검색 대상 영상이 나타나며 3차원 객체에 대해서는 회전, 이동, 확대 축소 등이 가능하다. 우측 하단의 영상은 검색된 결과에서 선택된 객체의 3차원 객체영상을 표시한다. 이 역시 회전, 이동, 확대 축소 등이 가능하다. 좌측의 List는 검색된 객체들을 유사도가 높은 순서대로 표시한다. 표시된 결과 중 원하는 객체를 더블클릭하면 3차원 객체가 서버로부터 전송되어 우측 하단의 영상 박스에 표시된다.

개발된 3차원 객체 형상불변벡터 추출을 위해 DXF 형식의 3차원 객체를 해석하고 분석하여 필요한 정보를 추출하는 TMLparser 클래스의 구성도와 3차원 형상불변 특징벡터를 추출하는 FVext 라이브러리를 구성하고 있는 객체들의 체계도, 그리고 각 클래스의 기능에 대한 자세한 설명은 부록에 설명되어 있다.



<그림 4.3.4> 시범 시스템의 검색화면 예

제 5 장 결 론

디지털 컨텐츠의 발전방향은 문자에서 멀티미디어, 그리고 가상현실로 발전해가고 있다. 이러한 디지털 컨텐츠의 내용을 효과적으로 검색하고 재활용하기 위한 기반기술인 내용기반 검색기술은 정지영상은 물론 동영상 및 가상현실의 기본 구성요소까지 포함해야 한다. 즉 3차원 영상에 대한 검색이 가능해야 한다.

본 논문에서는 3차원 형상을 인식하고 검색하는데 있어서 특징벡터 추출 알고리즘을 제시하였고 임의의 객체에 대해 추출된 특징벡터는 그 객체에만 나타나는 고유의 특징 벡터라고 할 수 있으며 이 특징 벡터는 물체의 위치 이동, 크기 변화, 회전 변화에 불변이고 다른 객체와의 변별력을 가진 것으로 판단되므로 이는 기하학적인 변형에 불변인 객체의 형상적 특징을 대표하는 3차원 불변특징벡터를 추출하는데 유효성을 크게 보여 주었다.

3차원 형상 데이타의 메타검색의 대상은 3D 데이터에 대한 2D 클러스터링된 특징벡터와 3D 특징벡터로 표현되며, 기존의 Image의 Database가 문자 기반인 것에 반해, 3D 형상 검색을 위한 이미지 내용 기반의 Database이며 또한, 검색할 수 있는 Meta Database를 시스템 내에서 자동으로 구축이 가능하다.

이러한 알고리즘을 이용하여 3차원 디지털 컨텐츠의 내용 기반의 검색기술과 응용시스템을 구현하였다. 막대한 정보 계산량을 효과적인 알고리즘으로 컴퓨터가 자동적으로 컨텐츠의 특징을 추출하고 색인, 저장하며 그 특징을 이용한 유사 검색을 할 수 있는 효과적인 3차원 디지털 컨텐츠 관리시스템은 관련 분야의 기술과 산업 발전에 일익을 기여할 것이다.

향후 3차원의 수많은 형상들의 다양한 속성들을 표현하는 특징벡터의 추출 방법이 그 정밀도와 처리속도가 현실을 고려한 효율적 방안이 추구되어야 할 것이며 이들을 서로 복합하여 질의를 수행할 수 있는 기술 및 시스템에 대한 실제적 연구가 계속 진행되어야 하겠다. 또한 폐곡면이 아닌 개방형상을 포함한 일반적인 형상객체에 대한 속성들을 표현하는 기술 및 시스템 연구를 통해 다차원 형상에 대한 효과적인 연구가 가능할 것이다.

참 고 문 헌

[Allen1998] R. V. H. Allen T. Craig, Introduction to Mathmatical Statistics, Prentice Hall International, Inc. 1998

[Arbter1990] K. Arbter, W.E. Snyder, H. Burkhardt, G Herzin-ger, "Application of affine-invariant Fourier descriptor to recognition of 3-D objests", IEEE Trans. On Pattern Analysis and Machine Intelligence, 12(1990) 640-647.

[Berchtold1998] S. Berchtold, C. Behm, and H. P. Kriegel, "The Pyramid-Technique:Towards Breaking the Curse of Dimensionality", ACM SIGMOD, pp.142-153, June 1998

[Berman1997] A.P.Berman and L. G. Shapiro, "Efficient image retrieval with multiple distance measures,"in Proc. SPIE Conf. Storage and Retrieval for Image and Video Databases, Vol. 3022, pp. 12-21, Fed. 1997

[Canterakis1999] N. Canterakis, "3D Zernike moments and Zernike affine invariants for 3D image analysis and recognition," Proc. 11th Intern. Conf. on Image Analysis, Kangerlussuaq, Greenland, June 1999.

[Chen1999] M. Chen, S. Wang, "Fuzzy clustering analysis for optimizing membership functions", Fuzzy Sets and Systems 103(1999) 239-254

[Cox1994] T. F. Cox and M. A.A. Cox, Multidimensional

Scaling, Chapman & Hall, London SE1 8Hn, Uk, 1994

[Ericsson1999] A. Ericsson, and D. Weber, " Towards 3-D Face Recognition. IEEE Africon, Vol.1, pp401-4-6, 1999

[Faloutsos1995] Christos Faloutsos. Fast Searching by Content in Multimedia Databases. Data Engineering, 18(4), 1995.

[Gonzales] R. C. Gonzales and R. E. Woods. Digital Image Processing. Addison-Wesley Publishing Company

[Gose] E. Gose, R. Johnsonbraugh and S. Jost, Pattern Recognition and Image Analysis, Prentice Hall

[Gudivada1995] Venkat N. Gudivada and Vija V. Raghavan. Content-Based Image Retrieval Systems. IEEE Computer, 28(9), 1995.

[Kelly1995] P.M. Kelly, T.M. Cannon and D.R. Hush. Query by image example: the CANDID approach. Proc. SPIE Storage and Retrieval for image and Video Databases III, 2420:238-248, 1995.

[Khotanzad1990] A. Khotanzad, Y. Hong, "Invariant Image Recognition by Zernike moments", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol., 12 No. 5 May 1990

[Kim1994] W. Y. Kim, Po Yuan, "A practical pattern recognition system for translation, scale and rotation

invariance," Proc. Of the IEEE International Conf. on Computer Wision and Pattern Recognition June 1994, pp.391-396

[Lee1997] Dong Ho Lee, Young Jun Song, Hyoung-Joo Kim. SCARLET: Design and Implementation of Content-based Image Retrieval System using Wavelet Transform. Journal of KISS,3(4),1997.

[Looney2002] C. G. Looney, "Interactive clustering and merging with a new fuzzy expected value,"Pattern Recognition, Volume 35, Issue 11, November 2002, Pages 2413-2423

[Lotlikar2000] R. Lotlikar, R. Kothari, "Adaptive linear dimensionalith reduction for classification", Pattern Recognition 33,pp185-194, 2000

[Mico1996] L. Mico, J. Oncina, R, C. Carrasco, "A fast branch & bound nearest neighbour classifier in metric spases" Pattern Recognition Letters 17, pp731-739, 1996

[Nadler1993] M. Nadler, E. P. Smith, Pattern Recognition Engineering, John Wiley & Sons Inc., 1993

[Ng1994] R. T. Ng and J. Han, "Efficient and Effective Clustering Methods for Spatial Data Mining", Proc. Of VLDB, 1994

[Niblack1993] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovis, P. Yanker, C. Faloutsos, G. Taubin. The QBIC project: Querying image by content using color,

texture, and shape. Proceedings SPIE Storage and Retrieval for Image and Video Databases, pages 173-187, February 1993.

[Omachi2000] S. Omachi and H. Aso, "A Fast Algorithm for a k-NN Classifier Based on Branch and Bound Method and Computational Quantity Estimation",Systems and Computers in Japan, vol.31, no.6, pp.1-9, May 2000

[Pal1995] N. R. Pal, J. C. Bezdez, "On cluster validity for Fuzzy c-means model," IEEE Trans. Fuzzy Systems 3. 1995.

[Paquet2000] E. Paquet, A. Murching, T. Naveen, A. Tabatabai, and M. Rioux, "Description of shape information for 2-D and 3-D objects,"Signal Processing: Image Communication, 16:103.122,2000.

[Rao1998] R. M. Rao, A. S. Bopardikar, Wavelet Transforms: Introduction to Theoty and Applications, Addison Wesley Longman, Inc.1998

[Schalkoff1992] R. J. Schalkoff, Pattern Recognition: Statistical, Structural and Neural Approaches, John Wiley & Sons, Inc., 1992

[Sim2000] Dong-Gyu Sim, Hae-Kwang Kim, Dae-Il Oh, "Translation, Scale, and Rotion Invariant Texture Discriptor for Texture-based Image Retrieval," Image Processing 2000, Vol.3, pp742-745, Sep. 2000

[Smith1996] John R. Smith and Shih-Fu Chang, VisualSEEk: a fully automated content-based image query system. ACM Multimedia 96, Boston, MA,1996.

[Subrahmanian1998] V. S. Subrahmanian, Principles of Multimedia Database systems, Morgan Kaufmann Publishers, Inc., 1998

[Weber1998] R. Weber, H-J. Schek, S. Blott, "A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces", VLDB, pp.194-205,1998

[Weiyu2000] Z. Weiyu, S. Levinson, "Edge orientation-based multi-view object recognition", Pattern Recognition, 2000. Proceedings. 15th International Conference, vol.1, pp936-939, 2000

[Xie1991] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," IEEE Trans. Pattern Analysis and Machine Intelligence, vol 13, no. 8, 841-847, 1991.

[이상경1996] 이상경, "스펙트럴 분석 및 신경-유전자-퍼지 복합망을 이용한 이동, 크기 및 회전 변형에 무관한 패턴인식", 고려대학교 대학원 박사학위 논문, 1996.

부록 : 소프트웨어 도큐먼트

Block / Unit 구성도 및 기능 설명

1. 객체 처리 (Object Handler) Block

    1-1. Object Handler Block _ Parser Unit

2. 특징벡터 추출 (Feature Vector Abstraction) Block

    2-1. 특징벡터 추출 Block _ 2D Unit

    2-2. 특징벡터 추출 Block _ 3D Unit

3. 통신 (Communication) Block

    3-1. Communication Block _ Communication Unit

4. 자료 / 검색 (DB / Search) Block

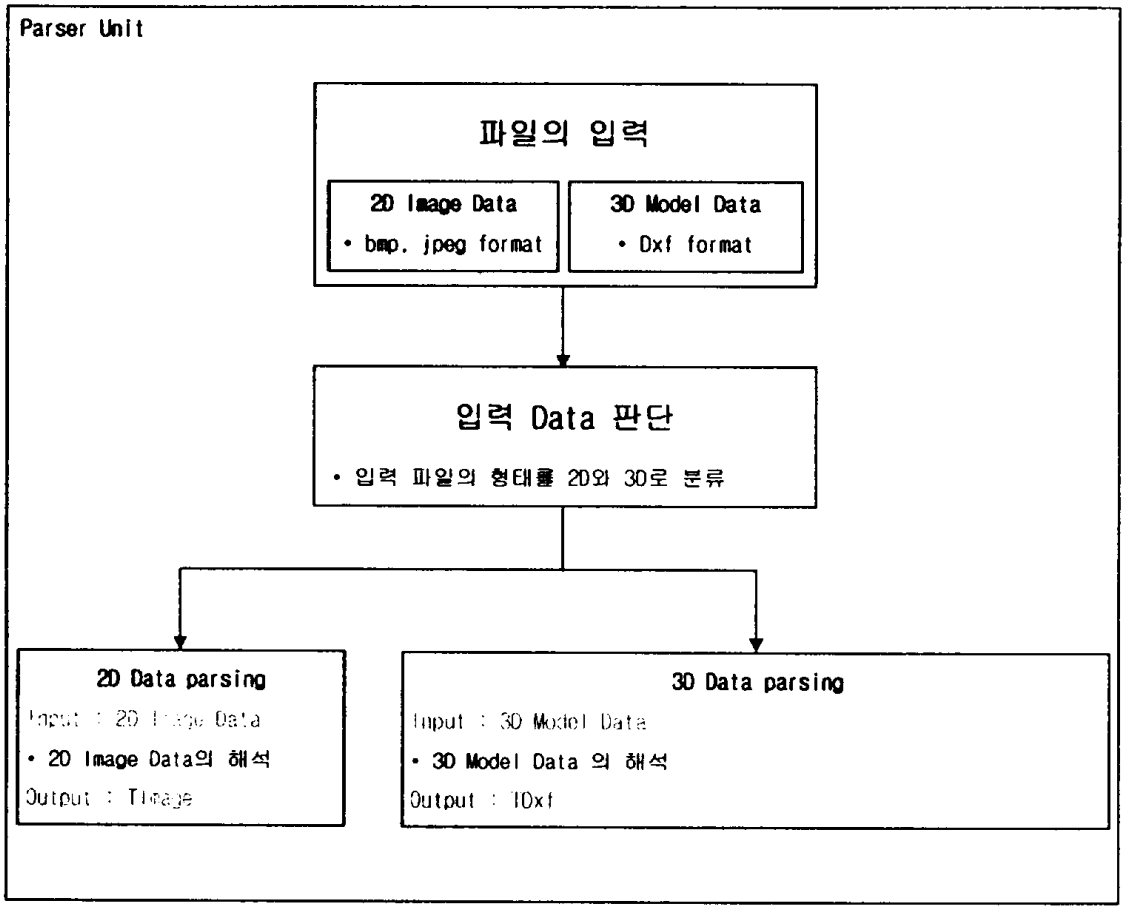
    4-1. DB / Search Block _ DB/검색 Unit

System 구성	시스템 명	3차원형상매칭	버전	V1.0
	Block 단위 명	Object Handler Block body		

Block / Unit

1. Object Handler Block

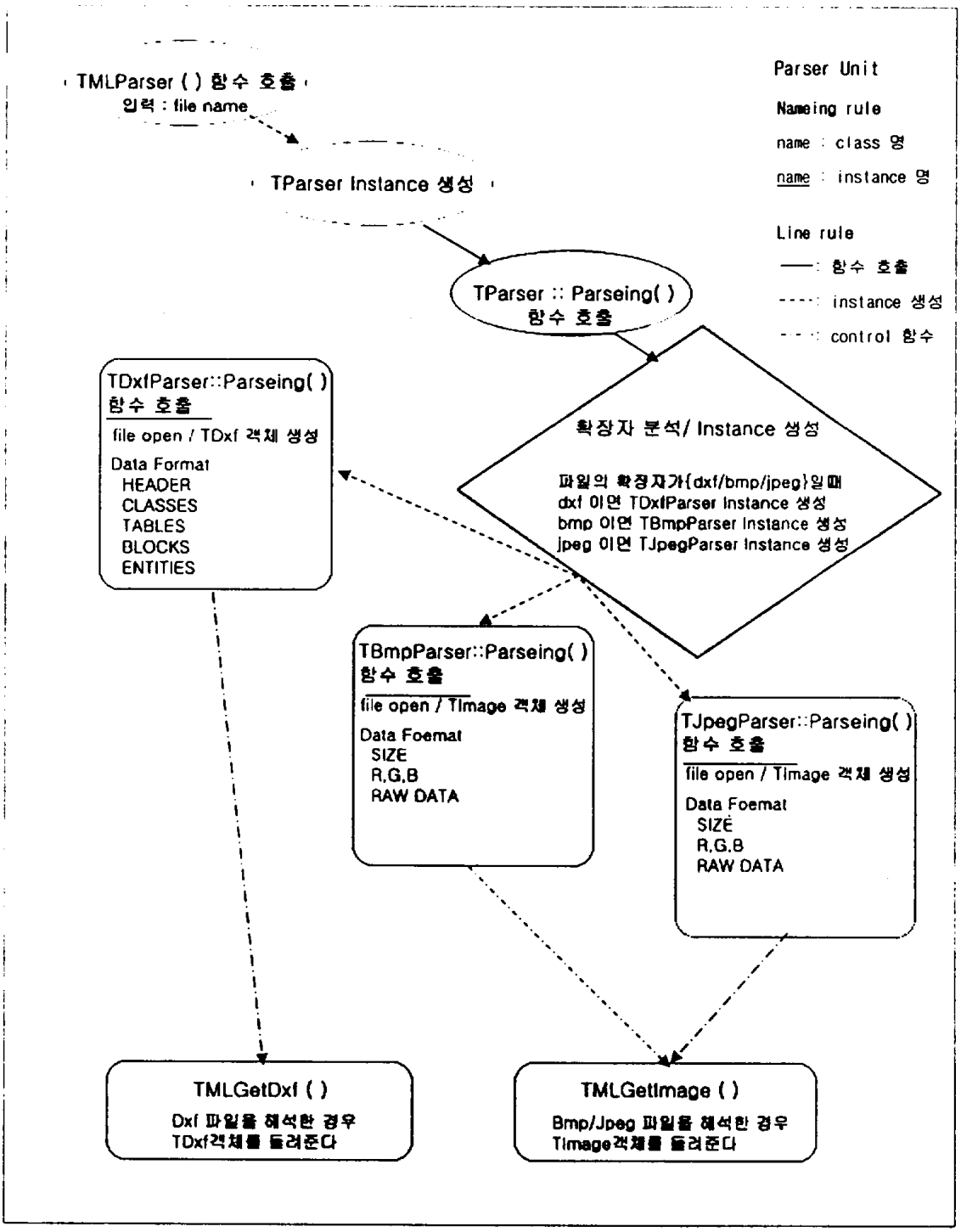
• 구성도



System 구성	시스템 명	3차원형상매칭	버전	V1.0
	Block 단위 명	Object Handler Block _ Parser Unit diagram		

1-1. Object Handler Block _ Parser Unit

• 구성도



System 구성	시스템 명	3차원형상매칭	버전	V1.0
	Block 단위 명	Object Handler Block _ Parser Unit diagram		

• Parser Unit Interface 함수

함수명	세부 설명

_TMLParsersing()	원형: int _TMLParsing(PARSINGINFO* pInfo)
	int _TMLParsingEx(char* fileName, TCallback *pCallBack)
	Input :
	struct tagParsing {
	char* fileName; // 입력파일명
	int iVal; // 파일의 종류(DXF = 1, BMP = 2, JPG = 3)
	TData* pParse; // Parsing된 메모리 객체
	TCallback *pCall; // Callback Class
	} PARSINGINFO;
	fileName : Parsing할 파일 이름.
	기능
	파일명을 입력받아 그 파일을 읽어 해석한 후 파일의 종류에 맞는 인스턴스를 생성한다. 생성된 메모리 객체는 입력된 구조체에 전달이 되며, DXF Model 의 경우는 TDxf 객체를 PARSINGINFO 구조체의 pParser 에, iVal 에는 FILE_DXF(1)의 값이 저장된다. BMP 및 JPEG 파일의 경우에는 TImage 객체를 pParser 에 저장하며, iVal 에는 FILE_BMP(2) 또는 FILE_JPEG(3)의 값이 저장된다.
	파일명만을 입력인자로 넘겨 준 경우에는 아래의 함수들을 사용하여 결과값을 받아온다.
	Return
	값이 0 이상이면 handle를 넘겨주고 오류 이면 -1 을 넘겨준다.

_TMLGetDxf(int handle)	원형: TDxf* TMLGetDxf() Input : int handle : TMLParser() 함수에서 받은 Handle 값 기능 Dxf File 을 입력하여 해석한 경우 TDxf 객체를 돌려준다. TMLParser 를 호출한 후 구조체에 pParser 에 입력된 값과 동일하다. Return DXF 파일을 해석한 경우에는 TDxf 객체를, 그렇지 않은 경우에는 NULL 을 돌려준다.
_TMLGetImage(int handle)	원형: TImage* TMLGetImage() Input : int handle : TMLParser() 함수에서 받은 Handle 값 기능 BMP, JPEG 파일을 해석한 경우 TImage 객체를 돌려준다. TMLParser 함수 호출 후 구조체에 pParser 값과 동일하다. Return BMP 및 JPEG 파일을 입력했을 경우 TImage 객체를, 그렇지 않은 경우 NULL 값을 돌려준다.

_TMLFree(int handle)	원형 : void TMLFree() Input : int handle : TMLParser() 함수에서 받은 Handle 값 기능 Parser 를 사용한 후 내부에 생선된 메모리등을 해제한다. 이 함수 호출후에는 TMLGetDxf()등의 함수 호출은 무효화된다. Return : void
_TMLGetLastError()	오류 코드

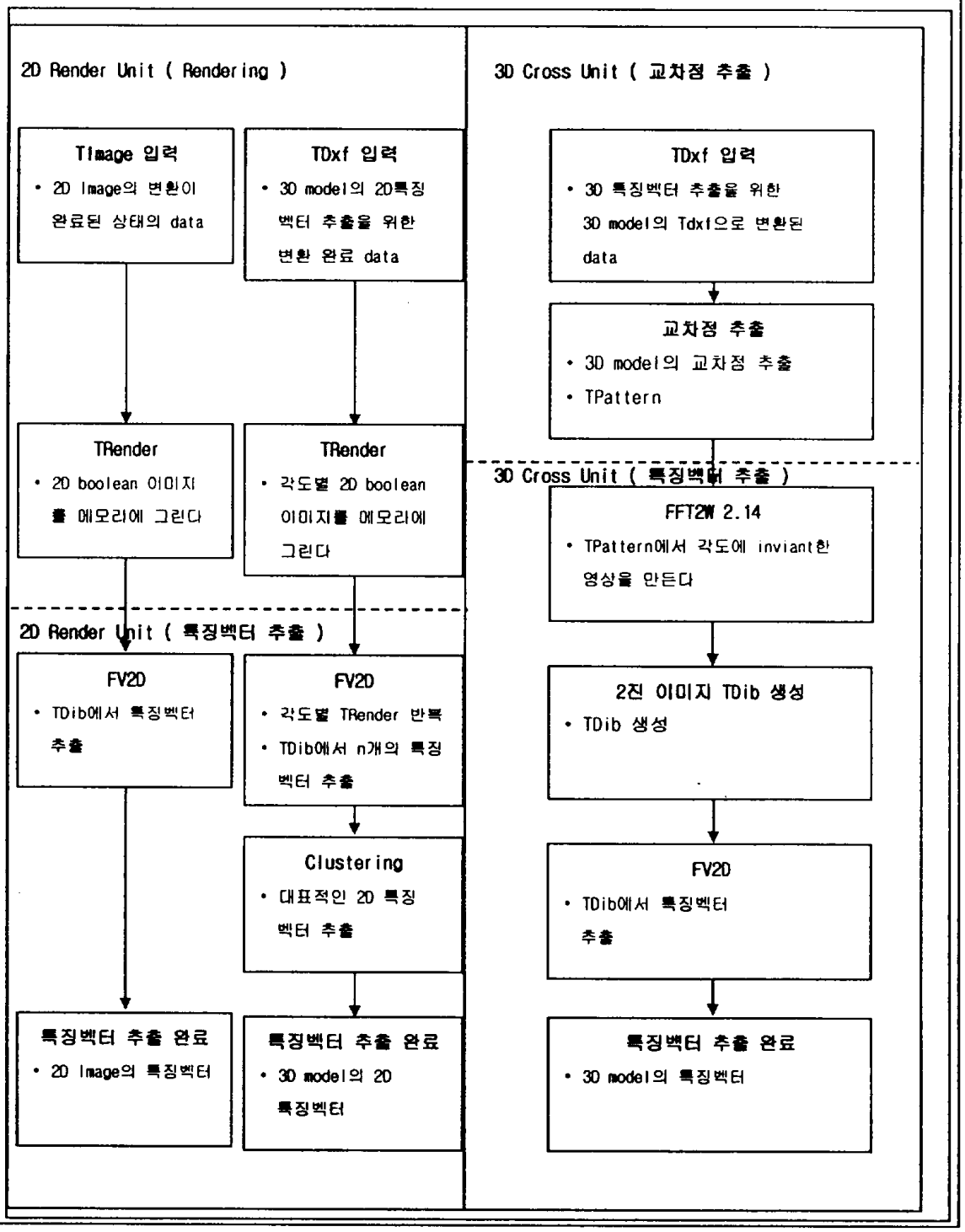
오류 상수	값	설 명
R_OK	0	성공
CouldNotOpen	1	파일을 열 수 없슴
InvalidFile	2	DXF 파일이 아님
ErrorDxf	3	TDxf객체 오류
InsufficientMemory	4	메모리 부족
InvalidSize	5	크기설정 오류
NotFound	6	교차점 없음
InvalidCompress	8	알 수 없는 압축방식
UnknownError	255	알 수 없는 오류

_TMLGetVer	원형 : const char* _TMLGetVer() Input : void 기능 라이브러리의 버전정보를 돌려준다. Return : Version 문자열

System 구성	시스템 명	3차원형상매칭	버전	V1.0
	Block 단위 명	특징벡터 추출 Block body		

2. 특징벡터 추출 Block

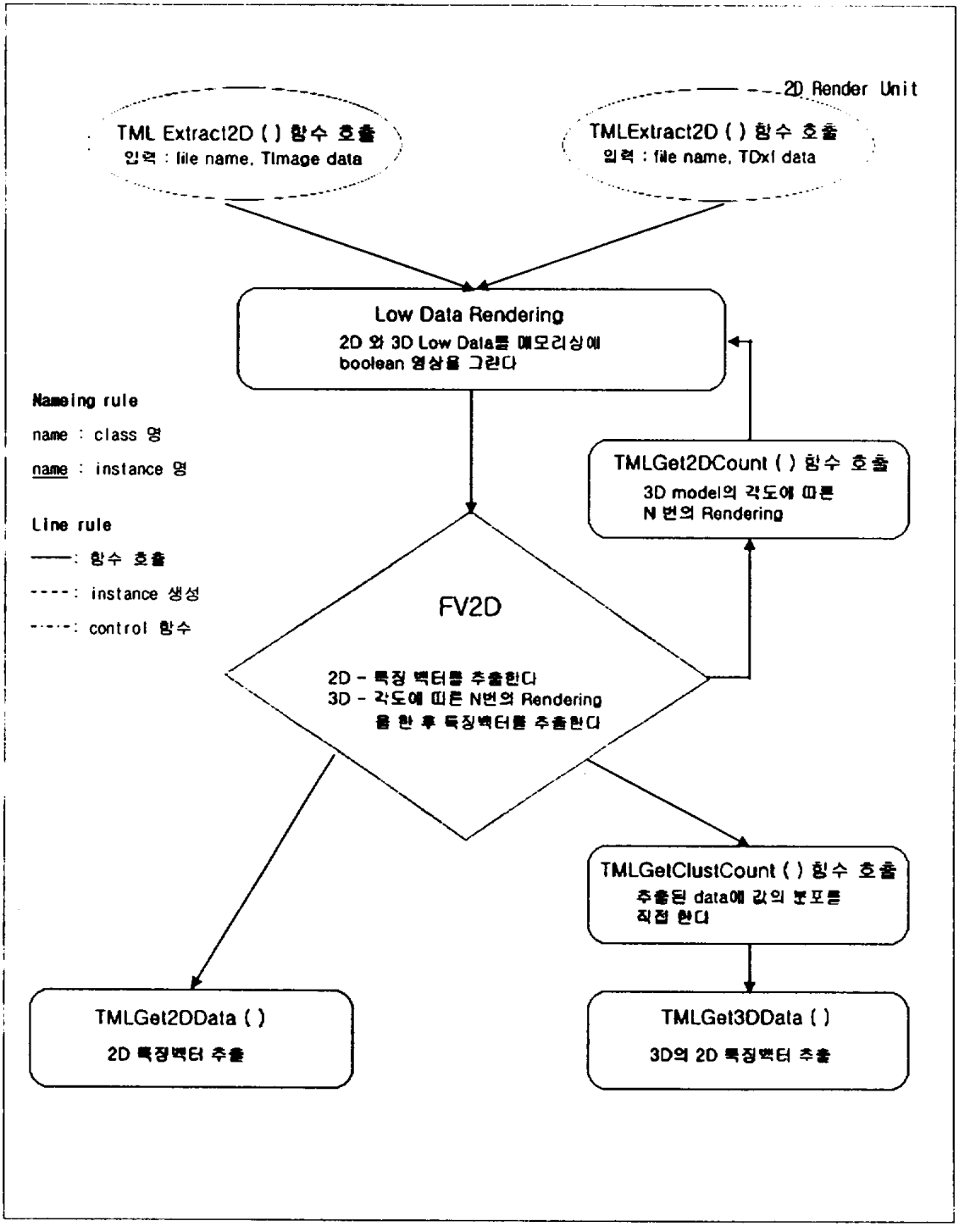
• 구성도



System 구성	시스템 명	3차원형상매칭	버전	V1.0
	Block 단위 명	특징벡터 추출 Block _ 2D Render Unit diagram		

2-1. 특징벡터 추출 Block _ 2D Render Unit

• 구성도



System 구성	시스템 명	3차원형상매칭	버전	V1.0
	Block 단위 명	특징벡터 추출 Block _ 2D Render Unit diagram		

• 2D Render Unit Interface 함수

함수명	세부 설명
TMLSetFileName()	원형 : int TMLSetFileName(char* fileName, BOOL bParsing); 목적 : Data File을 지정한다. 가장 초기에 지정해야 하는 부분이다. Input : fileName : 파일명을 지정한다. bParsing : 파일명을 지정 후 메모리에 읽어들일지를 지정한다 기능 : TRUE인 경우 메모리로 읽어 들이며, FALSE인 경우에는 TMLExtract2D, TMLExtrac3D 등을 사용할 때 읽어들인다. Return : 0 이상의 핸들값을 돌려준다. 만약 0보다 작다면 오류가 발생한 것이며, 오류 코드를 참조하도록 한다.
TMLSetEnvironment2D()	원형 : int TMLSetEnvironment2D(int handle, int level, int start, BOOL bEdge, int interval, BOOL bUseZ); 목적 : 2D 특징벡터를 추출하기 위한 설정을 한다. Input : handle : TMLSetFileName, TMLSetDxf, TMLSetImage 에서 돌려받은 handle 값. level : Zernike Moment 를 계산할 때의 차수. 차수가 높을 수록 특징벡터의 개수가 많아진다. (10 : 36 개, 8 : 25 개) 차수에 대한 계산식은 다음과 같다. (level / 2 + level % 2 + 1) * (level / 2 + 1) 기본값은 8이다. start : 특징벡터의 시작위치를 지정한다. 일반적으로 Zernike Moment 의 첫 번째는 화소의 개수 두 번째는 항상 0 이 된다. 특히 첫 번째 값은 Normalize 의 요소가 되므로, Normalize 를 하게 되면 항상 동일한 값이 된다. 그러므로 두 값을 제외해도 무방하다. 그러므로 기본값은 2 로 설정되어 있다. 이럴때 전체 특징벡터의 개수는 level 에서 지정된 개수에 이 값을 빼주면 된다. bEdge : 특징벡터를 추출할 때 외곽선을 이용할지 전체를 이용할지 결정한다. TRUE 인 경우는 외곽선을, FALSE 인 경우는 전체 화소를 가지고 계산한다. 외곽선을 이용하는 것이 계산이 빠르다. 기본값이 TRUE이다. Interval : 하나의 3D Model 을 각 축으로 몇도 간격으로 회전시킬지 결정한다. 기본값으로 30도이다. bUseZ : Z축 회전을 사용할지 여부. Return : 필요없음. -> void로 변경

상세 설명	시스템 명	3차원형상매칭	버전	V1.0
	Block 단위 명	특징벡터 추출 Block _ 2D Render Unit diagram		

함수명	세부 설명
TMLGetDimCount2D()	원형: int TMLGetDimCount2D(int handle); 목적 : 2D 특징벡터의 차수를 돌려준다. TMLSet2DEnvironment 의 level 과 start 에 의해 차수가 결정된다. level 이 10 이고 start 가 2 이면 특징벡터의 차수는 34 가 된다. 즉 34 개의 실수값으로 특징벡터가 생성된다. Input : int handle : TMLSet..에 의해 돌려진 handle 값. Retrun : 2D 특징벡터의 차수가 돌려진다.
TMLGetCount2D()	원형: int TMLGetCount2D(int handle); 목적 : 2D 특징벡터의 개수를 돌려준다. 3D 모델에서만 적용이 된다. Input : handle : handle 값. Return : 2D 특징벡터의 개수를 돌려준다. 2D 이미지의 경우는 1 을 돌려주며, 0 이면 특징벡터가 추출되지 않은경우이다. TMLSet2DEnvironment 의 interval 에 의해 개수가 결정되며, 30 도 간격인 경우 1728 개의 특징벡터를 생성한다. 계산식은 다음과 같다. $(interval / 360)^3$
TMLGetClustCount()	원형: int TMLGetClustCount(int handle); 목적 : Clustering 된 특징벡터의 개수를 돌려준다. 3D 모델에서만 적용이 된다. Input : handle : TMLSet..에 의해 돌려진 handle 값. Return : Cluster 의 개수를 돌려준다. 0 이하인 경우는 오류를 나타내며, 0 인 경우는 2D 이미지의 특징벡터이거나, Clustering 을 하지 않은 경우이다.
TMLExtract2D()	원형: int TMLExtract2D(int handle, BOOL bCluster); 목적 : 2D 특징벡터를 추출한다. Input : handle : 핸들값. bCluster : 클러스터링을 할지 여부를 지정한다. Return : 특징벡터의 차수를 돌려준다. TMLGet2DDimCount 로도 알 수 있다. 0 이하인 경우는 오류를 나타낸다.
TMLGetData2D()	원형 : float TMLGetData2D(int handle, int index, int col); 목적 : 특징벡터의 값을 가져온다. Input : handle : 핸들값. index : 0 부터 2D 특징벡터의 개수, 몇 번째 특징벡터인지를 지정한다. 2D 이미지인경우는 항상 0 이 되어야 한다. col : 차수를 지정한다.

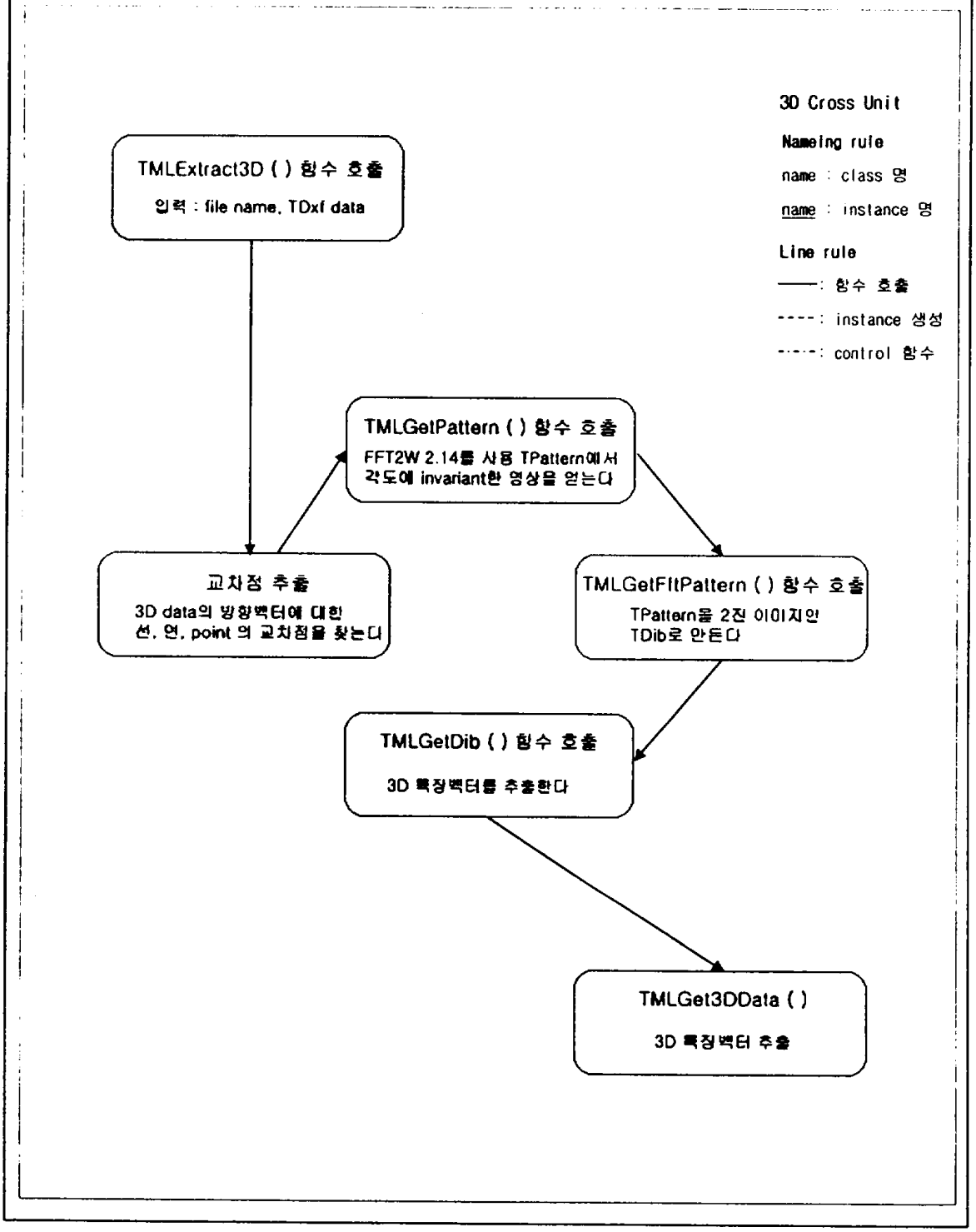
상세 설명	시스템 명	3차원형상매칭	버전	V1.0
	Block 단위 명	특징벡터 추출 Block _ 2D Render Unit diagram		

함수명	세부 설명
	Return : 특징벡터 값. 오류가 있거나, 잘못된 위치를 지정한 경우는 -1을 돌려준다.
TMLGetDataClust()	원형: float TMLGetDataClust(int handle, int index, int col); 목적 : Clustering된 2D 대표특징벡터값을 가져온다. Input : handle : 핸들값. index : 0부터 Cluster 갯수. col : 0부터 2D 특징벡터의 차수 Return : 특징벡터값. 오류가 있거나, 잘못된 위치를 지정한 경우는 -1을 돌려준다.
TMLGetNormData2D()	원형: float TMLGetNormData2D(int handle, int index, int col); 용도 : TMLGet2DData와 동일. 단 Normalize된 값을 돌려준다.
TMLFVFree()	원형 : void TMLFVFree(int handle); 목적 : 메모리를 해제한다. 반드시 사용후에 호출해줘야 한다. Input : handle : 핸들값. Return : 없음.

System 구성	시스템 명	3차원형상매칭	버전	V1.0
	Block 단위 명	특징벡터 추출 Block _ 3D Cross Unit diagram		

2-2. 특징벡터 추출 Block _ 3D Cross Unit

• 구성도



System 구성	시스템 명	3차원형상매칭	버전	V1.0
	Block 단위 명	특징벡터 추출 Block _ 3D Cross Unit diagram		

• 3D Corss Unit Interface 함수

함수명	세부 설명
TMLSetFileName()	원형: int TMLSetFileName(char* fileName, BOOL bParsing); 목적 : Data File을 지정한다. 가장 초기에 지정해야 하는 부분이다. Input : fileName : 파일명을 지정한다. bParsing : 파일명을 지정 후 메모리에 읽어들일지를 지정한다. TRUE 인 경우 메모리로 읽어 들이며, FALSE 인 경우에는 TMLExtract2D, TMLExtrac3D 등을 사용할 때 읽어들인다. Return : 0 이상의 핸들값을 돌려준다. 만약 0보다 작다면 오류가 발생한 것이며, 오류 코드를 참조하도록 한다.
TMLSetEnvironment3D()	원형: int TMLSetEnvironment3D(int handle, int level, int start, BOOL bEdge, float thresh, float alpha, float beta); 목적 : 3D 특징벡터를 추출하기 위한 설정을 한다. Input : handle : 2D와 동일 level, start, bEdge : 2D와 동일. 단 level의 기본값은 10이다. thresh : FFT 결과를 이진영상화 할 때의 Threshold 값이다. FFT 변환 값의 크기가 이값 이상인 경우에 이미지로 적용된다. 기본값은 0.3 이며, 1.0까지 설정한다. alpha, beta : 극좌표계의 Alpha 와 Beta 의 간격이다. 기본값은 2 도간격으로 교차점을 추출한다. 값이 작을수록 패턴이 성세해지지만, 계산 시간이 길어진다. 1도 이상으로 설정한다. Return : 없음 -> void로 변경
TMLGetDimCount3D()	원형: int TMLGetDimCount3D(int handle); 목적 : 3D 특징벡터의 차수를 돌려준다. TMLSet3DEnvironment 의 level 과 start 에 의해 결정된다. Input : handle : TMLSet...에 의해 돌려진 handle 값. Return : 2D 특징벡터의 차수가 돌려진다.
TMLExtract3D()	원형 : int TMLExtract3D(int handle); 목적 : 3D 특징벡터를 추출한다. Input : handle : 핸들값. Return : 특징벡터의 차수를 돌려준다. TMLGet3DDimCount 로도 알 수 있다. 0 이하인 경우는 오류를 나타낸다.
TMLGetData3D()	원형: float TMLGetData3D(int handle, int col); 목적 : 3D 특징벡터의 값을 가져온다.

System 구성	시스템 명	3차원형상매칭	버전	V1.0
	Block 단위 명	특징벡터 추출 Block _ 3D Cross Unit diagram		

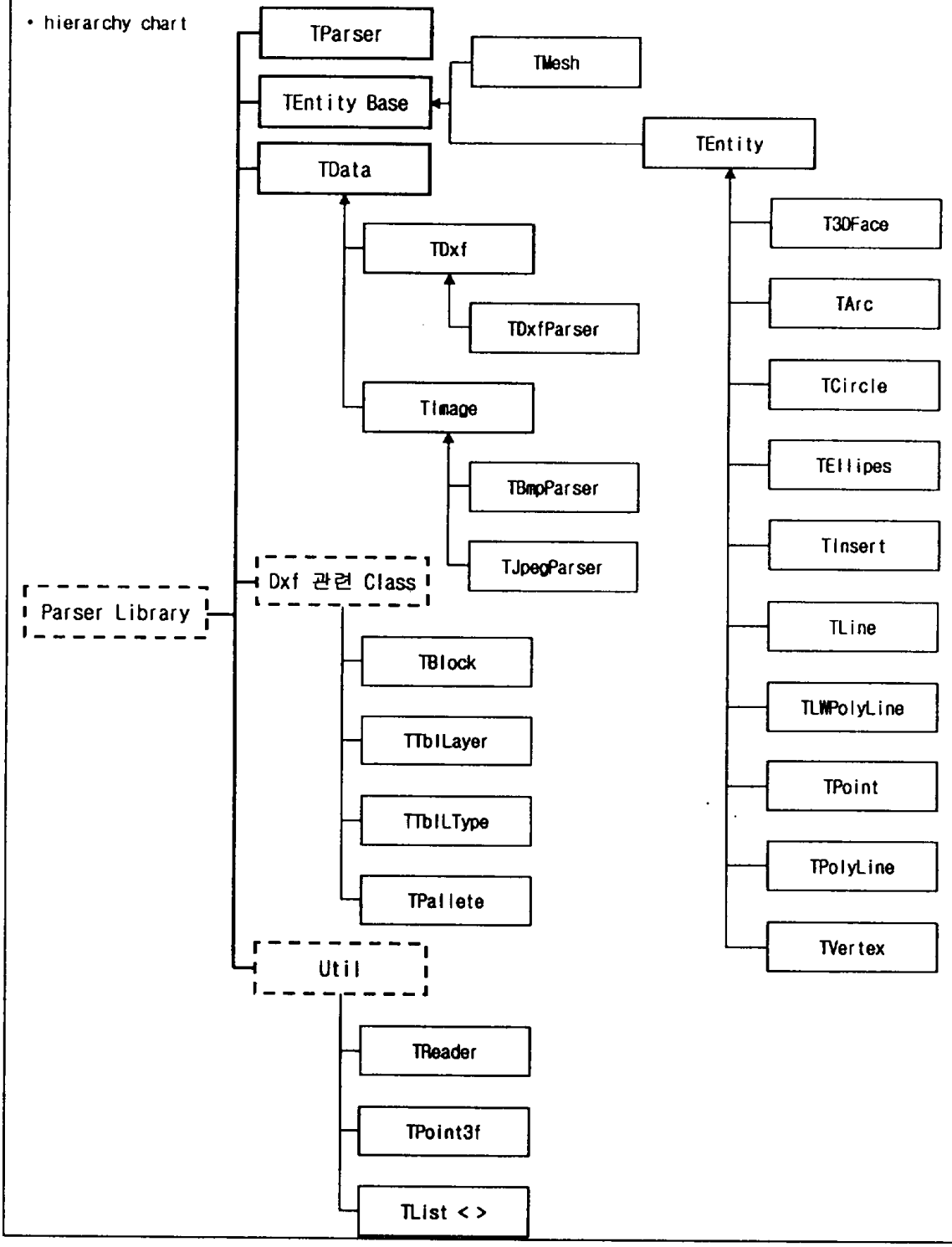
함수명	세부 설명
	Input : handle : 핸들값. col : 0부터 3D 특징벡터의 차수. Return : 특징벡터 값. 오류가 있거나, 잘못된 위치를 지정한 경우는 -1을 돌려준다.
TMLGetNormData3D()	원형: float TMLGetNormData3D(int handle, int col); 용도 : TMLGet3DData와 동일. 단 Normailize된 값을 돌려준다.
TMLFVFree()	원형: void TMLFVFree(int handle); 목적 : 메모리를 해제한다. 반드시 사용후에 호출해줘야 한다. Input : handle : 핸들값. Return : 없음.
TMLGetPattern()	원형 : TPattern* TMLGetPattern(int handle); 목적 : 3D 모델의 거리값 패턴을 가져온다. Input : handle : 핸들값. Return : 3D 특징벡터를 추출한 패턴을 가져온다. 특징벡터를 추출하지 않았으면 NULL을 돌려준다.
TMLGetFftPattern()	원형 : TPattern* TMLGetFftPattern(int handle); 목적 : 3D 모델의 FFT 변환 결과를 가져온다. Input : handle : 핸들값. Return : 3D 특징벡터를 추출할때의 FFT 변환 결과를 가져온다. 없을 경우 NULL을 돌려준다.
TMLGetDib()	원형 : TDib* TMLGetDib(int handle, float x, float y, float z); 목적 : 렌더링한 이진영상을 가져온다. 입력된 값에 따라 다르다. Input : handle : 핸들값. x, y, z : 각 축의 회전값. 2D 이미지에서는 아무역할도 하지 않으며, 3D 특징벡터를 추출한 경우도 아무 의미가 없다. Return : 이진영상을 가져온다. 없을 경우 NULL을 돌려준다.

상세 설명	시스템 명	3차원형상매칭	버전	V1.0
	Block 단위 명	Object Handler Block _ Parser Uint details		

세부 설명

2-1. Object Handler Block _ Parser Unit

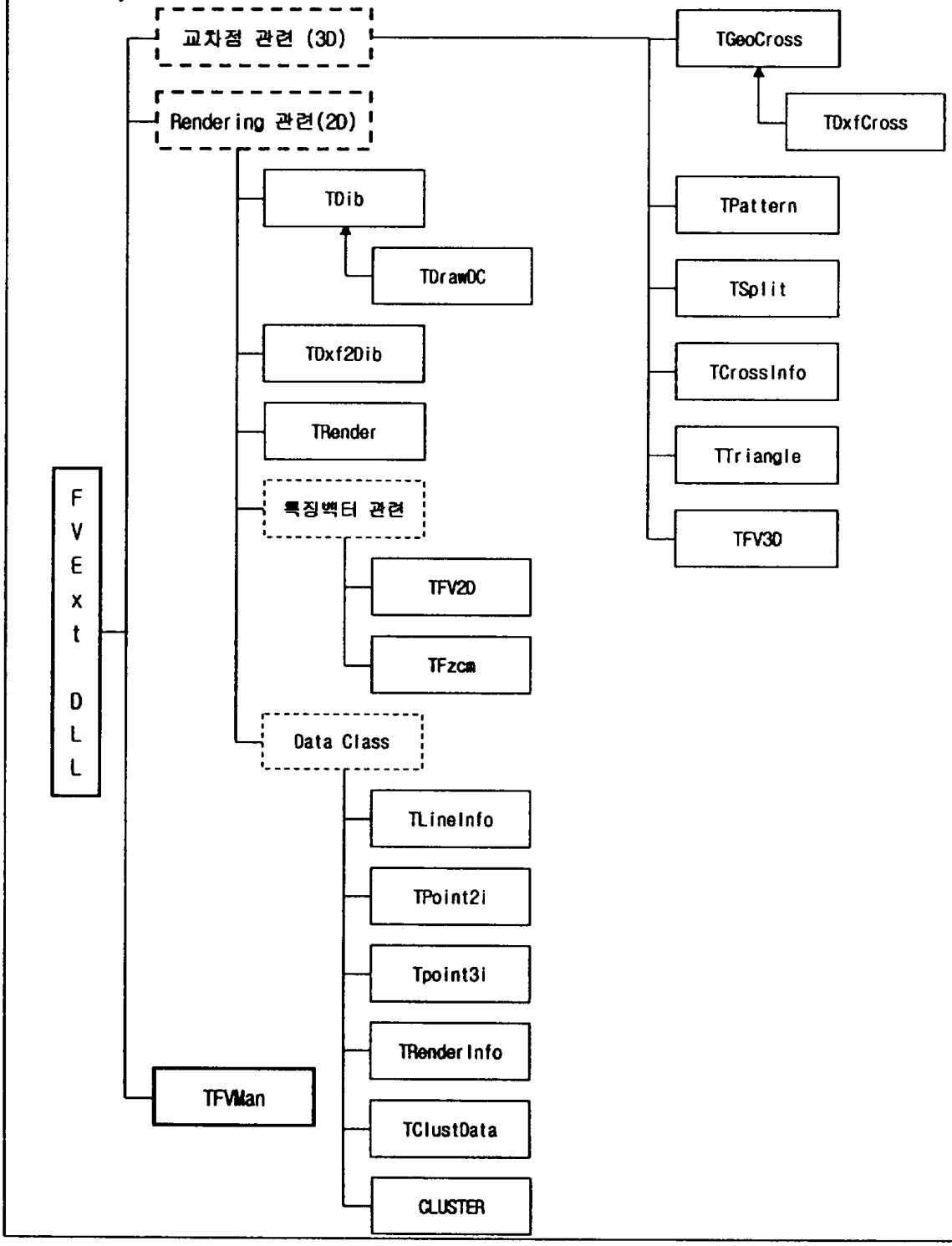
• hierarchy chart



상세 설명	시스템 명	3차원형상매칭	버전	V1.0
	Block 단위 명	특징벡터 추출 Block		

2-2. 특징벡터 추출 Block

• hierarchy chart



상세 설명	시스템 명	3차원형상매칭	버전	V1.0
	Block 단위 명	특징벡터 추출 Block		

극좌표에서 직교좌표로의 변환

방향의 단위벡터를 i, j, k라 할 때 극좌표 (a, b)와는 다음의 식으로 변환될 수 있다.

i = cos(a) * cos(b)

j = sin(a) * cos(b)

k = sin(b)

구현함수 : Angle2Direction()

방향벡터와 선분의 교차판별

방향벡터 V(i, j, k)와 선분 A1(x1, y1, z1), A2(x2, y2, z2)와의 교차점은 다음식으로 판별한다.

방향벡터를 하나의 선분으로 생각하면 sV라 할 수 있다. 여기서 s > 0 이어야 한다. 이 두 선분이 교차한다면 교차점은 방향벡터의 연장선 상에 있으므로 (si, sj, sk)라고 할 수 있다. 두 선분이 모두 이 교차점을 통과한다면 다음의 식을 만족시켜야 한다.

이를 행렬식으로 변환한다면,

$$\begin{vmatrix} si & -(x_2-x_1)t & x_1 \\ sj & -(y_2-y_1)t & = y_1 \\ sk & -(z_2-z_1)t & z_1 \end{vmatrix}$$

$$\begin{vmatrix} i & -\Gamma x \\ j & -\Gamma y \\ k & -\Gamma z \end{vmatrix} \begin{matrix} s \\ t \end{matrix} = \begin{vmatrix} x_1 \\ y_1 \\ z_1 \end{vmatrix} \quad \begin{cases} \Gamma x = x_2 - x_1 \\ \Gamma y = y_2 - y_1 \\ \Gamma z = z_2 - z_1 \end{cases}$$

이 되고, 이를 전개하면

$$\begin{vmatrix} I & -J \\ -J & K \end{vmatrix} \begin{vmatrix} s \\ t \end{vmatrix} = \begin{vmatrix} L \\ -M \end{vmatrix}$$

$$\begin{matrix} s \\ \hline t \end{matrix} \quad \begin{matrix} 1 \\ I \quad K-J^2 \end{matrix} \begin{vmatrix} K & J & L \\ J & I & -M \end{vmatrix}$$

여기서 I, J, K, L, M의 값을 계산하여 s, t의 값을 구하면 된다. s는 방향벡터의 비율이므로 0보다 커야하고 t는 선분내의 위치이므로 0과 1사이의 값을 가질 때 두 선분이 교차한다고 말할 수 있다.

구현함수 : GetLineCross()

상세 설명	시스템 명	3차원형상매칭	버전	V1.0
	Block 단위 영	특징벡터 추출 Block		

방향벡터와 평면과의 교차판별 :

3D 상에서 평면의 방정식은 ax + by + cx + d = 0이고 임의의 세점은 평면을 나타낼 수 있다. 평면 방정식을 풀기 위해 다음 식을 전개한다.

$$ax_1 + by_1 + cz_1 + d = 0$$
$$ax_2 + by_2 + cz_2 + d = 0$$
$$ax_3 + by_3 + cz_3 + d = 0$$

위의 방정식을 풀면

$$a = (y_3 - y_1)\cdot(z_2 - z_1) - (z_3 - z_1)\cdot(y_2 - y_1)$$
$$b = (z_3 - z_1)\cdot(x_2 - x_1) - (x_3 - x_1)\cdot(z_2 - z_1)$$
$$c = (x_3 - x_1)\cdot(y_2 - y_1) - (y_3 - y_1)\cdot(x_2 - x_1)$$
$$d = -(a\cdot x_1 + b\cdot y_1 + c\cdot z_1)$$

으로 풀 수 있다. (구현함수 : TTriangle::FindPalne())

우리가 구하고자 하는 교차점은 방향벡터 (i, j, k)의 연장선상에 있으므로 (it, jt, kt)라 할 수 있고 다음의 식을 만족하여야 한다.

$$a\cdot i\cdot t + b\cdot j\cdot t + c\cdot k\cdot t + d = 0$$

여기서 t를 구하면

$$t = \frac{-d}{a\cdot i + b\cdot j + c\cdot k}$$

이다. t는 방향벡터의 연장선이므로 t>0 이면 평면과 방향벡터는 교차한다고 판별 할 수 있다.

이후에는 위의 평면방정식은 영역이 무한정으로 뻗어 있기 때문에 주어진 3점의 안에 포함되어 있는지 확인을 해야 한다. 이는 Votex를 이용한 방법을 사용하여 판별하였다.

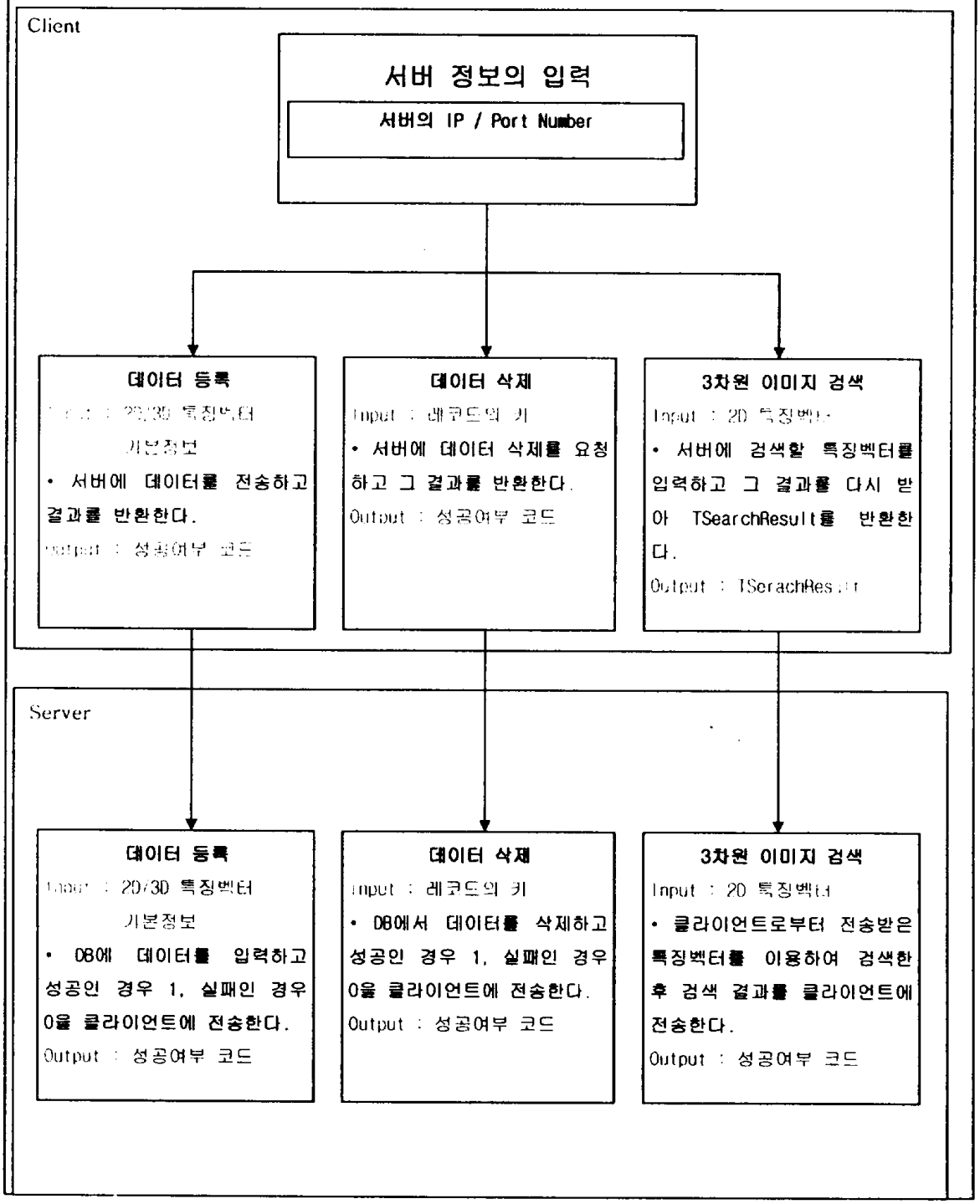
구현함수 : GetTriCross(…)

System 구성	시스템 명	3차원형상매칭	버전	V1.0
	Block 단위 명	통신 Block body		

Block / Unit

3. 통신 Block

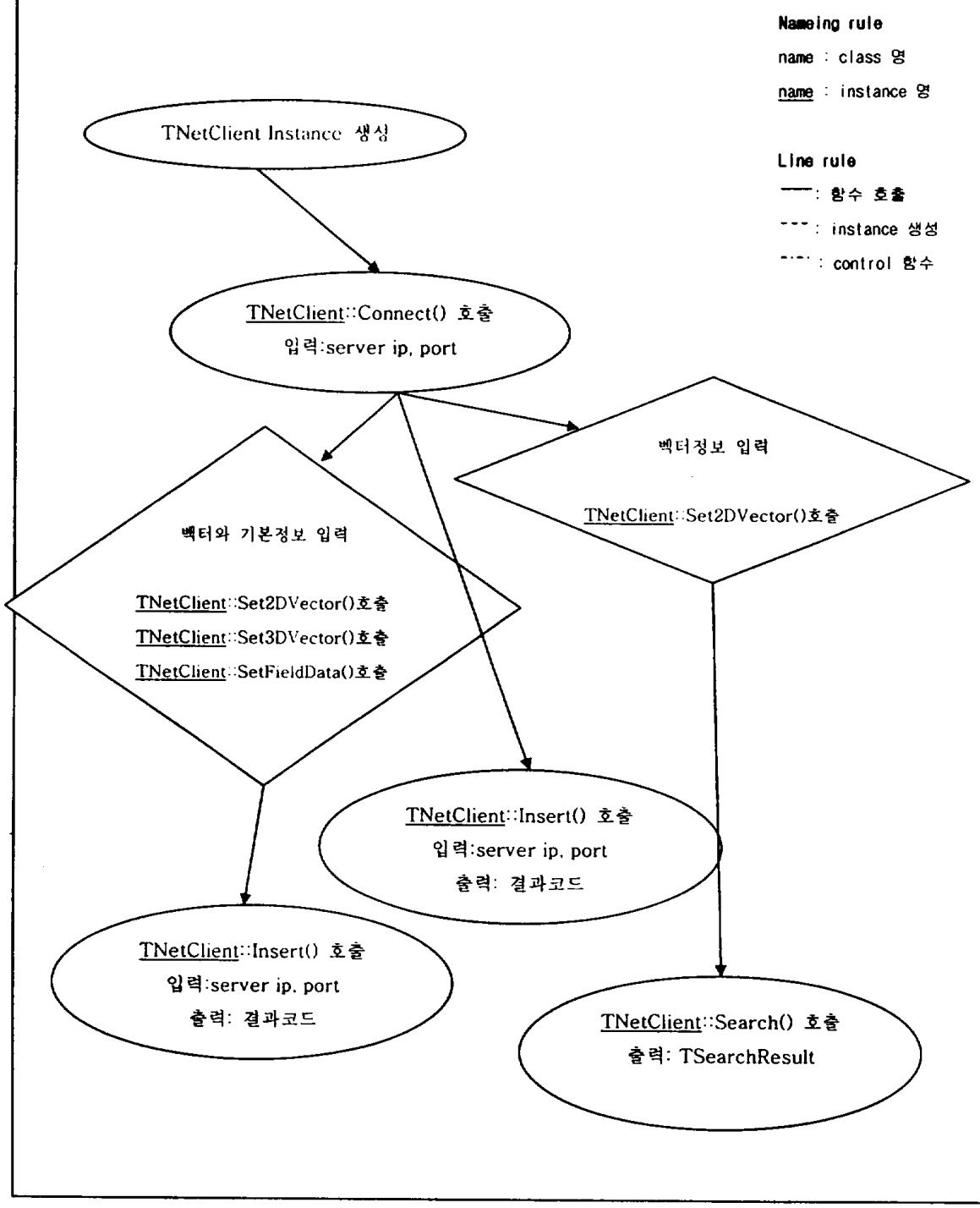
• 구성도



System 구성	시스템 명	3차원형상매칭	버전	V1.0
	Block 단위 명	통신 Block body		

3. 통신 Block

• 구성도

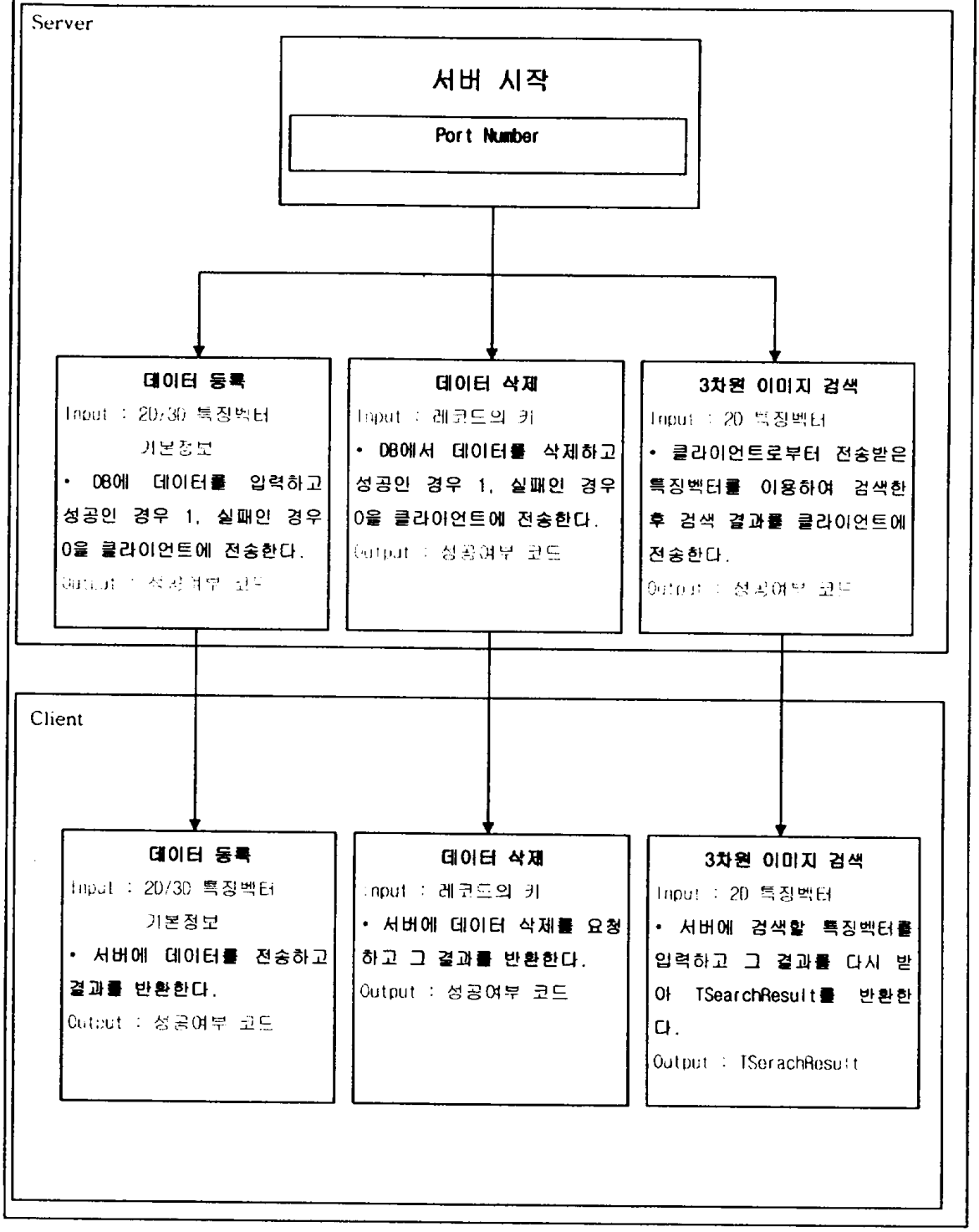


System 구성	시스템 명	3차원형상매칭	버전	V1.0
	Block 단위 명	DB / 검색 Block body		

Block / Unit

4. DB / 검색 Block

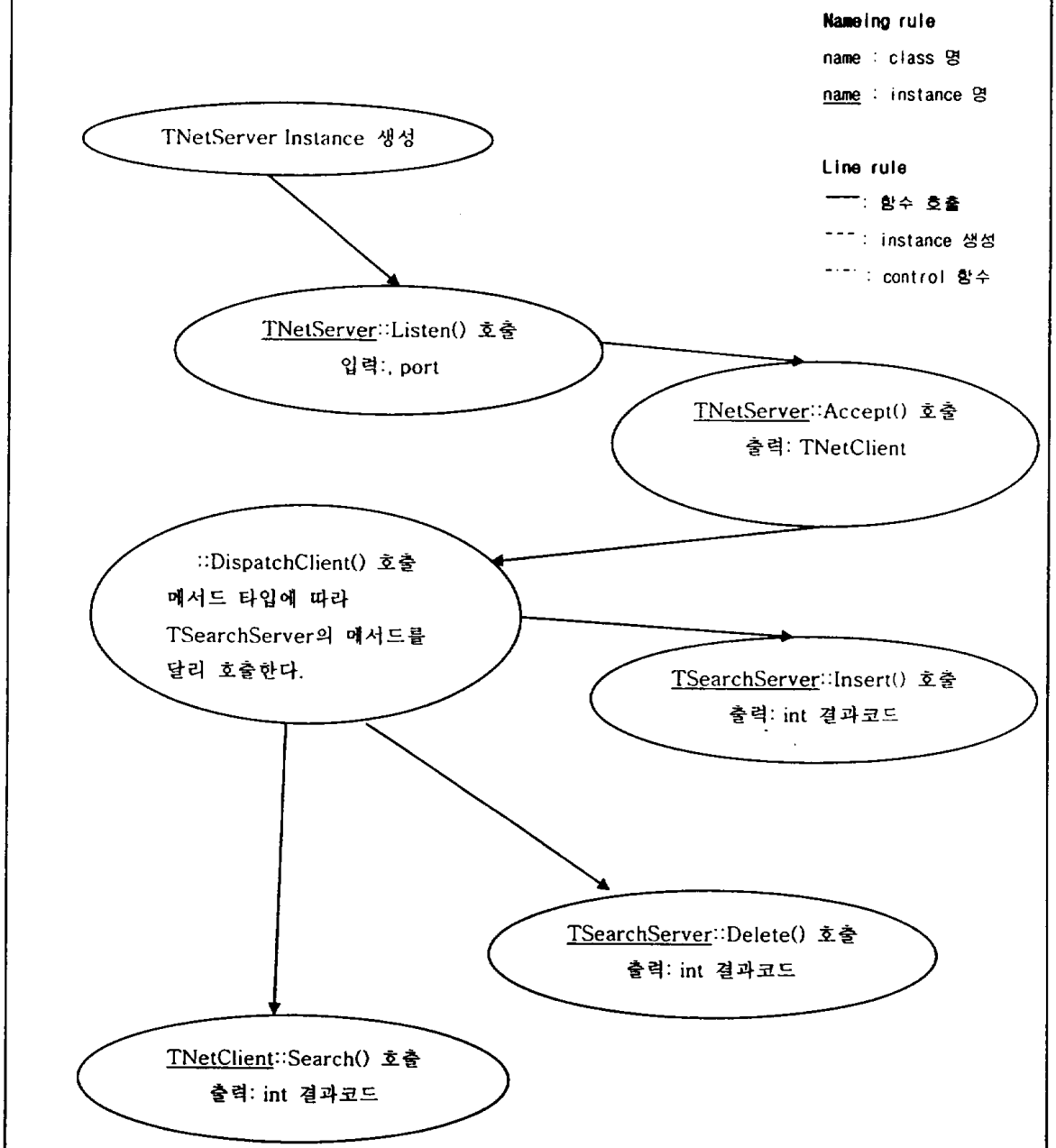
• 구성도



System 구성	시스템 명	3차원형상매칭	버전	V1.0
	Block 단위 명	DB / 검색 Block body		

4. DB / 검색 Block

• 구성도



감 사 의 글!

석사학위 후 긴 공백기간 동안 격정 속에 많은 일을 해 왔다 싶지만 어렵던 지난 일들은 맞닿은 시간의 파일에 누적돼 압축된 기억처럼 뚜렷하게 이룬 것도 없이 오늘의 현실에 쫓기며 햇수만 지난 것 같습니다. 대학원 과정동안 실무현장에서 겪고 느꼈던 많은 것을 연구하려고 다짐했었지만 초심과 같이 최선을 다하지 못함을 근간에 처해있는 여건 속에 주경야독 한다는 명분으로 합리화 하려 했던 자신에 대해 부끄러움과 아쉬움이 남습니다.

이젠 다가오는 세월이 다음이라는 기회가 없다는 생각에 이 학위를 계기로 그 간에 마음먹어 왔던 보람 있는 도전을 통해서 여러 음덕을 베풀어 주신 분들께 보답고저 합니다.

문무겸장의 학자로서 언제나 학도 입장에서 배려하시고 문제를 해결해 주시며, 부족한 저에게 학위과정에서 논문완성에 이르기까지 지도편달을 해주신 지도교수 홍윤기 교수님께 진심으로 감사드립니다. 아울러 강의와 연구 활동의 바쁘신 일정 속에서 본 논문의 심사를 맡아 핵심을 살펴 주시고 지도해 주신 이재득 교수님, 안식년임에도 활발한 연구 활동을 하시면서 귀한 시간과 지혜를 주신 위남숙 교수님, 논문의 질적 향상을 위해 세밀한 검토와 지도를 해 주신 서윤호 교수님, 정열적이고 실체적인 강의와 사려 깊은 지도를 해주신 나 건 교수님께 깊은 감사를 드리며, 또한 학위과정 동안 많은 지도편달과 세심한 배려를 아끼지 않으셨던 원형규 교수님, 김대홍 교수님, 박명환 교수님, 유재건 교수님, 정병용 교수님, 홍정완 교수님, 고려대 장동식 교수님, 김문화 교수님, 그리고 석사과정 때부터 지금껏 한결같이 지도 편달해 주신 고려대 김영휘 교수님께 진심으

로 감사드립니다.

본 연구가 결실을 맺을 수 있도록 지혜와 수고를 아끼지 않았던 한국산업안전공단 이동경 박사님, 리라공고 김상빈 교감님, 김세종 교수님, 현대조선 오순영 과장님께도 동문수학의 깊은 정을 드립니다.

수학기간 중 바쁜 업무 중에도 여러 가지로 지원해 주신 한미데이타 박창현 대표님, 문외환 팀장, 이영부 팀장, 유재룡 팀장을 비롯한 임직원께 감사를 드리며, 신뢰와 기대로 항상 지켜준 동동회, 산우회, 칠우회, 경성회 동문 여러분께 신의 예로서 감사 드립니다.

그리고 어지러운 세파에서 흔들리는 저를 바로 세워 주시고 영육간에 인도해 주신 길음동 성당 심용섭(아우구스티노) 주임신부님과 고재호(바오로) 대부모님, 16구역 지학남(스테파노) 구역장님과 형제자매님들께 깊은 감사를 드립니다.

끝으로 오늘이 있기까지의 저를 키워 주시고 이 기쁨을 거룩하게 기다려 주신 부모님께 이 영광을 받칩니다. 또한 물심양면으로 지원하고 격려해 주신 근호 형님내외, 순호 동생내외, 혜정 동생내외와 친가, 외가의 어르신께 애틋한 가족애와 감사의 마음을 드리며 학창시절부터 언제나 믿음과 인내로 동반해 준 사랑하는 아내 강윤경 선생, 멋있고 사랑스런 아들 택일과 오늘의 기쁨을 함께 나누고 싶습니다.

2003 년 12 월

이 찬 호(바오로) 배상