

# 기계 도면에서 객체의 분리 추출에 기반한 치수선의 인식

## (Recognition of Dimension Lines Based on Extraction of the Object in Mechanical Drawings)

鄭閔守\*, 朴吉欽\*

(Yoon-Su Jung and Kil-Houm Park)

### 요 약

본 논문에서는 기계 도면으로부터 형태선, 꼬리선 및 연장선 등의 치수선을 자동적으로 인식하는 새로운 방법을 제안한다. 제안된 방법은 문자열이 제거된 기계 도면으로부터 객체 및 페루프 심볼을 추출한 후, 세선화, 선분 벡터화 및 벡터 결합 알고리즘을 이용하여 객체와 설명선을 벡터화한다. 마지막으로, 패턴 매칭의 방법을 이용하여 화살표를 인식한 후, 화살표의 방향 벡터와 무게 중심 정보를 이용하여 설명선으로부터 치수선을 인식한다. 제안된 방법은 기하학적인 모델링과 수학적인 연산을 이용함으로써 치수선의 인식이 간단해지는 이점이 있다. 제안된 방법을 KS 제도 표준에 따라 그려진 도면에 적용한 실험 결과가 제공된다.

### Abstract

This paper presents a new method that automatically recognizes the dimension lines (consisting of shape lines, tail lines and extension lines) from the mechanical drawings. In the proposed method, the object and closed-loop symbols are separated from the character-free drawings. Then the object lines and interpretation lines are vectorized by using several techniques such as thinning, line-vectorization, and vector-clustering. Finally, after recognizing arrowheads by using pattern matching, we recognize dimension lines from interpretation lines by using arrowhead's directional vector and centroid. By using the methods of geometric modeling and mathematical operation, the proposed method readily recognizes the dimension lines from complex drawings. Experimental results are presented, which are obtained by applying the proposed method to drawings drawn in compliance with the KS drafting standard.

### I. 서 론

도면 인식은 컴퓨터에 도면 입력의 자동화를 실현하기 위한 기술로서 1970년대부터 패턴 인식의 한 분야로서 본격적인 연구가 시작되었고 많은 분야에서 도면의 입력 자동화를 위한 연구가 진행되고 있다. 이들 가운데 논리 회로 도면 및 전자 회로 도면의 자동 입력에 대한 연구 성과는 상당히 진행되어 실용적인 시스템의 구현에까지 이르고 있다. 이에 반해 기계 도면

의 경우에는 도면의 복잡성으로 인해 입력 자동화에 관한 연구가 상대적으로 뒤쳐져 있는 상태이나 최근에는 이에 대한 관심이 증대되면서 연구가 가속되고 있으며 특히, 미국이나 일본 등에서 상당한 연구 성과가 나타나고 있다.

일반적인 기계 도면은 3차원 기계 부품을 2차원 평면상에 직교 투영한 객체(object)와 이러한 객체의 정의를 제공하는 치수선 및 치수문자, 그리고 보조적인 선분 및 심볼로 구성된다(그림 1). 여기서, 치수 문자를 제외한 치수선 및 보조적인 선분, 그리고 심볼들을 묶어서 설명선(interpretation line)이라고 정의한다<sup>1)</sup>

<sup>1)</sup> 이 때, 치수선은 꼬리선(tail line), 형태선(shape

\* 正會員, 慶北大學校 電子工學科

(Dept. of Elec. Eng., Kyungpook Nat'l Univ.)

接受日字: 1996年9月14日, 수정완료일: 1997年8月12日

연장선(extension line)으로 구성된다. 그림 2와 같이 화살표의 뒤에 붙어 있는 직선이 꼬리선이며 이러한 꼬리선 중에서 화살표를 쌍으로 가지는 직선을 형태선이라 정의한다. 그리고, 화살표와 직각을 이루는 직선을 연장선으로 나타낸다.

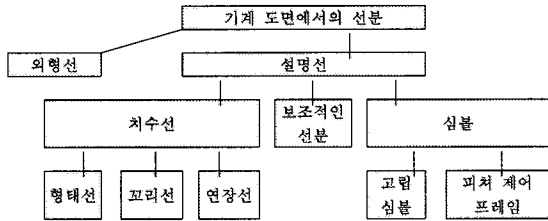


그림 1. 기계 도면에서 선분의 분류  
Fig. 1. Classification of lines in machine drawings.

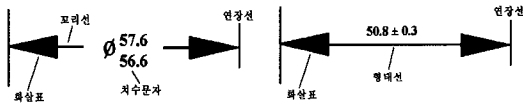


그림 2. 치수선의 예  
Fig. 2. Example of dimension lines.

현재, 기계 도면의 인식에 관한 연구는 도면의 자동 입력을 위한 벡터화 시스템의 개발<sup>[2-6]</sup>, 기계 도면의 주요한 요소인 문자, 해칭선의 추출 및 인식<sup>[7-9]</sup>, 기계 도면의 검색을 위한 객체의 지능적인 인식<sup>[10-11]</sup>, 그리고 기계 도면의 3차원 재구성을 위한 전제가 되는 치수 집합의 문법적인 표현<sup>[1]</sup>에 관한 연구가 부분적으로 진행되고 있다. 특히, 기계도면의 벡터화는 기계 도면의 자동 입력을 위한 영역으로서만이 아니라, 다른 분야의 연구를 위한 전처리 과정으로서도 큰 역할을 하고 있다.

하지만, 이러한 연구들이 전체 기계 도면 인식 시스템의 구현을 위해 이용되기 위해서는 다음과 같은 연구들이 수행되어야 한다. 첫째로, 기계 도면으로부터 기계 부품을 나타내는 객체와 객체에 대한 정의를 제공하는 치수선의 분리가 선행되어야 한다. 이전의 객체 인식 과정에서는 이러한 객체는 전처리로 분리되어 있다는 전제하에서 연구들이 수행되어 왔다. 따라서, 이러한 객체의 분리는 객체의 지능적인 인식 및 데이터 베이스의 구축을 위해서 반드시 필요한 부분이라고 할 수 있다. 둘째로, 기계도면의 자동 입력을 위해서는 화

살표, 치수선 및 치수 문자가 하나의 벡터로 인식되어야 한다. 그리고 이러한 치수선 및 치수 문자의 인식은 기계 도면의 3차원 재구성을 위한 전제라는 점에서 중요한 과제이다.

본 논문에서는 이러한 문제의 해결을 위해 다음과 같은 방법을 이용하여 기계도면의 인식을 수행한다. 첫째로, 기계 도면으로부터 객체를 효과적으로 분리한다. 이 때, 객체의 분리는 객체와 객체외의 선분과의 두께 차에 근거하여 형태학적인 연산을 이용함으로써 객체의 완전한 분리를 수행한다. 이러한 객체의 분리는 향후 객체의 벡터화 및 지능적인 인식을 통해 기계 도면의 검색에 효과적으로 이용될 수 있을 것이다. 둘째로, 객체가 제거된 객체 외의 선분을 벡터화한 후 벡터화된 선분으로부터 치수선을 인식한다. 이 때, 치수선의 인식은 화살표와 치수선간의 기하학적인 모델로부터 유도된 산술식에 의해 수행함으로써 치수선의 인식을 위한 알고리즘을 간단히 하였다.

본 논문은 다음과 같이 구성된다. 본 논문과 관련된 기존의 연구 및 본 논문에서 제안하는 알고리즘에 대한 설명은 II장에서 나타난다. 치수선의 인식을 위한 첫 단계인 기계 도면 요소의 분할은 III장에서 나타나며, 문자열의 추출, 객체, 화살표, 심볼들을 분리하는 방법들이 제안된다. 치수선 인식의 두 번째 단계인 설명선의 벡터화는 IV장에서 기술된다. 그리고, 치수선의 인식을 위한 방법들을 V장에서 기술하며, 이러한 방법들을 이용하여 기계 도면의 인식을 수행한 실험 결과 및 분석을 VI장에서 나타낸다. 마지막으로, VII장에서 본 논문에 대한 결론을 맺는다.

## II. 연구 배경

### 1. 기존의 연구

객체와 치수선의 분리 추출과 관련된 논문으로는 Lai의 연구가 있다<sup>[12]</sup>. Lai는 미국 표준 규격에 따라 그려진 기계 도면을 대상 도면으로 하였는데, 치수선의 분리를 위해 문자가 제거된 기계 도면을 먼저 골격화한다. 이러한 골격화 과정에는 화살표도 함께 골격화되어 골격화된 형태에 따라 화살표를 인식한다. 미국 표준 규격에서 사용되는 화살표는 가로 세로의 비가 2:1 이고 내부가 채워져 있으므로, 이러한 표준 화살표의 골격 패턴과 일치하는 도면상의 골격선을 화살표로 인식하게 된다. 그리고, 인식된 화살표와 인접한 골격선

끌격선들을 추적하여 치수선을 추출하는 것이 Lai 는 문의 핵심이다. 즉, Lai는 화살표와 인접한 선분들을 치수선의 후보로 선택한 후 선분이 끝나는 지점까지 추적하는 방법을 선택하였던 것이다.

이러한 방법의 문제점을 요약하면 다음과 같다. 첫째로, 이 방법은 기계 도면과 같이 선분의 교차가 많은 도면에서는 선 추적 알고리즘이 정교해져야 하며, 특히, 객체와 인접한 치수선을 추적하는 과정에서 객체의 일부분을 치수선으로 오추출하는 문제를 발생시킨다. 이러한 객체의 훼손은 객체의 벡터화 및 인식에 치명적인 영향을 끼칠수밖에 없게 된다. 둘째로, 도면을 끌격화한 후 끌격화된 형태에 따라 화살표를 인식하도록 되어 있어서 끌격화의 결과에 따라 화살표가 인식되지 못하는 경우가 발생한다. 특히, 설명선 및 객체와 겹치는 화살표의 경우에는 정확한 인식이 되지 못하며 이러한 화살표와 연결되는 치수선은 추출되지 못하게 된다. 셋째로, 연장선을 추출하기 위해서는 화살표와 연장선이 항상 인접해야만 선분 추적이 가능하며, 연장선이 화살표와 떨어져 있는 경우에는 연장선의 추출이 이루어지지 못할 것이다. 넷째로, 그래픽 데이터로 남아있는 추출된 치수선들을 벡터화하여 형태선, 꼬리선 및 연장선으로 인식해야 하는 과제가 남게 된다.

2. 제안된 알고리즘

본 논문에서는 기존의 방법과는 달리, 기계 도면의 주요한 요소인 객체와 화살표 및 심볼을 전처리 과정에서 먼저 분리한 후, 객체와 화살표 및 심볼이 제거된 설명선을 벡터화한 후 벡터화된 설명선으로부터 치수선을 인식한다. 치수선의 인식을 위한 전처리로서 객체 및 화살표를 원도면으로부터 분리하고, 도면 내에 존재하는 페루프 심볼을 제거한다. 그리고, 페루프 심볼들이 제거된 설명선을 세선화, 선분 벡터화 그리고 벡터 결합 과정을 통해 벡터화한다. 설명선의 벡터화가 수행되면 화살표, 꼬리선, 형태선 및 연장선에 대하여, 기하학적인 모델에 근거하여 유도된 산술적인 연산을 이용하여, 설명선으로부터 치수선을 인식한다(그림 3). 이 때, 본 논문에서 주로 다루는 부분은 그림 3에서 점선으로 블록지워진 영역이다.

제안된 방법은 선추적이 아니라 형태학적인 연산을 이용하여 객체를 분리함으로써 알고리즘이 간단하고, 설명선과 겹치는 화살표의 경우에도 정확하게 추출 및

인식할 수 있으며, 객체의 경우에 선분이 끊기는 문제는 발생하지 않았다. 이와 함께, 치수선의 인식을 산술적인 연산을 통해 수행함으로써 치수선의 인식을 위한 알고리즘이 간단하고 소요시간에 대한 부담이 거의 없는 이점이 있으며, 화살표와 인접하지 않는 연장선의 경우에도 정확하게 인식할 수 있었다. 마지막으로, 형태선, 꼬리선, 연장선의 벡터 정보와 화살표의 벡터 정보가 함께 저장됨에 따라, 문자 인식을 이용하여 문자를 인식할 경우 치수선의 CAD 데이터화가 용이하게 된다.

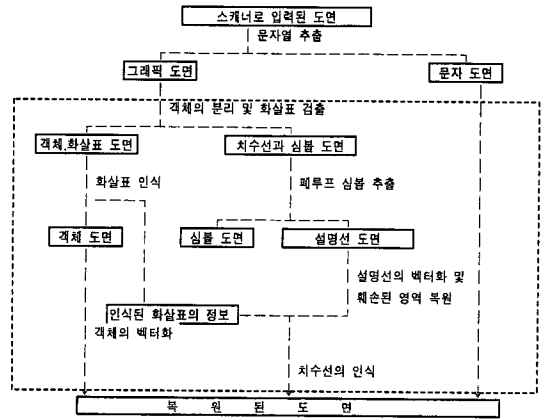


그림 3. 제안된 알고리즘  
Fig. 3. Proposed algorithm.

Ⅲ. 객체, 화살표, 심볼, 그리고 설명선의 분리

도면 인식을 위해서는 도면에 존재하는 문자 성분의 제거가 가장 선행되어야 할 과제이다. 본 논문에서는 이러한 문자의 분리 추출을 위하여 Lai가 제안한 방법 [12]을 이용하여 문자열을 분리 추출하였다. 이 방법은 문자 영역은 상대적으로 작은 크기의 고립 영역을 가지며, 고립 문자 영역은 정방형에 가까운 직사각형의 형태를 가진다는 특성을 이용하여 문자 영역을 분리 추출한다. 그리고, 추출된 문자를 다시 문자열로 인식함으로써 도면에 존재하는 점선등이 문자로 추출되지 않고 그래픽 영역에 남게 된다

문자가 제거되고 남은 그래픽 도면에는 객체, 화살표 및 설명선이 남게 된다. 객체는 굵은 실선으로 표시된 부분으로서 실제 3차원 기계 부품을 2차원 평면 상에 투영한 것으로 정의되며, 설명선은 이러한 객체를 설명하기 위한 꼬리선, 형태선, 연장선 등의 치수선

과 점선 등의 보조적인 선분 및 심볼들로 객체에 비해 상대적으로 가는 선분들이다. 이 중, 그래픽 도면에서 고립 영역으로 존재하는 심볼들은 원 및 원추형의 심볼들로써 문자의 분리 추출과 같은 방법을 이용하여 추출한다.

KS 기계 제도 규격(KS B 0001 5)<sup>[13]</sup>에 따른 선의 굵기 표준은 표 1과 같다. 표 1에서 알 수 있듯이 객체를 나타내는 외형선은 실선으로 0.8~0.4mm의 굵기를 가지고, 객체외의 꼬리선, 형태선, 연장선 등은 실선으로 0.3mm 이하의 굵기로 제정되어 있다. 이 때, 선분의 굵기는 도면의 종류에 따라 다르게 규정되지만, 일반적으로 도면의 종류에는 관계없이 객체와 치수선의 두께 비는 최소 2.67배 이상의 차이를 가지게 된다. 따라서, 이러한 선의 두께 차이를 고려하여 형태학적 연산을 적용함으로써 간단하면서도 효과적으로 객체를 분리 추출한다.

표 1. 도면의 종류에 따른 선의 굵기  
Table 1. Thickness of lines according to the kind of drawings.

도면의 종류 선의 종류	1호 (큰도면)	2호 (보통도면)	3호 (작은도면)
외형선	0.8 mm	0.6 mm	0.4 mm
꼬리선	0.3 mm	0.2 mm	0.1 mm
형태선	0.3 mm	0.2 mm	0.1 mm
연장선	0.3 mm	0.2 mm	0.1 mm

객체의 분리 추출을 위한 첫 단계는 OPENING 연산<sup>[14]</sup>을 이용하는데, 이 단계를 거치면 평활화된 형태의 두꺼운 영역만이 남게 되며 이를 다시 원 영상과 LOGICAL AND 연산을 시키면 두꺼운 영역인 객체와 화살표만으로 된 도면을 얻을 수 있다. 이러한 객체와 화살표 도면에는 형태학적 연산의 수행 과정에서 설명선간의 교차점이 잡영으로 나타날 수 있는데 이러한 잡영들은 화살표보다 작은 고립 영역으로 구성되어 있으므로 화살표의 분리 추출 과정에서 제거된다. 이와 같이 화살표와 잡영이 제거되고 나면 원하는 객체를 분리할 수 있다. 마지막으로, 분리 추출된 객체와 화살표만으로 구성되어 있는 도면과 원 도면과의 NAND 연산을 이용하여 객체와 화살표가 제거된 설명선 도면을 얻는다. 이러한 알고리즘으로 객체를 분리 추출할 경우 객체는 잘 분리되지만, 설명선의 경우 객체와 겹치는 부분에서 선분의 끊어짐과 같은 훼손이

발생한다. 그러나, 이러한 훼손은 설명선의 벡터화 과정에서 분리 추출된 객체 정보를 이용하여 복구할 수 있다.

객체와 화살표가 제거되고 남은 설명선에는 피쳐 제어 프레임의 심볼이 존재한다. 이러한 피쳐 제어 프레임은 다음과 같은 방법으로 분리한다. 먼저 화살의 0/1 반전 영상을 만들면 페루프 안은 고립 영역이 된다. 그런 다음 영역 레이블링 알고리즘<sup>[15]</sup>을 이용하여 고립 영역이 심볼 성분의 후보가 되게 한다. 하지만, 후보로 선정된 고립 영역에는 실제 심볼뿐 아니라 설명선이 교차하면서 발생한 페루프 역시 심볼 후보로 선정된다. 따라서 선정된 각 후보 영역에 대해 특징 파라메타를 적용하여 실제의 심볼 성분만을 분리해 낸다. 이 때, 특징 파라메타는 면적대 화소의 비를 사용하여 그 비가 1:1에 가까운 심볼을 피쳐 제어 프레임으로 분리 추출한다.

#### IV. 설명선의 벡터화

##### 1. 선분 벡터화

객체, 화살표 및 심볼이 제거된 설명선의 벡터화를 수행하기 위해서는 세선화 및 선분 벡터화가 먼저 수행되어야 한다. 본 논문에서는 선분 벡터 추출을 위한 세선화 과정에 Chen-Hsu<sup>[16]</sup>의 세선화 기법을 사용한다. Chen-Hsu의 세선화 알고리즘은 2개의 부루프(sub-loop)를 가지는 병렬 알고리즘으로, 완벽한 8-방향 세선화 결과를 보인다. 이 기법은 경계 잡음에 강하고, 심각한 선의 잡식과 선 연결성 문제를 해결한 알고리즘이다.

세선화가 끝난 설명선을 선분 벡터화 하기 위해 Ramer의 곡선 근사화 방법<sup>[17]</sup>을 이용하는데 이는 곡선의 선분 근사화 알고리즘으로 매우 간단하며, 그 적용이 용이하다. 이 방법을 이용하여 선분 벡터를 추출하기 위한 알고리즘은 다음과 같다. 첫 번째 단계로 세선화된 도면의 골격선을 따라 3×3 윈도우를 씌워 가며 윈도우의 8-이웃 화소 중 흑화소의 개수를 계산한다. 8-이웃 화소 중 흑화소가 하나 뿐인 화소는 선분의 끝점인 최종점(end point)으로, 8-이웃 화소 중 흑화소가 3개 이상인 화소는 선분과 선분이 만나는 분기점(branch point)으로 표시해 둔다. 그런 다음 최종점 또는 분기점과 다른 최종점 또는 분기점 사이의 화소들의 체인을 구해 이 체인의 시작과 끝을 연결하는

벡터를 생성한다. 두 번째로 체인 상의 각 점과 벡터 사이의 거리를 구해 벡터로부터 가장 멀리 떨어져 있는 점을 구한다. 세 번째로 그 거리가 미리 정해진 문턱값 보다 크면, 벡터는 그 점에서 두 개로 분리된다. 만약 거리가 문턱 값보다 작거나 같으면, 벡터는 세선화된 영상을 근사화한 결과라 간주한다. 이 때, 사용된 문턱 값은 1.0이다. 모든 도면내의 체인에 대해 위의 과정을 반복함으로써 선분 벡터의 추출을 완료한다. 이상의 반복적인 선근사화 과정을 그림 4에서 보여주고 있다. 그림 4의 예는 한 개의 곡선이 세 개의 벡터로 근사화되는 것을 나타낸다.

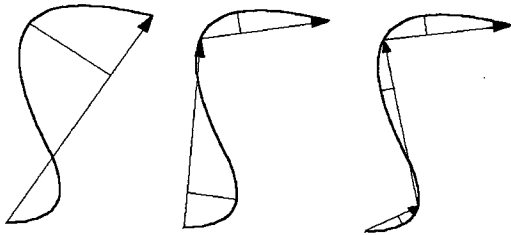


그림 4. Ramer의 방법 이용한 선분 벡터화  
Fig. 4. Vectorization of line using Ramer's method.

2. 벡터화된 설명선의 결합과 객체 추출시 훼손된 설명선 복구

직선 및 원호 벡터들의 결합을 위해 우선 벡터화된 설명선에 존재하는 원호 성분을 추출 및 인식함으로써 직선 벡터의 결합을 용이하게 한다. 원호 벡터의 추출 및 인식을 위해 본 논문에서는 추출된 선분 벡터의 연결 형태를 살펴봄으로써 원호를 인식한다. 선분 벡터로부터 원호를 추출하기 위한 과정은 원호의 후보가 되는 벡터를 발견하고, 후보 벡터로부터 원호 정보를 구하는 두 단계로 구성된다. 그림 5에서와 같이 원호의 중심점, 반지름, 시작점, 끝점, 방향을 원호 정보로 저장한다.

선분 벡터가 원호의 후보 벡터로 선택되기 위해서는 선분 벡터의 기하학적 형태가 원호에 근접해야 한다. 이를 위해 원호의 후보 벡터가 되기 위해서는 그림 6에서 알 수 있듯이 절점이 최소한 3개 이상인 다중 선분 벡터여야 하며, 선분과 선분이 만나는 각도는 예각이어야 하고, 인접하는 선분의 길이가 비슷해야 한다. 선분 벡터들 중에서 위의 조건을 만족하는 벡터를 원호의 후보로 선택한 후 원호 정보를 구해 원호로 인식한다.

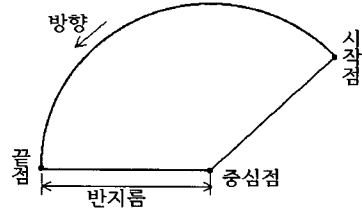


그림 5. 필요한 원호 정보  
Fig. 5. Required arc information.

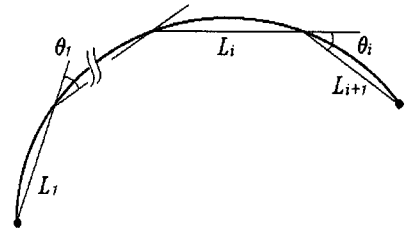


그림 6. 원호 후보 벡터를 구하기 위한 파라메타  
Fig. 6. Parameter for finding arc-candidate vector.

원호 성분이 제거되면 설명선에는 직선 벡터 성분들만 남게 된다. 이 때, 직선 벡터들은 동일 선상에 있는 직선임에도 불구하고 몇 화소 정도가 떨어진 일련의 벡터들로 되는 경우가 생기므로 벡터 결합 (clustering) 과정을 통해 최소, 최적의 선분 벡터로 추출해야 한다<sup>[18]</sup>. 그림 7과 같이 인접한 선분 벡터들의 각도 ( $\theta_1, \theta_2$ ), 길이 ( $L_1, L_2$ ), 그리고 선분 벡터간의 거리 D를 구하여 각도 차와 길이비, 그리고 거리 D가 문턱값보다 작으면 두 벡터를 결합하여 준다. 이러한 과정을 더 이상의 결합될 벡터가 없을 때까지 반복 수행함으로써 벡터 결합을 완료한다. 이 때, 각도 차는 10도, 길이비는 2:1이내, 그리고 거리 D는 8.0을 문턱 값으로 선택하였다.

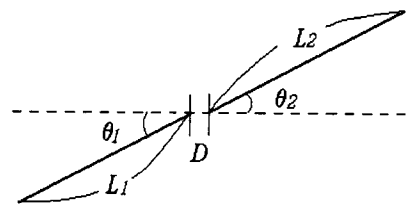


그림 7. 인접한 두 선분 벡터  
Fig. 7. Adjacent two line-vectors.

결합이 끝나면 객체 분리 추출시 훼손된 설명선을 복구하는데 이는 객체가 존재하는 영역을 탐색 영역으로 지정하여 영역 내에 존재하는 설명 선분 벡터간의 거리가 객체의 두께와 큰 차이가 없는 경우 두 벡터를 결합하는 것으로 이 과정이 수행되고 나면 설명선의 최종 벡터화가 이루어진다. 본 논문에서 객체는 평균 7-8의 두께를 가지므로 훼손된 설명선의 복원을 위해 사용된 문턱 값은 10.0을 이용하였다.

### V. 치수선의 인식

#### 1. 화살표의 인식

일반적으로 표준 규격의 도면에서는 화살표의 가로 세로 비가 2:1이고 내부가 채워진 형태로 되어 있으며, 화살표의 크기는 동일하다<sup>[8]</sup>. 본 논문에서는 이러한 화살표의 무게 중심과 방향 벡터를 구해 화살표의 인식 및 치수선의 인식에 이용한다.

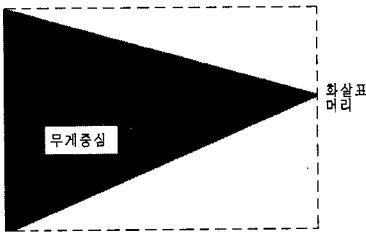


그림 8. 표준 화살표와 화살표의 최소 외접 사각형  
Fig. 8. Standard arrowhead and MBR of arrowhead.

그림 8과 같이 화살표의 무게 중심은 화살표의 꼬리(arrowback) 부분에 존재하고, 화살표의 머리(arrowtip)는 화살표 최소 외접 사각형의 경계 좌표 위에 존재하는 혹은 화소 중 무게 중심으로부터 가장 멀리 떨어진 점에 존재한다. 따라서 화살표의 무게 중심과 머리가 결정되면 이 두 점을 연결하는 벡터를 화살표의 방향 벡터로 정의한다. 제한하는 화살표의 인식 과정은 다음과 같은 순서로 이루어진다.

- ① 도면 내에 존재하는 모든 화살표 후보의 무게 중심과 방향 벡터를 구한다.
- ② 방향 벡터가 0도인 화살표들의 평균 폭과 높이를 이용하여 가로 세로의 비가 2:1 인 화살표의 기본 패턴을 만든다.
- ③ 방향 벡터의 각도만큼 회전시킨 화살표의 패턴과

- 도면 내의 화살표 후보와의 정합도를 고려하여 정합도가 75% 이상이면 화살표로 인식한다.
- ④ 인식된 화살표의 경우에는 무게 중심과 화살표의 머리 정보를 함께 저장한다.

#### 2. 꼬리선과 형태선의 인식

일반적인 기계 도면에서 꼬리선 및 형태선은 화살표의 방향 벡터와 나란하고 연장선은 화살표의 방향 벡터와 수직인 특성을 가진다. 본 논문에서는 이러한 화살표, 꼬리선, 형태선, 그리고 연장선간의 기하학적인 특징을 이용하여 치수선을 인식한다. 그리고 인식된 결과를 토대로 두 개의 화살표를 쌍으로 가지는 선분을 형태선으로 인식하고 나머지 선분을 꼬리선으로 결정한다.

일반적으로 치수선을 구성하는 선분에는 직선 성분과 원호 성분의 두 가지 형태가 나타나므로 각각에 대하여 인식을 수행해야 한다. 먼저, 설명선의 직선 벡터들 중에서 꼬리선과 형태선의 인식은 다음과 같은 순서로 이루어진다.

- ① 설명 선분 벡터중 화살표의 방향 벡터와 동일한 방향을 가지는 벡터를 후보로 선정한다.
- ② 그림 9와 같이, 후보 벡터들 각각에 대하여 화살표의 무게 중심 G와 후보 벡터의 한 끝점 L사이의 거리 D를 구한다.
- ③ 거리 D가 문턱 값 이내의 선분들을 꼬리선 또는 형태선의 후보로 선택한다. 이 때 문턱 값은 12.3이다.
- ④ 선택된 벡터중 화살표를 쌍으로 가지는 것을 형태선으로 나머지를 꼬리선으로 인식한다.

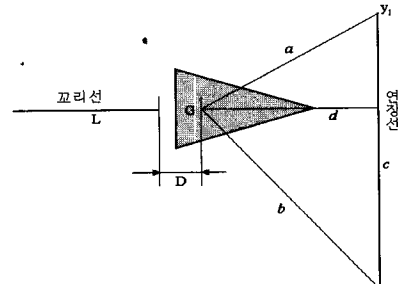


그림 9. 화살표, 꼬리선 및 연장선의 관계에 대한 기하학적인 모델  
Fig. 9. Geometric model for relation of arrowhead, tail and witness.

다음으로, 설명선의 원호 벡터중에서 꼬리선 및 형태선을 인식한다. 일반적으로 화살표와 이 화살표의 꼬리선 및 형태선이 되는 원호 벡터는 동일한 원호 상에 존재하게 되며 이 때, 중심점 및 반지름은 원호 벡터의 중심점 및 반지름 정보와 일치한다. 따라서, 그림 10과 같이 화살표의 무게 중심과 원호벡터의 중심점과의 거리L과 원호 벡터의 반지름 R과의 거리차의 절대값을 구해 이 값이 정해진 문턱값 이내에 들어오면 이를 꼬리선 및 형태선의 후보로 결정한다. 그런 다음 직선의 경우와 마찬가지로 후보 벡터와 연결된 화살표의 개수를 이용하여 각각을 꼬리선과 형태선으로 인식한다.

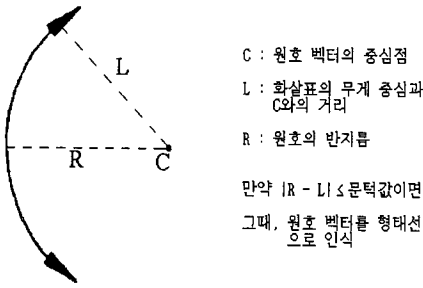


그림 10. 원호 벡터 중에서 형태선의 인식  
Fig. 10. Recognition of shape line out of arc-vectors.

3. 연장선의 인식

꼬리선 및 형태선이 인식되고 나면 다음과 같이 도형 방정식을 이용하여 연장선을 인식한다. 일반적으로 연장선의 후보 벡터로는 그림 9에서처럼 화살표의 방향 벡터에 대하여 직교하는 벡터들이다. 따라서, 화살표의 인식 과정에서 구한 무게 중심의 좌표를 G, 후보 벡터의 양 끝 좌표를 y1, y2라고 각각 정의한다. 그리고, 이 세 점으로 이루어지는 삼각형의 세 변을 그림 9에서와 같이 각각 a, b, c로 정의하고 변 b와 c 사이의 각을  $\theta$ 라 할 때 화살표의 무게 중심 G와 연장선간의 거리 d는

$$d = b \cos(90^\circ - \theta)$$

$$= b \sin \theta$$

, 여기서  $\sin \theta = \sqrt{1 - \cos^2 \theta}$

$$\cos \theta = \frac{b^2 + c^2 - a^2}{2bc}$$

이다. 이렇게 구해진 거리 d가 가장 작으면서 문턱 값 이내에 들어오는 벡터를 연장선으로 인식한다. 이 때, 사용된 문턱 값은 35.0을 이용하였다.

VI. 실험 및 고찰

본 실험의 실험 영상은 KS 기계 제도 규격에 따라 그려진 실제 자동차 부품 도면을 사용하였다. 영상 입력 장치는 인치당 300화소를 가지는 스캐너로 도면을 이진화하여 사용하였다. 그림 11-(a), 그림 12-(a)는 실험에서 사용한 도면으로 각각 2208×2078, 3123×1489의 해상도를 가진다. 이 도면들은 객체, 심블, 화살표, 치수선 및 보조적인 선분들을 포함하는 일반적인 기계 도면들의 특성을 가지고 있다. 도면 처리의 첫 번째 과정인 문자 분리를 위한 척도는 가로 및 세로의 폭이 55 x 55가 되게 하였으며, 실험 결과 효과적인 추출이 가능하였다. 하지만, 추출된 문자에는 피쳐 제어 프레임내에 존재하는 그래픽 성분이 문자로 추출된다. 이러한 그래픽 성분은 향후 문자의 인식 과정에서 그래픽 성분으로 인식 되어야할 것이다. 이와 함께, 그래픽 영역과 인접하는 문자의 경우 현재의 문자 추출 알고리즘으로는 분리가 되지 않으며 이러한 경우의 문자 분리에 대한 연구가 필요할 것이다.

문자가 제거된 그래픽 도면에서 객체 및 화살표의 분리 추출을 위해 OPENING 연산을 이용하였는데, 그림 11-(b), 그림 12-(b)는 OPENING 연산을 이용하여 객체, 화살표 및 심블을 추출한 결과를 나타낸다. 이 때, OPENING 연산을 위한 수축과 팽창을 위한 마스크는 각각 5 x 5 와 7 x 7 크기의 창을 이용하여 연산의 횟수는 각각 1회를 수행하였다. 기존의 방법에서는 화살표를 인식하고 이와 연결된 선분을 추적하여 제거한 후 남은 영역을 객체로 분리하는데 이 방식은 선 추적 알고리즘이 복잡하고 객체와 교차하거나 접하는 치수선이 객체로 오추출되는 문제점이 있다. 이에 반해 제안된 방법은 알고리즘이 훨씬 간단하고 객체의 안정된 추출이 보장되는 장점이 있지만 제도 표준에 따라 그려진 도면이 아니고 자의적으로 그려진 도면의 경우 새로운 분리 추출 알고리즘의 개발이 필요할 것이다. 제안된 방법에 의한 객체의 분리 추출시 두 가지 문제가 발생하였다. 하나는 OPENING 연산의 불가피한 결과로서 객체와 겹치는 설명선의 일부가 끊어지는 것인데, 이러한 문제는 설명선의 벡터화 과

정에서 객체의 정보를 이용하여 복구가 가능하였다. 다른 하나는 설명선이 서로 교차하는 부분의 일부가 객체 영상에서 잡음으로 나타난다는 점이다. 이러한 잡음 성분은 객체에 비해 상대적으로 작은 크기의 고립 영역이므로 간단히 최소 외접 사각형을 이용하여 제거하였다. 이상의 과정을 거치면 최종적으로 두꺼운 성분인 객체와 화살표가 추출되고, 이렇게 추출된 객체와 화살표를 문자가 제거된 그래픽 도면과 NAND 연산을 수행하면 설명선과 페루프 심볼이 남게 된다.

영상 분할의 마지막 단계는 피쳐 제어 프레임의 분리 추출이다. 실험 결과, 효과적인 추출이 가능하였지만, 설명선과 연결되어 있지 않은 피쳐 제어 프레임의 경우 원 및 원추형의 심볼을 분리하는 과정에서 같이 추출되게 된다. 따라서, 추출된 원 및 원추형 심볼로 구성된 영상에 대하여 피쳐 제어 프레임의 추출을 위한 알고리즘을 적용함으로써 피쳐 제어 프레임을 완벽하게 추출할 수 있었다(그림 11-(b), 그림 12-(b)).

페루프 심볼 및 피쳐 제어 프레임이 제거된 설명선은 Chen-Hsu의 알고리즘으로 세선화 한 후 선분 근사화 방법을 이용하여 선분 벡터를 추출한다. 추출된 선분 벡터들 중에서 원호 벡터를 인식하고 동일 선상의 일련의 직선 벡터들을 하나의 벡터로 결합하는 과정을 거치면 최종적인 설명선의 벡터화가 이루어진다(그림 11-(c), 그림 12-(c)). 결합 과정을 거친 최소, 최적의 벡터 결과가 표 2와 같다. 인식된 원호 벡터의 개수는 원호의 인식 및 결합 과정을 통해 최종 벡터화된 결과를 나타낸다. 그리고, 직선 벡터의 경우 결합 전 벡터의 수는 근사화 방법에 의해 분할된 벡터중 원호의 인식을 위해 사용된 벡터를 제외한 나머지 직선 벡터의 개수를 나타낸다. 표 2에서 나타난 것처럼 결합후의 벡터의 개수는 상당히 줄어들게 되며 이러한 벡터화의 결과는 벡터의 저장을 위한 공간을 줄일 수 있는 이점이 있다. 이러한 벡터화 과정에는 객체와 겹치는 설명선의 복원도 함께 수행되며 그 결과가 표 3에 나타난다. 표 3에서 복원이 되지 못한 벡터는 원이나 타원을 구성하는 벡터들이다. 이러한 벡터들의 경우 객체에 의해 분할된 원의 작은 조각이 원호로 인식되지 못하고 직선 성분으로 남아있게 되어 원호의 결합과정에서 결합되지 못하기 때문이다. 이러한 문제들은 향후 CAD 데이터화된 데이터의 편집 과정에서 해결되어야 할 것이다. 그리고, 본 논문에서는 화살표에 의해 훼손된 설명선의 경우에도 벡터 결합 과정에서

화살표의 정보를 이용하여 효과적인 복원이 가능하였다.

표 2. 설명선의 최종 벡터화 결과

Table 2. Final vectorization result of interpretation line.

		도면 1 (2208×2078)	도면 2 (3123×1489)
		직선 벡터	264개
	결합 후	87개	88개
인식된 원호 벡터		11개	6개

표 3. 객체와 겹치는 설명선의 복원율

Table 3. Reconstruction rate of interpretation line adjacent with object.

		도면 1 (2208×2078)	도면 2 (3123×1489)
		복원되어야 할 벡터의 수	15개
복원되지 못한 벡터의 수		2개	1개

형태선, 꼬리선 및 연장선의 치수선 인식 결과가 표 4에 나타난다. 본 논문에서는 직선 및 원호 벡터들의 정보를 이용한 신술적인 계산으로 이러한 치수선을 인식한다. 실험에서는 화살표와 연결된 설명선 중에서 화살표의 방향 벡터와 수평인 것을 꼬리선과 형태선으로(그림 11-(d), 12-(d)), 수직인 것을 연장선으로(그림 11-(e), 12-(e)) 인식하였다. 하지만, 화살표가 객체와 붙어 있는 경우에는 화살표의 인식자체가 불가능하므로 이 경우 이러한 화살표와 연결된 치수선은 제대로 인식될 수 없었다. 도면 2의 경우 형태선 및 꼬리선이 인식되지 못한 결과가 나타나는데, 이는 화살표가 객체와 붙어 있어 화살표의 인식이 이루어지지 못했기 때문에 치수선은 인식될 수 없었다.

표 4. 형태선 및 꼬리선, 연장선의 인식 결과

Table 4. Recognized result of shape line, tail line, and extension line.

		도면 1 (2208×2078)	도면 2 (3123×1489)
		형태선 및 꼬리선	인식된 개수
	인식되지 못한 개수	0	2
연장선	인식된 개수	24	28
	인식되지 못한 개수	0	0

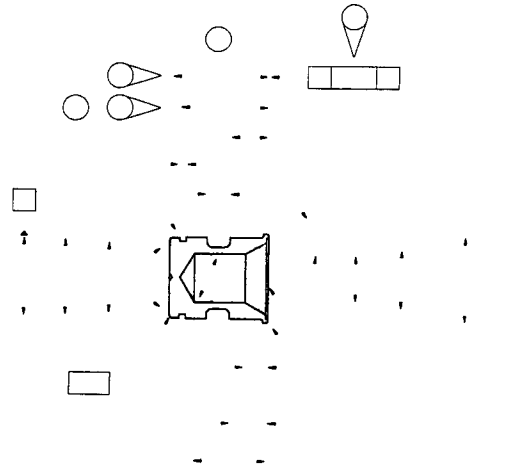


제안한 시스템의 성능을 평가하기 위하여, 알고리즘 수행에 필요한 소요 시간을 표 5에서 나타내었다. 표 5의 결과는 펜티엄 150MHz의 PC환경에서 GCC 컴파일러를 이용하여 실험한 결과이다. 실험 결과, 벡터화의 전처리 과정에서 많은 시간이 소요되지만, 치수선의 인식에서는 적은 시간이 소요됨을 보여주고 있다. 그리고, 문자열 및 심볼의 분리에서는 고속의 영역 레이블링을 이용함으로써 처리 시간이 거의 들지 않음을 보여 주고 있다. 하지만, 형태학적인 연산의 수행과정에서 많은 시간이 소요되는데, 이는 도면 입력을 위한 스캐닝 해상도와 수축 및 팽창을 위한 창의 크기가 커질수록 더욱 커질것이다.

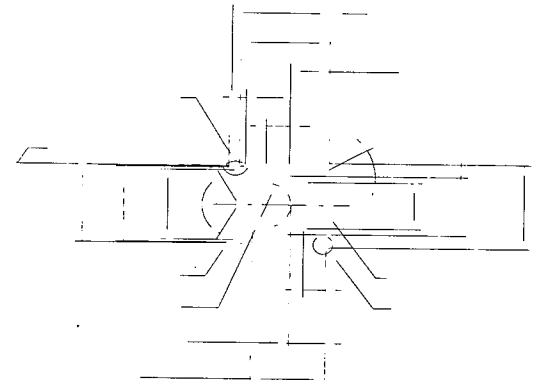
표 5. 각각의 알고리즘을 처리하는데 소요되는 시간

Table 5. Time for processing of each algorithm.

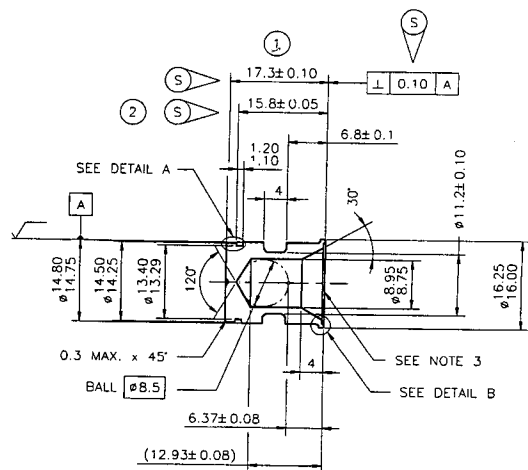
단 계	2208 x 2078 도면	3123 x 1489 도면
문자열의 분리	3.01(s)	4.01(s)
형태학적 연산을 이용한 객체, 화살표 및 치수선의 분리	40.66(s)	42.98(s)
피쳐 제어 프레임 및 페루프 심볼의 추출	4.10(s)	4.68(s)
치수선 및 객체의 세선화	11.98(s)	14.01(s)
치수선 및 객체의 벡터화	11.64(s)	11.47(s)
치수선의 인식	1.09(s)	1.20(s)
전체 처리 시간	72.78(s)	78.35(s)



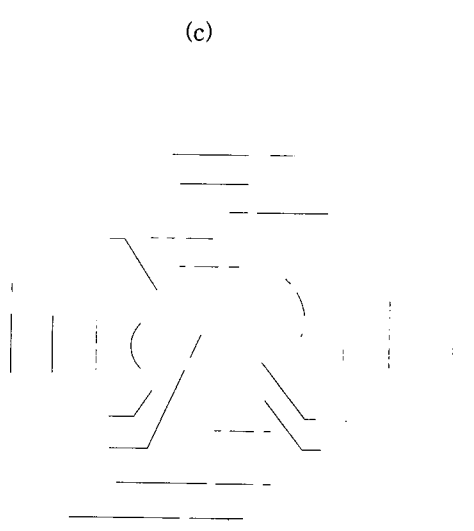
(b)



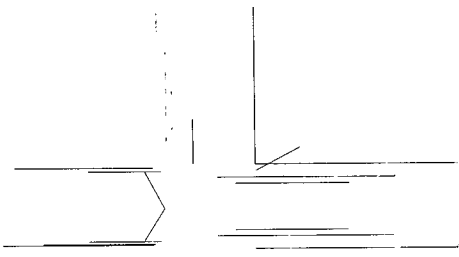
(c)



(a)



(d)



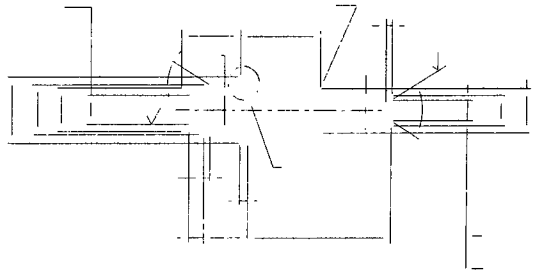
(e)

그림 11. 실험에 사용한 기계 도면의 예(2208×2078 화소)

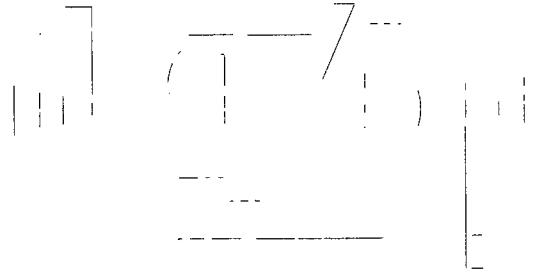
(a) 입력된 도면 (b) 화살표, 객체 및 심볼  
(c) 벡터화된 설명선 (d) 인식된 형태선 및 꼬리선 (e) 인식된 연장선

Fig. 11. An example of mechanical drawing(2208×2078 pixel).

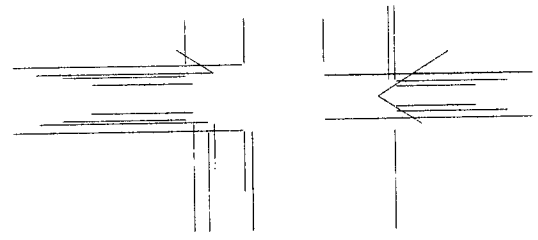
(a) Original drawing. (b) Arrowheads, object and symbols. (c) Result of vectorizing interpretation line. (d) Recognized shape and tail lines. (e) Recognized extension lines.



(c)



(d)



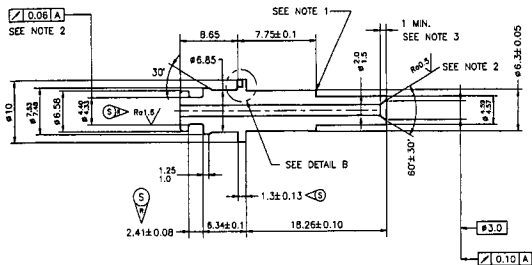
(e)

그림 12. 실험에 사용한 기계 도면의 예(3123×1489 화소)

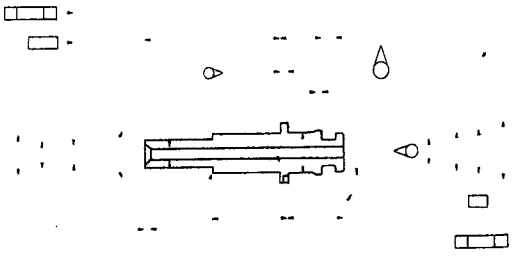
(a) 입력된 도면 (b) 화살표, 객체 및 심볼  
(c) 벡터화된 설명선 (d) 인식된 형태선 및 꼬리선 (e) 인식된 연장선

Fig. 12. An example of mechanical drawing(3123×1489 pixel).

(a) Original drawing. (b) Arrowheads, object and symbols. (c) Result of vectorizing interpretation line. (d) Recognized shape and tail lines. (e) Recognized extension lines.



(a)



(b)

VII. 결론

본 논문에서는 기계 도면의 자동 입력을 위해 도면의 주요 성분인 객체를 형태학적 연산을 이용하여 분리 추출한 후 추출된 객체와 설명선을 벡터화하여 치

수선을 인식하고 최종적으로 치수선의 인식을 수행하였다. 제안된 방법의 이점은 형태학적 연산을 이용함으로써 객체의 분리 추출 알고리즘이 간단해지고, 객체 및 설명선을 선분 벡터 추출과 벡터 결합 알고리즘을 이용하여 최종 벡터화함으로써 도면의 저장 및 관리가 용이하다. 또한, 화살표, 꼬리선, 형태선, 연장선 간의 기하학적인 모델링과 수학적 연산을 이용하여 치수선을 인식함으로써 알고리즘이 간단하며 인식된 결과는 도면의 자동 입력뿐 아니라 효율적 검색 등의 다양한 분야에 이용할 수 있다. 실험 결과, 제안된 방법은 제도 표준에 따라 그려진 도면에 대해서는 최소의 손상으로 객체의 효율적인 분리 추출이 가능하였으며 추출 과정에서 설명 선의 훼손된 부분은 객체의 위치 정보를 이용하여 정확한 복구가 가능하였다. 그리고 각 화살표에 연결된 꼬리선, 형태선, 연장선이 정확하게 인식되었다. 향후 연구 방향으로는 그래픽 영역과 겹쳐진 문자 영역의 분리 방법과 표준을 따르지 않고 자의적으로 그려진 도면에 대한 새로운 분리 추출 알고리즘의 개발이 필요할 것이다.

#### 참 고 문 헌

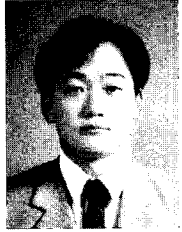
- [1] D. Dori, "A syntactic/geometric approach to recognition of dimensions in engineering machine drawings", *Computer Vision Graphics And Image Processing*, Vol. 47, pp. 271-291, 1989.
- [2] Nagasamy and N.A.Lagrana. "Engineering drawing processing and vectorization system", *Computer Vision, Graphics, and Image Processing*, Vol. 49, pp. 379-397, 1990.
- [3] D. Dori, Y. Liang, J Dowell and I. Chai, "Spare Pixel Recognition of Primitives in Engineering Drawings", *Machine Vision and Applications*, pp. 69-82, 1993.6.
- [4] Yuan Chen, Noshir A.Lagrana, and Atish K.Das, "Perfecting Vectorized Mechanical Drawings", *Computer Vision and Image Understanding*, Vol. 63, No.2, pp 273-286, 1996.3.
- [5] Rik D.T.Janssen and Albert M.Vossepoel, "Adaptive Vectorization of Line Drawing Images", *Computer Vision and Image Understanding*, Vol. 65, No.1, pp. 38-56, 1997.1.
- [6] D.Dori, "Vector-Based Arc Segmentation in the Machine Drawing Understanding System Environment", *IEEE Transactions on PAMI*, Vol. 17, pp. 1057-1068, 1995.
- [7] D.Dori and Liu W., "Vector-Based Segmentation of Text Connected to Graphics in Engineering Drawings", *Lecture Notes in Computer Science*, Vol. 1121, pp. 322-331, 1996.
- [8] Luca Boatto, Vincenzo Consorti, "An Interpretation System for Land Register Maps", *IEEE Computer*, Vol. 25, No. 7, pp.25-33, 1992.6.
- [9] Liu Wenyin, Dov Dori, "Recognition of Hatching Lines in Engineering Drawings", *Proc. 13th Israeli Symposium of AI, CV, and NN*, 1997.2.
- [10] S. Y. Lee and F. J. Hsu, "2D-String : a new spatial knowledge representation for image database systems", *Pattern Recognition*, Vol. 23, No. 10, pp. 1077-1087, 1990.
- [11] P. Vaxiviere and K. Tombre, "Celesstin : CAD conversion of mechanical drawings", *Computer*, Vol. 25, No. 7, pp. 46-54, Jul., 1992.
- [12] C. P. Lai and R. Kasturi, "Detection of dimension sets in engineering drawing", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 8, Aug.,1994.
- [13] 송삼홍, "표준 기계 제도", 문운당, 1994
- [14] Ramesh Jain, Rangachar Kasturi, Brian G.Schunck, "Machine Vision", *MIT Press and McGraw-Hill. Inc*, 1995.
- [15] 백영목, 임길택, 김우태, 진성일, "영역 레이블링 방식을 이용한 일반문서의 layout understanding", 제 6회 영상 처리 및 이해에 관한 워크샵 논문지, 1994.
- [16] Y. S. Chen and W. H. Hsu, "A modified fast parallel algorithm for thinning digital patterns", *Pattern Recognition Letters*, vol. 7, no. 2, pp. 99-106, 1988.
- [17] Urs Ramer, "An Iterative Procedure for the Polygonal Approximation of Plane Curves",

CGIP, pp. 244-256, 1972.

axiomatic approach to clustering line-segments", *ICDAR*, pp. 386-389, 1995.

[18] A. Jonk and A. W. M. Smeulders, "An

— 저 자 소 개 —



鄭 閔 守(正會員)

1969년 12월 23일생. 1993년 경북대학교 전자공학과 졸업(공학사). 1995년 경북대학교 전자공학과 석사과정 졸업(공학석사). 현재, 경북대학교 전자공학과 박사과정. 주관 분야는 영상신호처리 및 패턴인식, 영상 압축

朴 吉 欽(正會員) 第 34卷 S編 第 2號 參照

식, 영상 압축