

論文97-34S-11-13

# 기계 도면의 자동 입력을 위한 치수 집합의 인식 및 분류

## (Recognition And Classification of Dimension Set for Automatic Input of Mechanical Drawings)

鄭閔守\*, 朴吉欽\*

(Yoon-Su Jung and Kil-Houm Park)

### 요 약

본 논문에서는 기계 도면으로부터 치수 집합을 인식하고, 인식된 치수 집합을 기능적인 목적에 따라 6종류의 치수 양식으로 분류 및 CAD 데이터화하는 방법을 제안한다. 제안된 방법은 문자열이 제거된 기계 도면으로부터 객체 및 페루프 심볼을 추출한 후, 객체와 설명선을 벡터화한다. 그리고 화살표, 치수선, 문자열 및 피쳐 제어 프레임을 구성 요소로 하는 치수 집합을 인식한 후, 인식된 치수 집합을 수직 치수, 수평 치수, 각 치수, 직경 치수, 반경 치수, 리더 치수 양식으로 분류한다. 마지막으로, 분류된 치수 집합은 AutoLisp 언어를 이용하여 CAD 데이터로 변환한다. 제안된 방법은 벡터화된 설명선에 대해 기하학적인 모델링을 이용함으로써 치수 집합의 인식 및 분류가 간단해지는 이점이 있다. 제안된 방법을 KS 제도 표준에 따라 그려진 도면에 적용한 실험 결과가 제공된다.

### Abstract

This paper presents a method that automatically recognizes dimension sets from the mechanical drawings, and that classifies 6 types of dimension sets according to functional purpose. In the proposed method, the object and closed-loop symbols are separated from the character-free drawings. Then object lines and interpretation lines are vectorized. And, after recognizing dimension sets (consisting of arrowhead, shape line, tail lines, extension lines, text-string, and feature control frame), we classify recognized dimension sets as horizontal, vertical, angular, diametral, radial, and leader dimension sets. Finally the proposed method converts classified dimension sets into AutoCAD data by using AutoLisp language. By using the methods of geometric modeling, the proposed method readily recognizes and classifies dimension sets 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 도면

의 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 도면의 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.

\* 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: 1997年3月8日, 수정완료일: 1997年9月29日

일반적으로 기계 도면의 선분 구성은 다음과 같다. 3차원 기계 부품의 직교 투영을 나타내는 객체(또는 객체 선분)와 이러한 객체의 정의를 제공하는 치수선, 보조적인 선분 및 심볼등의 설명선으로 구성된다<sup>[1]</sup>. 이러한 선분 중에서 치수선은 화살표 및 치수 문자와 함께 치수 집합으로 정의되며, AutoCAD 포맷에서는 치수 집합 전체가 하나의 벡터로 표현된다. 이와 함께, 치수 집합은 6종류(수직 치수, 수평 치수, 각 치수, 직경 치수, 반경 치수, 리더 치수)의 독립된 벡터 표현 양식을 가진다. 따라서, 이러한 치수 집합의 인식 및 6 종류의 치수 양식으로의 분류는 기계 도면의 자동 입력을 위해 중요한 과제이다.

이러한 치수 집합의 인식 및 치수 집합의 문법적인 분류와 관련된 연구로는 Lai<sup>[2]</sup>와 Dori<sup>[1]</sup>의 연구가 있다. Lai의 연구는 화살표와 인접한 치수선들에 대한 선 추적 알고리즘을 이용하여 치수선을 포함하는 치수 집합의 요소를 추출한다. 이 방법은 기계 도면과 같이 선분의 교차가 많은 도면에서는 선 추적 알고리즘이 정교해져야 하며, 객체와 인접한 치수선을 추적하는 과정에서 객체의 일부분을 치수선으로 오추출하는 등의 문제를 발생시킨다. 이와 함께, 치수 집합의 인식 및 분류에 대한 방법들은 제시하지 못하고 있다.

한편, Dori의 연구는 치수 집합의 문법적인 표현에 관하여 다루고 있다. 하지만, 이 연구는 기계 도면에서 사용되는 치수 집합은 상세하게 분류하고 있지만, 실제 기계 도면에서 이러한 치수 집합의 분류를 어떻게 수행하는지는 다루고 있지 않다. 이는 치수 집합의 분류를 위한 전 처리 과정으로써 치수 집합의 인식이 수행되지 못하였기 때문이다.

본 논문에서는 기계 도면을 구성하는 선분들에 대한 분석을 토대로 하여 설명선의 벡터화에 기반한 새로운 치수 집합의 인식 및 분류 방법을 제안한다. 특히, 벡터화 기반의 치수 집합 인식은 인식의 문제를 선 추적이 아니라 산술적 계산의 문제로 전환하는 이점을 가진다. 그리고, 객체의 경우에 Lai의 방법에서와는 달리 완벽한 추출이 가능하게 된다. 이와 함께 본 논문에서는 인식된 치수 집합을 6종류의 치수 양식으로 분류하는 방법을 함께 제안하여 치수 집합의 CAD 데이터화를 수행하였다. 제안하는 알고리즘의 흐름도가 그림 1에 나타난다.

본 논문의 구성은 다음과 같다. II장에서는 도면 영상에서 문자열, 객체, 화살표 및 페루프 심볼의 추출을

나타낸다. III장에서는 객체와 설명선의 벡터화와 훼손된 설명선의 복구 방법을 제시하며, 패턴 매칭과 기하학적 모델링을 이용한 치수 집합의 인식 및 치수 집합의 분류는 IV장에서 설명한다. V장에서는 인식된 치수 집합의 CAD화를 나타내고 VI장에서는 기계 도면을 대상으로 한 실험 결과를 나타내며, 마지막으로 VII장에서 결론을 맺는다.

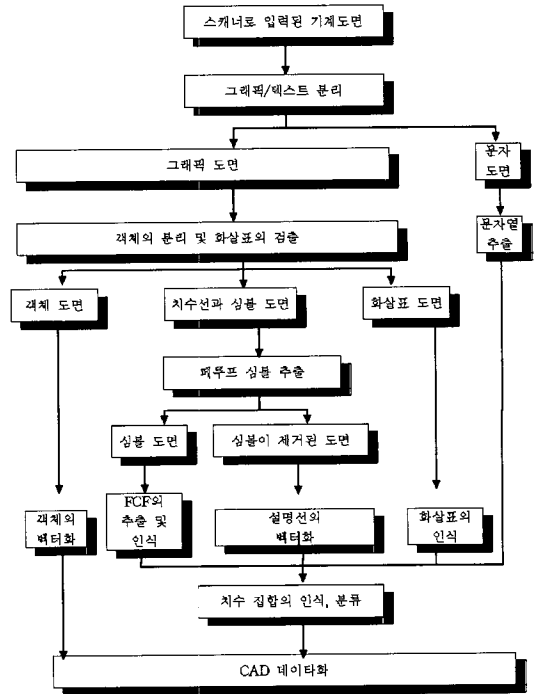


그림 1. 제안한 도면 인식 알고리즘

Fig. 1. The proposed algorithm for drawings recognition.

## II. 문자열, 객체, 화살표 및 페루프 심볼의 추출

도면 인식을 위해서는 도면에 존재하는 문자 성분의 분리가 가장 먼저 선행되어야한다. 본 논문에서는 이러한 문자의 분리 추출을 위하여 Lai가 제안한 방법<sup>[2]</sup>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 점선 등이 문자로 추출되지 않고 그래픽 영역에 남게 된다(그림 2).

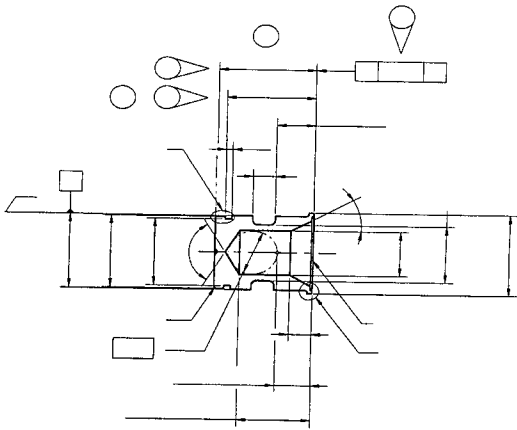


그림 2. 문자가 제거된 그래픽의 예  
Fig. 2. The example of graphic character-extracted.

문자가 제거된 그래픽 도면에는 객체, 설명선이 남게 된다. 객체는 그림 2에서 굵은 실선으로 표시된 부분으로서 실제 3차원 기계 부품을 2차원 평면상에 투영한 것으로 정의되며, 설명선은 이러한 객체를 설명하기 위한 꼬리선, 형태선, 연장선 등의 치수선과 점선 등의 보조적인 선분 및 심볼들로 객체에 비해 상대적으로 가는 선분들이다. 이 중, 그래픽 도면에서 고립 영역으로 존재하는 심볼들은 원 및 원호형의 심볼들로서 문자의 분리 추출과 같은 방법을 이용하여 추출한다.

KS 기계 제도 규격(KS B 0001 5)<sup>[3]</sup>에 따른 선의 굵기 표준에 따르면, 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.

Lai가 제안한 객체 추출 방법에서는 먼저 화살표를 인식하여 이와 연결된 선분을 치수선으로 가정한 다음 선 추적 방법에 의해 치수선을 분리하고, 분리 후 남은 부분을 객체로 추출하였다. 하지만 이러한 방법은 선분 추적에 많은 시간이 소요될 뿐 아니라, 치수선을 이루는 선분과 객체가 겹치는 경우에는 객체의 일부분이 치수선으로 잘못 추출되므로 객체의 정보가 소실되

어 추출된 객체를 정확하게 인식하는데 문제가 발생한다. 따라서, 본 논문에서는 객체와 설명선(치수선, 보조적인 선분, 심볼 등) 간의 선의 굵기 차이를 고려한 간단한 알고리즘으로 객체의 손상을 최소화하여 분리 추출하고자 한다.

객체의 분리 추출을 위한 첫 단계는 opening 연산<sup>[4]</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 설명선 간의 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 화상의 0/1 반전 영상을 만들면 페루프 안은 고립 영역이 된다. 그런 다음 영역 레이블링 알고리즘<sup>[5]</sup>을 이용하여 고립 영역이 심볼 성분의 후보가 되게 한다. 하지만, 후보로 선정된 고립 영역에는 실제 심볼 뿐 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 면적 대 화소의 비를 사용하여 그 비가 1:1에 가까운 심볼을 피쳐 제어 프레임으로 분리 추출한다.

### III. 객체와 설명선의 벡터화

1. 최대 허용 오차법을 이용한 설명선의 선분 벡터화  
선분 벡터의 추출을 위해서는 전처리 과정으로 세션

화가 필요한데 본 논문에서는 Chen-Hsu<sup>[6]</sup>의 세선화 기법을 사용한다. Chen-Hsu의 세선화 알고리즘은 2개의 부루프(sub-loop)를 가지는 병렬 알고리즘으로, 완벽한 8-방향 세선화 결과를 보인다.

세선화가 끝난 설명선을 선분 벡터화 하기 위해 최대 허용 오차법(ME : Maximum Error)<sup>[7]</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 헤아려 본다. 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 체인상의 각 점과 벡터 사이의 거리를 구해 벡터로부터 가장 멀리 떨어져 있는 점을 구한다. 세 번째로 그 거리가 미리 정해진 문턱 값 보다 크면, 벡터는 그 점에서 두 개로 분리된다. 만약 거리가 문턱 값(1.0)보다 작거나 같으면, 벡터는 세선화된 영상을 근사화한 결과라 간주한다. 모든 영상 체인에 대해 위의 과정을 반복함으로써 선분 벡터의 추출을 완료한다. 이상의 반복적인 선근사화 과정을 그림 3에서 보여주고 있다.

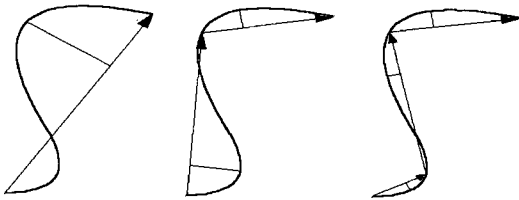


그림 3. 최대 허용 오차법을 이용한 선분 벡터화  
Fig. 3. Vectorization of line using ME.

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 배창석의 방법을

사용하여 원호를 인식한다<sup>[8]</sup>. 이 방법은 선분 벡터간의 연결 형태를 살펴봄으로써 원호를 인식하며 선분 벡터를 대상으로 하므로 수행 속도가 매우 빠르다. 선분 벡터로부터 원호를 추출하기 위한 과정은 원호의 후보가 되는 벡터를 발견하고, 후보 벡터로부터 원호 정보를 구하는 두 단계로 구성되며, 그림 4에서와 같이 원호의 중심점, 반지름, 시작점, 끝점, 방향을 원호 정보로 저장한다.

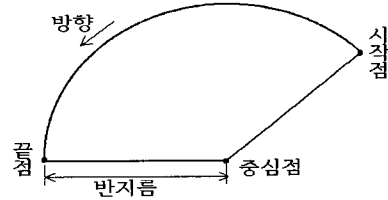


그림 4. 필요한 원호 정보  
Fig. 4. Required arc information.

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>[9]</sup>. 그림 5와 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 문턱 값보다 작으면 두 벡터를 결합하여 준다. 더 이상의 결합될 벡터가 없을 때까지 이러한 과정을 반복 수행함으로써 벡터 결합을 완료한다.

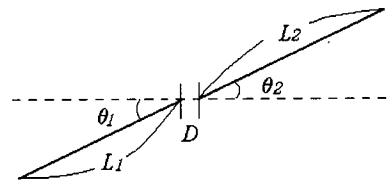


그림 5. 인접한 두 선분 벡터  
Fig. 5. Adjacent two line-vectors.

결합이 끝나면 객체 분리 추출시 훼손된 설명선을 복구하는데 이는 객체가 존재하는 영역을 탐색 영역으로 지정하여 영역 내에 존재하는 설명 선분 벡터간의 거리가 객체의 두께 (10.0)이하이면 두 벡터를 결합하는 것으로 이 과정이 수행되고 나면 설명선의 최종 벡

터화가 이루어진다.

마지막으로, 객체의 벡터화는 향후 연구되어야 할 과제인 객체의 인식을 위한 전처리 과정으로써 설명선과 동일한 방법으로 수행된다.

#### IV. 치수 집합의 인식 및 기능적인 목적에 따른 분류

일반적으로 치수 집합은 치수 집합의 구조에 따라 네종류로 분류되고, 기능적인 목적에 따라 6종류의 치수 양식으로 구분된다. 치수 집합의 구조에 따른 분류는 치수 집합의 대칭성과 규칙성에 의해 다음과 같이 네 종류로 분류된다. 이러한 분류에는 규칙적인 대칭 구조, 규칙적인 비대칭 구조, 비규칙적인 대칭 구조, 그리고 비규칙적인 비대칭 구조가 있다<sup>[11]</sup>. 그림 6에서처럼 규칙적인 비대칭 구조나 비규칙적인 대칭 구조에서는 형태선이 명확하게 그려지지 않는 경우가 있다. 이러한 경우를 가상의 형태선(phantom shape line)이라 한다. 가상의 형태선을 제외하면 모든 치수 집합은 기본적으로 형태선을 가지고 있다. 따라서, 형태선의 인식은 치수 집합을 인식하는 전제가 된다. 본 논문에서는 이러한 형태선의 인식을 기초로 치수 집합을 인식하고, 가상의 형태선을 가진 치수 요소는 치수 집합의 구조적인 특성을 이용하여 치수 집합으로 인식한다. 치수 집합이 인식되고 나면 치수 집합을 수평 치수, 수직 치수, 직경 치수, 반경 치수, 리더 치수, 각 치수 집합등 6종류로 분류한다. 그리고, 형태선의 인식을 위한 선결 과제로서 화살표와 치수선을 인식하여야 하는데, 화살표의 인식에는 패턴 매칭을, 치수선의 인식에는 화살표와 인접한 치수선 사이의 기하학적인 관계를 이용하였다.

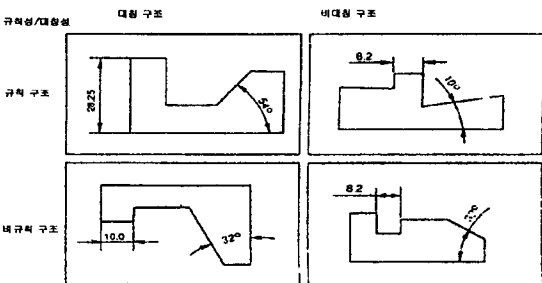


그림 6. 치수 집합의 구조에 따른 분류  
Fig. 6. Classification based on structure of dimension set.

#### 1. 화살표의 인식

일반적으로 표준 규격의 도면에서는 화살표의 가로 세로 비가 2:1로 이루어져 있으며 내부가 채워진 형태로 되어 있다<sup>[3]</sup>. 본 논문에서는 이러한 화살표의 무게 중심과 방향 벡터를 구해 화살표의 인식 및 치수선의 인식에 이용한다. 그림 7과 같이 화살표의 무게 중심은 화살표의 꼬리(arrow back) 부분에 존재하고, 화살표의 머리(arrow tip)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. 제안하는 화살표의 인식 과정은 다음과 같은 순서로 이루어진다.

- ① 가로 세로의 비가 2:1 인 화살표의 기본 패턴을 만든다.
- ② 도면 내에 존재하는 모든 화살표 후보의 무게 중심과 방향 벡터를 구한다.
- ③ 방향 벡터의 각도만큼 회전시킨 화살표의 패턴과 도면 내의 화살표 후보와의 정합도를 고려하여 정합도가 75% 이상이면 화살표로 인식한다.

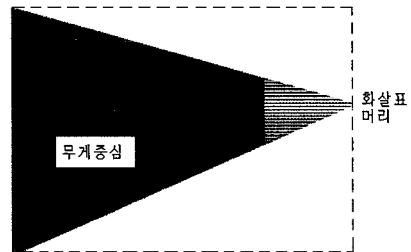


그림 7. 표준 화살표와 화살표의 최소 외접 사각형  
Fig. 7. Standard arrowhead and MBR of arrowhead.

#### 2. 꼬리선, 형태선 및 연장선의 인식

일반적인 기계 도면에서 치수집합(dimension set)을 표현하는 요소는 그림 8과 같이 치수문자(dimensioning text), 화살표(arrowhead), 꼬리선(tail line), 형태선(shape line), 연장선(extension line)으로 구성된다. 그리고 꼬리선, 형태선 및 연장선을 묶어서 치수선으로 정의한다. 이러한 치수선 중에서 꼬리선 및 형태선은 화살표의 방향 벡터와 나란하고 연장선은 화살표의 방향 벡터와 수직인 특성을 가진다. 이 때, 꼬리선은 하나의 화살표를 가지는데 반

해 형태선은 두 개의 화살표를 쌍으로 가지는 구조로 나타난다. 본 논문에서는 이러한 화살표, 꼬리선, 형태선, 그리고 연장선간의 기하학적인 특징을 이용하여 치수선을 인식한다.

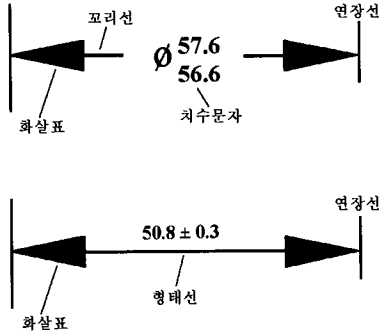


그림 8. 치수 집합의 예  
Fig. 8. Example of dimension set.

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.
- ④ 선택된 벡터중 화살표를 쌍으로 가지는 것을 형태선으로 나머지를 꼬리선으로 인식한다.

태선을 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 거리L와 원호 벡터의 반지름R과의 거리 차의 절대값을 구해 이 값이 정해진 문턱 값 이내에 들어오면 이를 꼬리선 및 형태선의 후보로 결정한다. 그런 다음 직선의 경우와 마찬가지로 후보 벡터와 연결된 화살표의 개수를 이용하여 각각을 꼬리선과 형태선으로 인식한다.

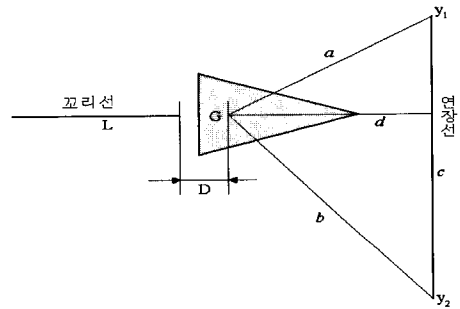


그림 9. 화살표, 꼬리선 및 연장선의 관계에 대한 기하학적인 모델  
Fig. 9. Geometric model for relation of arrowhead, tail and witness.

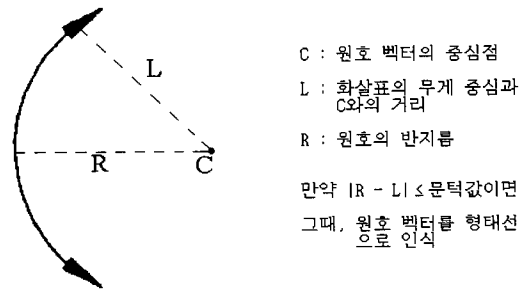


그림 10. 원호 벡터중에서 형태선의 인식  
Fig. 10. Recognition of shape out of arc-vectors.

하지만, 실제 치수 집합에는 화살표의 머리를 마주하는 형태선들이 존재하므로, 본 논문에서는 화살표의 머리와 가까이 있고 화살표와 동일한 방향 벡터를 가지는 선분을 꼬리선과 같이 분류함으로써 형태선의 인식에 이용하였다. 이와 함께 각각의 꼬리선의 끝점과 근거리 내에 있는 직선 벡터들을 꼬리선에 종속하는 벡터로 분류하여 리더 치수 집합의 인식에 이용하였다.

형태선 및 꼬리선이 인식되고 나면, 그림 8과 같이 형태선 사이에 기입할 문자가 많아서 형태선이 두 개의 꼬리선으로 나뉘어진 형태선을 인식한다. 이러한 경우, 문자열을 공유하는 두 꼬리선을 한 개의 형태선으로 인식한다.

다음으로, 설명선의 원호 벡터중에서 꼬리선 및 형

형 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.

3. 형태선 및 꼬리선과 문자열의 결합

일반적으로 치수 집합에 포함되는 문자열은 형태선이나 꼬리선과 평행하게 존재하거나, 꼬리선의 끝점에 연속하여 꼬리선과 동일한 방향으로 나타난다. 치수 집합에서 문자열의 위치 관계를 이용하여 다음과 같이 치수 집합에 속하는 문자열을 인식한다.

- ① 문자열의 추출 과정에서 추출된 문자열 블록의 MBR을 영역 레이블링한 영상을 만든다. 영역 레이블링 과정에서 각각의 MBR의 번호를 문자열 블록의 해당 화소에 기입하고 영상을 저장한다. 이러한 준비 과정이 수행되고 나면 ②로 넘어간다.
- ② 형태선의 중심점(형태선이 원호인 경우는 두 끝점의 중심점)의 수직 이동분선을 추적하여 추적 거리 내에 문자열이 있는지를 확인하고, 문자열이 있으면 그 문자열을 치수 집합에 포함시킨다. 추적 거리 내에 문자열이 없으면 ③을 수행한다.
- ③ 꼬리선의 중심점의 수직 이동분선을 추적하여 추적 거리 내에 문자열이 있는지를 확인하고, 문자열이 있으면 그 문자열을 치수 집합에 포함시킨다. 추적 거리 내에 문자열이 없으면 ④를 수행한다.
- ④ 꼬리선의 끝점에서 꼬리선과 동일한 방향으로 추적하여 추적 거리 내에 문자열이 있으면 그

문자열을 치수 집합에 포함시킨다.

위의 과정을 통해 인식된 문자열 정보를 치수 집합에 저장함으로써 문자열과 치수 집합의 결합은 완성된다.

4. 피쳐 제어 프레임의 인식

페루프 심볼의 분리 추출로 추출된 심볼에는 치수 집합의 한 요소인 피쳐 제어 프레임(FCF: feature control frame)이 있다. 피쳐 제어 프레임은 11과 같이 사각형의 블록안에 다양한 수치값들이 있으며 이 값들은 객체의 특성을 표현하는데 중요한 역할을 한다.

A	Ø10	13.4
---	-----	------

그림 11. 피쳐 제어 프레임  
Fig. 11. Feature control frame.

피쳐 제어 프레임의 인식을 위한 알고리즘의 흐름은 다음과 같다. 페루프 심볼들로부터 사각형의 피쳐 제어 프레임 심볼들의 MBR을 탐색영역으로 하여 피쳐 제어 프레임내에 있는 사각형의 중심 좌표 및 높이와 폭을 구한다. 그리고, 인접한 사각형간의 중심 좌표 및 높이와 폭의 정보를 이용하여 피쳐 제어 프레임의 최외각 좌표와 인접한 사각형간의 경계 좌표를 구함으로써 피쳐 제어 프레임을 인식한다. 마지막으로, 피쳐 제어 프레임내에 문자열이 존재하는지를 조사하고, 그 문자열의 번호를 인식하여 해당하는 문자열이 포함되는 치수 집합에 피쳐 제어 프레임을 포함시킴으로써 인식을 마무리한다.

5. 기능적인 목적에 따른 치수 집합의 분류

기계 도면에서 치수 집합은 6종류의 치수 양식으로 되어있다<sup>[10]</sup>. 인식된 치수 집합을 이러한 6종류의 치수 양식으로 분류하는 것은 치수 집합의 해석 및 CAD화를 위한 전제 조건이다. 본 논문에서는 치수 집합의 형태선과 연장선의 관계를 이용하여 치수 집합을 6종류의 치수 양식으로 분류한다. 이와 함께 가상의 형태선을 가지는 치수 집합(수직 치수, 수평 치수, 각 치수 집합)의 인식 방법도 함께 제시한다.

1) 수평 또는 수직 치수 집합의 분류

일반적으로 수평 또는 수직 치수 집합의 형태선은 0°, 90°, 180° 및 270°로 되어 있다. 본 논문에서는 형태선이 0°, 180°인 것을 수평 치수 집합의 후보로,

형태선이 90°, 270° 인 것을 수직 치수 집합의 후보로 분류한다(그림 12, 13). 그리고, 수평 또는 수직 치수 집합의 치수 표현은 두 개의 연장선이 서로 나란하며 두 연장선과 수직하게 형태선이 존재하므로, 이러한 조건을 만족하는 치수 집합을 수평 또는 수직 치수 양식으로 분류한다. 실제 형태선을 가지지 않고 가상의 형태선을 가지는 치수 집합은 두 개의 연장선이 서로 평행하며 두 개의 화살표가 서로 마주보고 있는 치수 집합을 수직, 또는 수평 치수 집합으로 분류한다. 이러한 치수 집합의 분류가 수행되고 나면 데이터를 CAD 화하기 위해, 연장선의 두 끝점, 형태선의 중심점, 치수 문자순으로 치수 집합을 저장한다.

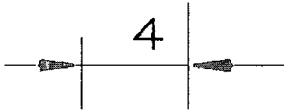


그림 12. 수평 치수 집합  
Fig. 12. Horizontal dimension set.

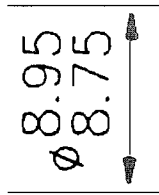


그림 13. 수직 치수 집합  
Fig. 13. Vertical dimension set.

2) 직경 치수 집합의 분류

일반적으로 직경 치수 집합에서 직경을 나타내고자 하는 대상은 원으로 되어있다. 특히, 도면에서 이러한 원은 설명선의 경우에는 일점 쇄선의 원으로 되어 있거나 객체의 경우에는 객체에 포함되는 하나의 원으로 되어있다. 이러한 직경 치수 집합의 인식을 위해서는 객체에 존재하는 원이나, 설명선에서 원의 인식이 전제되어야 한다. 본 논문에서는 연장선이 없는 치수 집합을 후보로 하여, 이러한 후보들 중에서 형태선의 중심점을 검출한 후 이러한 점과 일치하는 중심점을 가지는 원을 설명선 및 객체로부터 찾으므로써 직경 치수 집합으로 인식한다. 이러한 직경 치수 집합은 원의 정보를 이루는 시작점, 치수 문자순서로 저장한다. 본 논문에서는 설명선에서 존재하는 일점 쇄선의 원을 인식하지 않았기 때문에 이러한 치수 집합의 분류는 수

행되지 않을 것이며, 향후 일점 쇄선의 원에 대한 인식이 이루어지면 직경 치수 집합의 인식도 가능할 것으로 여겨진다.

3) 각 치수 집합의 분류

수직 치수, 수평 치수, 직경 치수 집합의 인식이 수행되고 나면, 형태선을 가지는 치수 집합은 각 치수 집합만 남게 된다. 특히 각 치수 집합의 형태선이나 꼬리선은 원호로 되어있으며, 두 개의 연장선은 교차하는 특성을 가진다. 본 논문에서는 이러한 특성을 이용하여 각 치수 집합을 인식한다. 그리고, 가상의 형태선을 가지는 각 치수 집합은 두 화살표의 꼬리선의 중심점이 서로 일치하는 화살표를 후보로 하여 두 연장선의 교차점과 꼬리선의 중심점이 일치하면 각 치수 집합으로 인식한다(그림 14). 이러한 인식이 수행되면 각 치수 집합은 두 연장선의 교차점, 연장선의 두 끝점, 형태선의 중심점, 치수 문자순서로 저장된다.

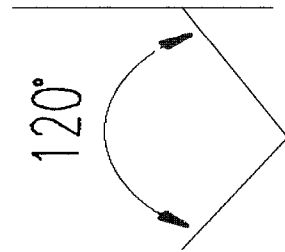


그림 14. 각 치수 집합  
Fig. 14. Angular dimension set.

4) 리더 치수 집합의 분류

형태선을 가지는 치수 집합의 인식이 모두 수행되고 나면, 마지막으로 형태선을 가지지 않는 고립 화살표들과 선분 벡터 및 치수 문자로 구성되는 치수 집합이 존재하는데, 이러한 치수 집합에는 리더 치수 집합과 반경 치수 집합이 있다. 특히, 리더 치수 집합은 화살표의 연장선, 형태선이 없고, 하나의 화살표에 연결되는 꼬리선이 두 개 이상의 벡터로 구성되므로 이러한 특성을 이용하여 리더 치수 집합을 인식한다(그림 15).



그림 15. 리더 치수 집합  
Fig. 15. Leader dimension set.



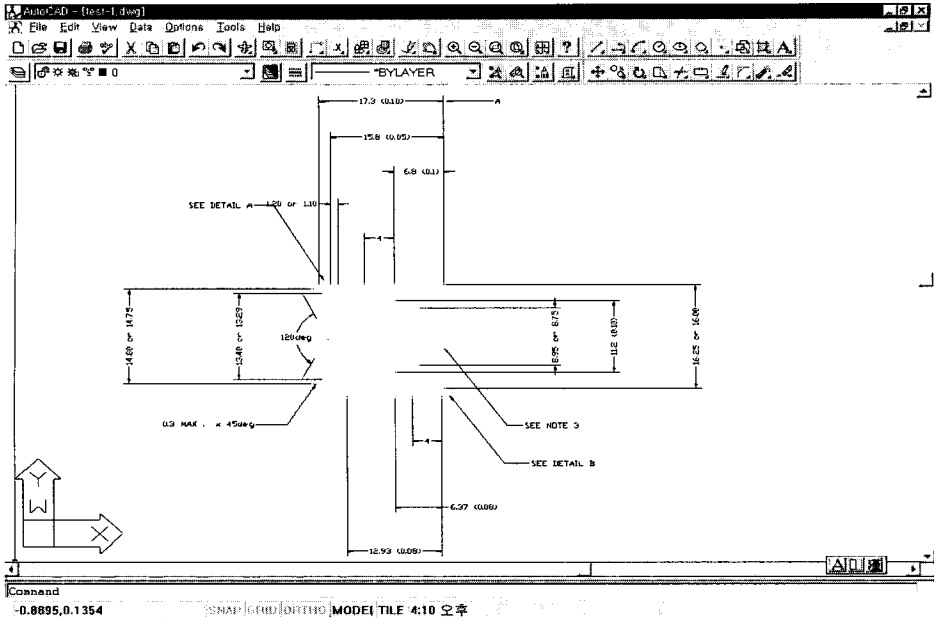


그림 16. 치수 집합의 AutoCAD 데이터  
 Fig.16. AutoCAD-data of dimension sets.

리더 치수 집합의 저장은 화살표의 끝점, 첫 째 벡터의 끝점, 두 째 벡터의 끝점, 치수 문자순서로 이루어진다.

5) 반경 치수 집합의 분류

반경 치수 집합은 형태선을 가지지 않는다는 점에서 리더 치수 집합과 유사하나, 객체나 설명선에 존재하는 원을 대상으로 하고 있다는 점에서 다르다. 반경 치수 집합의 경우도 직경 치수 집합의 인식과 유사하게 반경 치수의 시작점과 일치하는 원의 중심점을 검출함으로써 반경 치수 집합을 인식할 수 있다.

V. 인식된 치수 집합의 CAD 데이터화

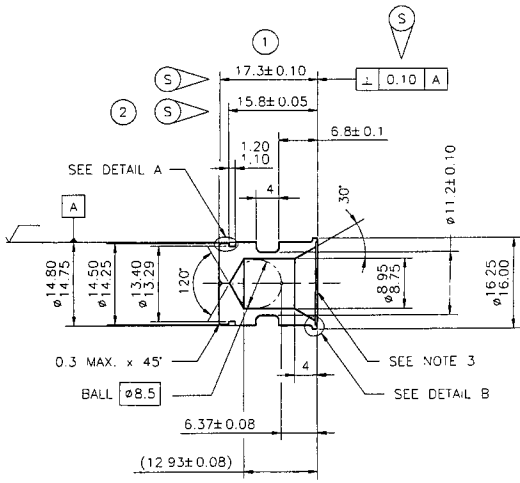
이러한 치수 집합의 분류가 끝나면, 현재 상용화된 CAD 프로그램중 가장 많이 사용되고 있는 AutoCAD 데이터화한다. 본 논문에서는 치수 집합을 AutoLISP를 이용하여 AutoCAD상에 로딩한 뒤 AutoCAD의 기본 파일 포맷인 DWG파일로 전환한다. 이렇게 AutoCAD로 로딩시킴으로써 스캐너 입력시의 오차와, 벡터화 및 인식시 발생할 수 있는 오차를 AutoCAD상에서 수정, 편집이 가능하다. 본 연구에서는 치수 집합을 6종류의 치수 집합으로 분류한 후 각각에 대하여 AutoLISP 언어를 이용하여 Auto-

CAD 데이터화하였다. 이 과정에서 추출된 문자열은 텍스트 파일로 편집을 하여 치수 문자로 사용하였다. 이상의 과정을 통해 입력된 치수 집합의 AutoCAD 데이터가 그림 16에 나타난다.

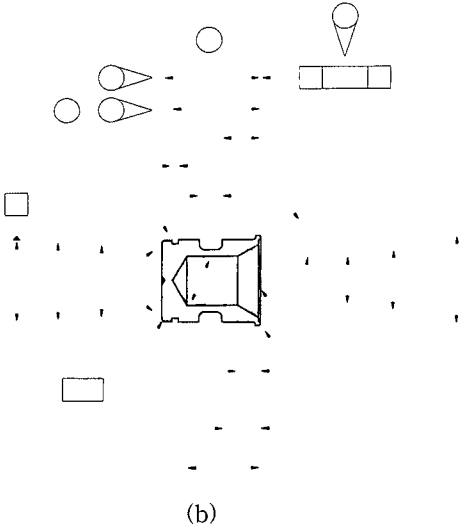
VI. 실험 및 고찰

본 실험의 실험 영상은 KS 기계 제도 규격에 따라 그려진 실제 자동차 부품 도면을 사용하였다. 영상 입력 장치는 인치당 300화소를 가지는 스캐너를 사용하였으며, 그림 17-(a), 그림 18-(a)는 실험에서 사용한 도면 영상으로 각각 2208×2078, 3123×1489의 해상도를 가진다. 도면 처리의 첫 번째 과정인 문자 분리는 고립 문자의 최소 외접 사각형을 이용하여 추출한다. 이때 55 x 55 이하의 고립 영역을 고립 문자로 판단하였다. 비교적 정확하고 높은 추출률을 보였으며 객체, 치수선, 심볼로 구성된 그래픽 영역은 대부분이 연결 선분으로 되어있으므로 이러한 것들을 문자로 추출하는 경우는 없었다(그림 2).

문자가 추출된 도면에서 객체 및 화살표의 분리 추출을 위해 형태학적 연산을 이용하였으며, 페루프 심볼은 영역 레이블링과 구조적 특징을 이용하여 분리하였다. 그림 17-(b), 그림 18-(b)는 각각 객체, 화살



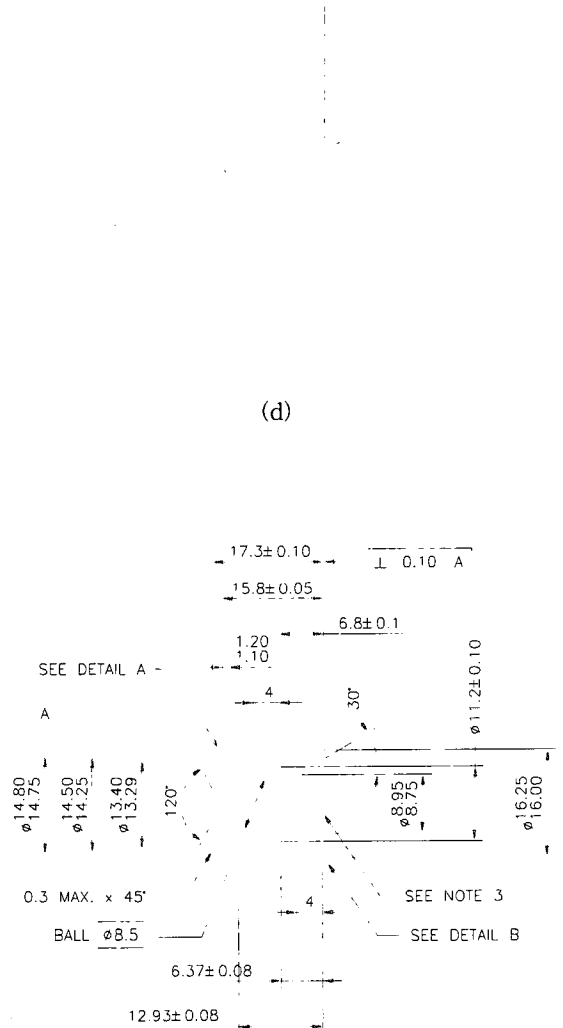
(a)



(b)



(c)



(d)

(e)

그림 17. 실험에 사용한 기계 도면의 예(2208×2078 화소)

(a) 원 도면 (b) 추출된 화살표, 객체 및 심볼 (c) 인식된 형태선 및 꼬리선 (d) 인식된 연장선 (e) 인식된 치수 집합

Fig. 17. An example of mechanical drawing(2208 × 2078 pixel).

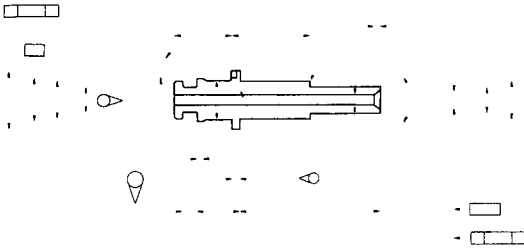
(a) Original drawing. (b) Arrowheads and object. (c) Recognized shape and tail lines. (d) Recognized extension lines. (e) Recognized dimension sets.



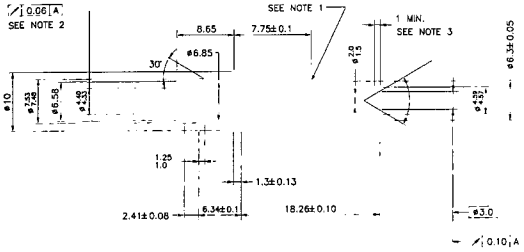
분류가 가능할 것으로 보인다. 향후 연구 방향으로서는 객체의 인식을 통한 치수 집합의 인식률을 높이는데 있을 것이다.

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] 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.
- [3] 송삼홍, "표준 기계 제도," 문운당, 1994
- [4] Ramesh Jain, Rangachar Kasturi, Brian G.Schunck, "Machine Vision", *MIT Press and McGraw-Hill. Inc*, 1995.
- [5] 백영목, 임길택, 김우태, 진성일, "영역 레이블링 방식을 이용한 일반문서의 layout understanding," 제 6회 영상 처리 및 이해에 관한 워크샵, 1994
- [6] 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.
- [7] Urs Ramer, "An Iterative Procedure for the Polygonal Approximation of Plane Curves", *Computer Graphic And Image Processing*, pp. 244-256, 1972.
- [8] 배창석, 민병우, "문자 영역 분리와 기하학적 도면 요소의 인식에 의한 도면 자동 입력," 전자 공학회 논문지 제 31권, 제 6호, pp. 91-103, 1994
- [9] A.Jonk and A. W. M. Smeulders, "An axiomatic approach to clustering line-segments," *Proceedings of the Third ICDAR*, pp. 386-389, 1995.
- [10] 선종화, "Auto Cad Release 12", 연학사, 1993



(b)



(c)

그림 18. 실험에 사용한 기계 도면의 예(3123×1489 화소)  
 (a) 원 도면 (b) 추출된 화살표, 객체 및 심볼 (c) 인식된 치수 집합

Fig. 18. An example of mechanical drawing(3123 ×1489 pixel). (a) Original drawing. (b)Arrowheads, object, symbols.(c) Recognized dimension sets.

VII. 결 론

본 논문에서는 기계 도면의 자동 입력을 위해 도면의 분할 및 벡터화에 기반하여 치수 집합의 인식 및 분류를 수행하였다. 제안된 방법의 이점은 형태학적 연산을 이용함으로써 객체의 분리 추출 알고리즘이 간단해지고, 화살표와 치수선간의 기하학적인 모델을 이용하여 치수 집합을 인식함으로써 알고리즘이 간단하며, 치수 집합을 기능적인 목적에 따라 분류함으로써 AutoCAD 데이터화를 용이하게 한데 있다. 하지만, 치수 집합의 인식 과정에서는 객체에 포함되는 원이나 원호를 대상으로 하는 반경 및 직경 치수 집합의 경우는 객체의 인식이 수행되어야만 정확한 치수 집합의