

P&ID의 파이프라인 인식 향상을 위한 라인 검출 개선에 관한 연구

오상진*, 채명훈*, 이 현*, 이영환**, 정은경**, 이현식*†

*현대엔지니어링, *디지털혁신실, **엔지니어링센터

A Study on the Improved Line Detection Method for Pipeline Recognition of P&ID

Sangjin Oh*, Myeonghoon Chae*, Hyun Lee*, Younghwan Lee**

Eunkyung Jeong**, Hyunsik Lee*†

*Digital Innovation Dept & **Engineering Center, Hyundai Engineering

ABSTRACT : For several decades, productivity in construction industry has been regressed and it is inevitable to improve productivity for major EPC players. One of challenges to achieve this goal is automatically extracting information from imaged drawings. Although computer vision technique has been advanced rapidly, it is still a problem to detect pipe lines in a drawing. Earlier works for line detection have problems that detected line elements be broken into small pieces and accuracy of detection is not enough for engineers. Thus, we adopted Contour and Hough Transform algorithm and reinforced these to improve detection results. First, Contour algorithm is used with Ramer Douglas Peucker algorithm(RDP). Weakness of contour algorithm is that some blank spaces are occasionally found in the middle of lines and RDP covers them around 17%. Second, HEC Hough Transform algorithm, we propose on this paper, is improved version of Hough Transform. It adopted iteration of Hough Transform and merged detected lines by conventional Hough Transform based on Euclidean Distance. As a result, performance of Our proposed method improved by 30% than previous.

Key words : P&ID, Computer vision, Line recognition, Hough transform

1. 서 론

4차 산업 혁명이 가속화됨에 따라 산업 전반에서 디지털 기술을 적용시켜 생산성의 혁신을 가져오려는 시도들이 빈번하게 일어나고 있으며, 이러한 DT(Digital Trans-

formation)는 건설 산업도 피해갈 수 없는 변화의 물결이다.

하지만, 건설 산업은 지난 20년간 생산성 향상이 25% 정도에 그쳐 전 산업 평균인 70%에도 못 미치는 저조한 수준이다[9]. 이런 문제를 극복하기 위해 많은 디지털 기술 접목을 통한 건설 생산성 향상이 시도되고 있으며, DT 기술 도입만으로도 생산성은 14~15%가 향상될 것이라고 전망되고 있다[10]. 이러한 생산성 향상을 위한 다양한 DT 기술 중 AI(Artificial Intelligence)/ML(Machine

† Corresponding Author, E-mail : hslee74@hec.co.kr

Learning)을 활용한 자동 이미지 인식 기술은 그동안 상당수 작업을 수작업에 의존하던 건설업의 생산성 향상에 큰 도움이 될 것으로 기대된다.

P&ID(Piping & Instrument Diagram)는 Engineering Flow Diagram 이라고도 하며 모든 Process 기계 장치류, 배관, Process 제어를 위한 Instrument 정보 및 상호 관계가 표시되고, 이는 후속 공정인 상세설계, 시공/시운전의 기본 자료로 이용된다. P&ID 는 플랜트 기본설계(Basic Engineering) 단계에서 산출되는 주요한 설계도면 중 하나이며, 이에 기재된 정보 및 연관관계를 누락 없이 추출하여 후속공정으로 연계하는 것이 매우 중요하다.

기존 P&ID 자동인식 관련 연구에서 파이프라인을 탐지하는 방법은 Hough Transform[1]과 같은 라인 탐지 기술을 주로 활용하였다[5,8]. 하지만 설계자의 관점에서 기존 Hough Transform의 라인 탐지 결과를 실무에 적용하기에는 탐지 성능의 부족이나 탐지 결과의 불안정성 등으로 인한 한계가 있었다. 특히, 프로세스의 흐름을 나타내야 하는 P&ID의 특성상 도면 요소 간의 연결 관계를 인식하는 것이 중요한데, 기존 Hough Transform의 인식 결과는 직선을 결점 없이 인식하는데 부족한 성능을 보여줌으로 활용하기 어렵다.

본 논문에서는 기존 Hough Transform의 문제를 보완하고자 두 가지 새로운 파이프라인 검출 방법을 제안한다. 첫 번째 제안하는 CC&R(Combination Contour & RDP)은 컴퓨터 비전 기술인 Contour와 RDP(Ramer-Douglas-Peucker)를 활용해 파이프라인을 직선이 아닌 다각형 관점으로 검출하는 방식이다. 두 번째로 제안하는 HEC Hough Transform은 기존의 Hough Transform이 갖는 문제를 보완하여 개선한 라인 검출 방법이다. 본 논문에서 제안하는 두 가지 라인 검출 방법을 평가하기 위해 실제 P&ID 스캔본 이미지 10장을 대상으로 기존 Hough Transform과 비교 실험을 진행하여 검증하였다.

2. 기반 기술 소개

2.1. Hough Transform

Hough Transform은 이미지에서 라인을 검출하는 대표

적인 컴퓨터 영상 처리 기법이며, 자율 주행 분야의 차선 인식, 건축 분야의 건물 구조 감지 등에 활용된다[2].

Hough Transform은 이미지에 존재하는 점의 집합 중 기울기 즉, 벡터 값이 동일한 점의 집합을 선으로 간주하여 라인으로 검출하는 원리이며, 더 나아가 원을 인식하는 Circle Hough Transform, 점집합의 확률적 분포를 고려해 특정 부분만 라인으로 인식하는 Probabilistic Hough Transform과 같은 기술로 발전했다.

Hough Transform은 OpenCV 라이브러리 형태로 제공되고 있으며 이를 활용할 때는 다양한 인자가 필요하다. 임계값은 다양한 인자 중 하나이며 이는 라인을 검출할 때 중요한 역할을 한다. 임계값의 크기가 작을수록 많은 라인을 검출하지만 정확도가 낮아지며 반대로 임계값의 크기가 커질수록 검출되는 라인의 수는 줄어들지만 정확도가 향상한다. 따라서 적절한 임계값의 설정이 필요하다. 흑백 사진에서는 이와 같은 임계값을 자동으로 설정하기 위해 일반적으로 THRESH_BINARY와 THRESH_OTSU를 활용한다. THRESH_BINARY는 흑백 이미지를 이진화하는 역할을 하며, THRESH_OTSU 이에 대하여 적절한 임계값을 자동으로 계산하는 역할을 한다.

2.2 Contour

Contour는 이미지에서 객체 검출과 형태 분석에 사용되는 영상분석 기법이다. 이는 이미지 내 픽셀들의 속성 값을 얻고, 이 픽셀 속성 값이 동일한 영역의 윤곽을 찾아내는 원리이다. Contour는 주로 등고선, Bounding box 등 도형의 윤곽선을 탐지할 때 활용되며 OpenCV에서 라이브러리 형태로 제공한다[3].

2.3 RDP(Ramer-Douglas-Peucker Algorithm)

연속되는 정점을 연결하여 생성된 일련의 선 세그먼트를 폴리 라인이라고 하며, 이를 구성하는 점이 많을수록 구체적이지만 복잡한 형태를 보인다. RDP는 다수의 점으로 구성된 폴리 라인을 근사화하기 위해 불필요한 점을 제거하고 단순화하는 알고리즘이다. 이를 통해 이미지에서 추출한 경계면의 노이즈를 줄이고 보다 직관적인 형태로 변형한다[4].

3. 시스템 설계

본 논문에서 제안하는 두 가지 시스템은 Fig. 1과 같으며 이를 수행하기에 앞서 공통적으로 라인 탐지를 하고자 하는 대상 이미지의 전처리 과정이 필요하다. 일반적으로 전처리 과정을 거치기 전의 이미지 데이터는 여과 없이 수집된 데이터이기 때문에 이미지 내에 노이즈가 존재한다. 이와 같은 노이즈는 추후 라인을 탐지하는 과정에서 잘못된 결과를 유도하여 성능 저하의 원인이 된다. 이와 같이 이미지 전처리는 필수적이고 이를 수행한 후 라인을 탐지하기 위해 본 논문에서 제안하는 두 가지 라인 검출 시스템을 각각 수행한다.

3.1 이미지 전처리

수집된 P&ID 도면은 대부분 스캔 이미지 파일이기 때문에 무수히 많은 노이즈가 존재한다. 따라서 노이즈를 필터링함으로써 인식률을 향상시키는 전처리 과정이 필수적이다. 이미지 전처리 과정에서는 입력 이미지를 Gray scale로 변환하였으며, 컴퓨터 영상 처리 분야에서 주로 사용하는 Threshold, Bitwise_not, Dilation 등 다양한 영상 처리 기법을 활용하였다.

Threshold와 Bitwise_not는 이미지 내에 존재하는 노이즈를 필터링하기 위해 활용한다. 이 기법들은 유의미한 정보를 갖고 있는 픽셀 주변의 잔상 혹은 스캔하는 과정에서 발생한 노이즈 등 흐릿한 픽셀을 제거한다.

Dilation은 CC&R에서 하나의 라인의 두께를 확장하여 선이 아닌 도형으로 보기 위해 필요한 영상 처리 기법이다.

3.2 CC&R(Combination of Contour & RDP)

본 논문에서 제안하는 첫 번째 방법은 Contour 기법과 RDP 알고리즘을 혼합한 CC&R이며, 이 시스템의 개요는 Fig. 1(a)과 같다.

먼저, CC&R을 수행하기 위해서는 대상 이미지의 전처리가 필요하다. 이 과정에서 노이즈를 제거하기 위한 Threshold, Bitwise_not 기법과 선을 도형으로 볼 수 있게 선의 두께를 확장 시키는 Dilation 기법이 적용되었다.

Line Detection 과정은 전처리된 이미지에 Contour와 RDP를 혼합 적용하여 라인을 탐지한다.

기존 Hough Transform이 검출한 라인은 하나의 직선에서 부분적으로 공백이 발생하여 다수의 직선으로 탐지되는 문제가 있었다. 이를 해결하기 위해 CC&R은 라인 성분을 선이 아닌 도형으로 보고, 선의 윤곽선을 탐지하기 위해 Contour 기법을 적용하였으며, 라인의 누락된 부분이 발생하는 현상을 최소화했다. 또한 RDP를 활용하여 라인을 근사화하였으며, 이는 Contour만으로 이미지 전처리를 할 경우 노이즈가 남기 때문에 불필요한 라인 탐지를 최소화하기 위한 과정이다.

3.3. HEC Hough Transform

본 논문에서 제안하는 두 번째 방법은 기존의 Hough Transform을 강화시킨 HEC Hough Transform이며, 이 시스템의 개요는 Fig. 1(b)과 같다.

3.3.1. Image Preprocessing

먼저 CC&R과 마찬가지로 Threshold와 Bitwise_not 기법을 통해 이미지 전처리 과정을 수행하여 노이즈를 제거한다. HEC Hough Transform은 선을 양 끝 두 점으로

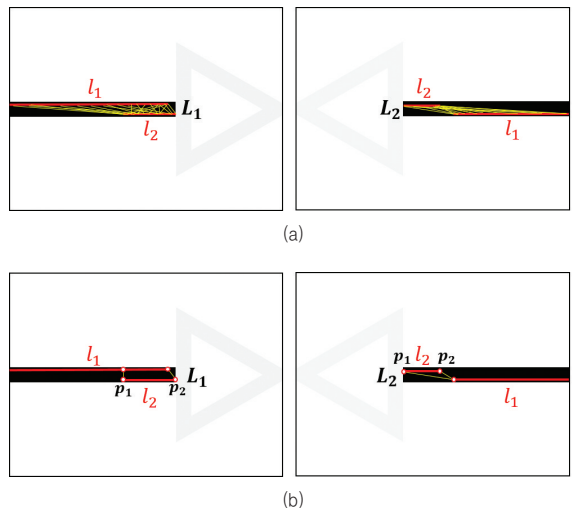


Fig. 2 (a) Calculate all distances between randomly chosen two points to find the shortest and (b) calculate shortest distance from each end points to another line.

구성된 형태로 인식하고자 하므로 CC&R에서 사용된 Dilation 기법은 적용하지 않는다.

3.3.2. Hough Transform & Classification of Line by Angle

이미지 전처리를 통해 얻은 이미지는 기존의 Hough Transform을 활용하여 라인을 검출하고, 문제점을 가진 상태로 결과가 반환된다. 이와 같은 문제점을 보완하기 위해 Hough Transform을 반복하는 Iteration Hough Transform 과정을 수행했다. 하지만 이와 같은 과정을 수행하기 전에 수평선과 수직선, 대각선이 잘못 병합되지 않도록 각도에 따라 분류하는 과정이 필요하다. 각도가 다른 라인과 함께 Line Merge Algorithm을 수행하면 병합이 잘못될 위험이 존재하기 때문에 해당 과정이 필수적이다.

3.3.3. Iteration Hough Transform

Line Merge Algorithm은 동일한 라인에서 조각난 라인들을 병합하는 과정이다. 기존의 Hough Transform을 수행하면 동일한 라인 내에 무수히 많은 조각난 라인들이 발생한다. 예를 들면 Fig. 2와 같이 하나의 라인(L_1, L_2 ; 검은색)에서 다수의 조각난 라인(l_1, l_2 ; 빨간색)이 탐지된다. 조각난 라인을 본래 하나의 라인으로 병합하기 위해서

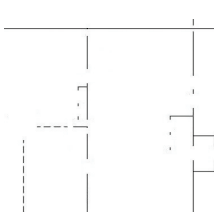
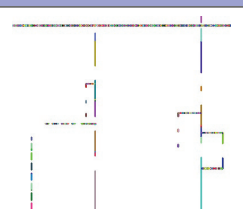
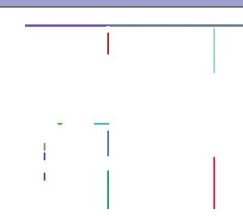
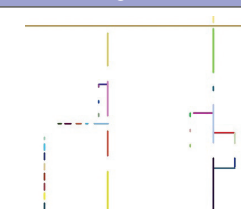
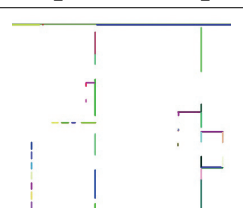
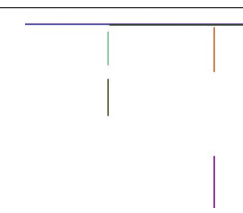

Euclidean distance를 활용하여 조각난 두 라인 간의 최소 거리를 계산하고 일정 기준 거리 이하인 경우 병합 처리 하였다. 최소 거리를 계산하는 방법은 Fig. 2(a) 와 Fig. 2(b) 두 가지 방법이 있다. Fig. 2(a)의 경우는 조각난 라인(빨간색)에 존재하는 무수히 많은 점들 간의 거리(노란색)를 구해야하기 때문에 자원과 시간 면에서 비효율적이다. 이를 해결하기 위해 본 연구에서는 Fig. 2(b)과 같은 방법을 제안한다. 이는 선0과 선0의 양끝 점0과의 최소 거리(노란색)를 측정한다.

이때 과는 더 긴 라인을 으로 선정한다. Fig. 2(b)는 Fig. 2(a)과 다르게 최소 거리를 구하기 위한 후보 라인이 적기 때문에 자원과 시간 면에서 보다 효율적이다.

Euclidean distance와 Fig. 2(b)를 활용하여 동일 직선으로 확인된 조각난 라인들을 병합하여 Merged Line을 반환한다. 이후 Line Filter를 통해 병합 혹은 탐지되지 않은 Residual Line을 반환한다.

반환 결과, Residual Line이 지속적으로 증가하면 이와 같은 과정을 반복 수행하고 반대로 Residual Line의 증가가 멈추거나 더 이상 남아 있지 않게 되면 Iteration Hough Transform을 종료함으로써 최종적인 Detected Line Image를 산출한다.

Table 1. Line detection results for patch in P&ID

Original Image	Hough Transform(Threshold)		Proposed Method
	Threshold = 0	Threshold = 50	HEC Hough Transform
			
	<p>Threshold = THRESH_BINARY+THRESH_OTSU</p> 	<p>Threshold = 100</p> 	<p>CC&R</p> 

4. 실험 결과 및 성능 평가

4.1. 실험 환경 및 데이터 셋

본 연구는 Intel® Core™ i7-6700 CPU @ 3.40 GHz, 8 GB RAM and Windows 10 환경에서 python 3.6, OpenCV 4.2.0를 활용하여 실험을 수행했다.

데이터 셋은 현대엔지니어링에서 수행한 프로젝트 도면 중 일부를 선별하여 10장의 실제 P&ID 스캔본 이미지 도면을 대상으로 실험을 수행하였다. 또한 P&ID에서 파이프 라인만을 검출하기 위해 다양한 Symbol과 Text를 삭제한 Skeleton 이미지를 생성하여 실험을 진행했다.

4.2 실험 결과 및 성능 분석

실험을 위해 본 논문에서 제안하는 두 가지 라인 검출 시스템 CC&R과 HEC Hough Transform을 각각 수행하였다. Table 1은 기존 Hough Transform과 본 논문에서 제안하는 두 가지 방법으로 라인을 탐지하였을 때 결과의 일부 도면 이미지를 보여준다. 하나의 라인이 얼마나 많은 조각으로 나뉘어졌는지를 보기 위해 탐지된 라인마다 서로 다른 색으로 나타내었다.

기존 Hough Transform은 Threshold의 값이 작을수록 더 많은 라인 탐지가 가능하지만 하나의 라인에서 무수히 많은 라인으로 탐지하는 것을 볼 수 있다. 반대로 Threshold 값이 커지게 된다면 조각난 라인의 개수는 줄어들지만 전체적으로 탐지된 라인의 개수가 현저히 줄어든다.

본 논문에서 제안하는 첫 번째 방법인 CC&R은 빠짐없이 모든 라인을 탐지하였다. 하지만 Table 1에서 CC&R 결과 이미지를 분석하면 직선들이 분기된 경우에도 동일한 색으로 표현되고 있다. 이는 각 직선 별로 나뉘어져 인식해야 하는 라인들이 하나의 도형으로 탐지된 것으로 분석된다. 이는 각 요소의 연결 관계를 정밀하게 인지해야 하는 P&ID 도면 인식에 불리한 요소이다. 따라서 CC&R은 정밀한 라인 탐지가 어렵다고 판단된다.

두 번째 제안하는 방법인 HEC Hough Transform은 보다 향상된 성능으로 라인을 검출하였다. 기존 Hough Transform을 통해 탐지된 라인을 병합하는 과정을 수행

Table 2. Experimental results

(%)

P&ID #	Precision	Recall	F1-score
Hough Transform			
P&ID 001	33.36	72.90	45.78
P&ID 002	28.65	53.33	37.28
P&ID 003	33.15	57.29	42.00
P&ID 004	33.06	66.48	44.16
P&ID 005	30.52	62.24	40.96
P&ID 006	31.83	67.38	43.23
P&ID 007	32.84	64.02	43.42
P&ID 008	33.15	72.81	45.55
P&ID 009	32.93	62.67	43.18
P&ID 010	29.31	59.68	39.32
Avg	31.88	63.88	42.49
CC&R			
P&ID 001	48.85	100	65.64
P&ID 002	52.58	100	68.92
P&ID 003	49.02	100	65.79
P&ID 004	49.05	100	65.82
P&ID 005	51.00	100	67.55
P&ID 006	50.63	99.95	67.22
P&ID 007	49.40	100	66.13
P&ID 008	49.13	100	65.88
P&ID 009	48.56	100	65.37
P&ID 010	52.89	100	69.18
Avg	50.11	99.99	66.75
HEC Hough Transform			
P&ID 001	98.81	98.49	98.65
P&ID 002	99.19	99.07	99.13
P&ID 003	98.18	93.99	96.04
P&ID 004	99.56	98.66	99.11
P&ID 005	92.90	93.68	93.29
P&ID 006	94.59	93.17	93.87
P&ID 007	93.92	88.22	90.98
P&ID 008	100	98.47	99.23
P&ID 009	100	95.53	97.71
P&ID 010	99.60	81.90	89.89
Avg	97.68	94.12	95.79

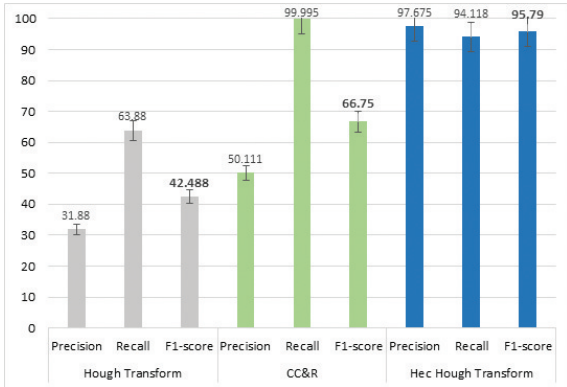


Fig. 3 Graph of Average Experimental Results

하였기 때문에 하나의 라인이 다수의 라인으로 탐지되는 문제를 해결하였으며, CC&R과 같이 교차하는 라인을 하나의 도형으로 처리하는 문제도 해결하였다.

이와 같은 결과를 수치적 해석을 위해 Precision, Recall, F1-score를 활용하였다. 먼저 탐지된 라인의 좌표를 OpenCV를 통해 새로운 라인 이미지로 생성 후 원본 이미지와 픽셀 단위로 비교하여 차이가 나는 픽셀의 수만 큼 오차를 측정하였다. 이에 대한 결과는 Table 2와 Fig. 3과 같다. 본 논문에서 제안하는 두 가지 라인 탐지 기법이 전체적으로 기존 Hough Transform 보다 향상된 성능을 보였다.

또한, 이 두 가지 방법의 결과를 비교하면 CC&R이 HEC Hough Transform에 비해 평균 Recall의 값이 약 5.8%p 높았지만 HEC Hough Transform은 CC&R에 비해 평균 Precision이 47.5%p 높았으며, 최종적으로 HEC Hough Transform의 평균 F1-score가 CC&R에 보다 약 29%p 뛰어난 성능을 보였다.

HEC Hough Transform의 높은 Precision은 오탐이 적다는 것을 시사한다. 하지만 Recall 값의 비교적 낮은 수치는 라인의 미탐이 발생할 가능성이 있다고 볼 수 있다. 이에 대한 원인은 탐지된 라인 중에 두께와 길이가 동일한 아주 미세한 라인은 점과 같기 때문에 탐지하기가 어렵다고 분석된다. 이를 해결하기 위해서 Fig. 1(b)에서 Hough Transform 과정을 수행할 때 탐지 임계값을 적절히 조정하여 미세한 점과 같은 라인을 탐지해야 하며, 상대적으로 높은 Recall 값을 보인 CC&R의 적절한 조화가 필요하다.

5. 결론

일반적으로 기존 P&ID 자동 인식 연구[5,6,7,8]에서 파이프라인을 기존의 Hough Transform으로 검출하였다. 하지만 P&ID에서 파이프라인을 검출하는 경우 하나의 라인을 다수의 조각난 라인으로 검출하는 문제로 인해 응용 단계에서 각 파이프라인들과의 연결정보를 도출하는 데에 어려움이 있었다.

이를 해결하고자 본 연구에서는 향상된 P&ID의 파이프라인 인식을 위해 두 가지 라인 검출 시스템을 제안하였다. HEC Hough Transform은 동일한 라인에서 검출된 미세한 라인들을 반복 병합하여 기존 Hough Transform이 갖는 문제를 보완하였다. 또한, CC&R은 라인을 선이 아닌 도형으로 뭉으로써 선의 윤곽선을 탐지하는 Contour와 이를 근사화하는 RDP를 혼합하여 파이프라인을 검출하는 시스템을 제안하였다.

본 연구에서 제안하는 HEC Hough Transform과 CC&R 두 개의 시스템의 성능은 실험을 통해 각각 95.79%, 66.75%의 F1-score를 보였다. 이는 기존 Hough Transform보다 약 53%p, 24%p 이상 향상된 성능이다.

특히, HEC Hough Transform은 기존 Hough Transform뿐만 아니라 CC&R보다도 뛰어난 성능을 보였다. CC&R은 기존 Hough Transform에 비해 성능을 다소 높였지만 불필요한 정보들을 비교적 많이 가지고 있었고, 교차되어 있는 라인을 하나의 도형으로 인식함으로써 향후 파이프라인과 심볼 간의 연결정보를 도출하는데 불리했다.

반면 HEC Hough Transform은 높은 성능을 보였을 뿐만 아니라 교차된 라인도 조각난 라인으로 인식하였기 때문에 향후 파이프라인과 심볼 간의 연결정보를 도출하는데 큰 기여를 할 것으로 기대된다. 하지만 본 연구의 실험에 따르면 HEC Hough Transform의 Recall은 기존의 Hough Transform의 Recall 보다 향상됨을 보였지만, CC&R의 Recall에 비해 낮은 성능을 보인다. 이를 개선하기 위해서 향후 HEC Hough Transform과 CC&R의 병합을 통해 HEC Hough Transform의 한계점 해결 및 성능 향상을 시도하고자 한다.

Reference

- [1] Hough, P. V. C., 1962, "Method and means for recognizing complex patterns." U.S. Patent No. 3,069,654.
- [2] Hassanein, A. S., Mohammad, S., Sameer, M., and Ragab, M. E., 2015, "A survey on Hough transform, theory, techniques and applications," arXiv preprint arXiv:1502.02160.
- [3] Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J., 2010, "Contour detection and hierarchical image segmentation," IEEE transactions on pattern analysis and machine intelligence, Vol. 33, No. 5, pp. 898~916.
- [4] Wu, S. T., and Marquez, M. R. G., 2003, "A non-self-intersection Douglas-Peucker algorithm," In 16th Brazilian symposium on computer graphics and Image Processing (SIBGRAPI 2003), IEEE, pp. 60~66.
- [5] Rahul, R., Paliwal, S., Sharma, M., and Vig, L., 2019, "Automatic Information Extraction from Piping and Instrumentation Diagrams," arXiv preprint arXiv:1901.11383.
- [6] Elyan, E., Garcia, C. M., and Jayne, C., 2018, "Symbols classification in engineering drawings," In 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, pp. 1~8.
- [7] Moreno-Garcia, C. F., 2018, Digital interpretation of sensor-equipment diagrams. CEUR Workshop Proceedings, pp. 1~7.
- [8] Rica, E., Moreno-Garcia, C. F., Alvarez, S., and Serratosa, F., 2020, "Reducing human effort in engineering drawing validation," Computers in Industry, 117, 103198, pp. 1~7.
- [9] The global construction industry is now at war on 'increasing productivity' through BIM and automation, <http://www.koscaj.com/news/articleView.html?idxno=108611>
- [10] Digital Transformation of Construction Industry, http://m.cnews.co.kr/m_home/view.jsp?idxno=202001021656224490094 