

논문 2007-44SC-6-2

선분 세그먼트 기반 Randomized Hough Transform

(Line Segment Based Randomized Hough Transform)

한 광 수*, 한 영 준**, 한 현 수**

(Kwangsoo Hahn, Youngjoon Han, and Hernsoo Hahn)

요 약

기존 Hough transform을 이용한 타원 검출의 수행 속도와 개수의 추정을 개선하기 위해 본 논문에서는 선분 세그먼트 기반 Randomized Hough Transform (RHT)을 제안한다. 제안하는 방법은 에지 영상을 선분 세그먼트 단위로 분할한 후 임의의 선분 세그먼트 쌍을 RHT를 이용해서 타원을 추정하여 병합여부를 판단한다. 이와 같이 선분 세그먼트 단위로 RHT를 적용하면 적은 반복수행으로 타원을 추정할 수 있으며 복잡한 에지 영상에서도 보다 정확한 타원의 개수를 추정할 수 있다. 제안된 방법의 효율성은 계산속도 및 타원검출의 정확도로 평가하였으며 다양한 입력영상에 대한 실험을 통해 입증하였다.

Abstract

This paper proposes a new efficient method to detect ellipses using a segment merging based Randomized Hough Transform. The key idea of the proposed method is to separate single line segments from an edge image, to estimate ellipses from any pair of the single line segments using Randomized Hough Transform (RHT), and to merge the ellipses. This algorithm is able to accurately estimate the number of ellipses and largely improves the computational time by reducing iterations.

Keywords : Edge Segment, Randomized Hough Transform, Ellipse Detection, Occluded Ellipse.

I. 서 론

에지영상에서 타원을 검출하는 일은 영상에 포함된 물체의 특징 정보를 추출하거나 물체의 형태를 인식하는 목적에 사용된다. 실제 산업현장에서 사용되는 볼트, 링, 자동차 타이어와 같이 많은 산업용 부품들뿐만 아니라 인간의 얼굴이나 적혈구 같이 인체에서도 타원으로 표현할 수 있는 요소들이 많기 때문에 타원검출 기법은 컴퓨터비전에서 매우 중요한 부분을 차지하고 있다. 따라서 타원 검출을 위한 연구는 오래된 역사를 가지고 있으면서도 지금까지 많은 연구들이 진행되고 있다.

타원검출 알고리즘들은 대부분 1962년 Hough^[1]가 제안한 Hough Transform (HT)에 기초하고 있다.^[2~6] 그

중 가장 최적화된 HT의 하나인 Randomized Hough Transform (RHT)은 Lei Xu^[5]에 의해 제안되었으며, RHT는 타원 검출을 위해 두 과정으로 나누어 수행한다. 먼저 임의의 세 점의 중점과 접선의 중점으로 타원의 중심점을 추정하고 간소화된 타원 방정식으로부터 나머지 세 개의 변수를 결정하는 방법을 사용하고 있다. 기존의 HT에서 타원 검출을 위해 사용되는 5차원 공간을 타원의 원점에 해당하는 2차원 공간과 나머지 변수들에 해당하는 3차원 공간으로 나눔으로써 저장 공간을 많이 감소시켰다. 그리고 검출 하고자 하는 대상 화소를 선택할 때 임의의 화소를 선택하는 방법을 사용하여 계산 시간을 상당히 줄일 수 있었다. 하지만 영상에서 타원을 결정하기 위해 세 개의 에지 픽셀을 임의로 선택하기 때문에 타원을 잘못 찾을 수 있고, 전체 화소 수에 비례하는 수행시간을 가지며 타원의 개수를 정확하게 추정하지 못하는 문제점이 있다.

Yonghong Xie^[7]은 타원 장축의 이점을 이용하여 타원 변수를 빠르고 정확하게 검출하는 방법을 제안했다.

* 학생회원, ** 정회원 숭실대학교

(Department of Electrical Engineering, Soongsil University)

접수일자: 2007년3월9일, 수정완료일: 2007년10월19일

타원의 경계에 해당하는 임의의 두 점을 선택하여 그 점들을 타원의 장축의 양 끝점이라고 가정한다. 그러면 이 두 점으로 타원의 중심점과 주축의 길이와 타원의 방향을 계산할 수 있다. 이로서 5개의 타원 변수 중 4개를 구했다. 그리고 또 다른 세 번째 점에 타원 기하학을 적용하여 타원의 단축의 길이를 구할 수 있다. 이 단축의 길이를 1 차원 누적 배열에 저장하고 이 누적 배열로부터 타원을 결정할 수 있다. 이 알고리즘은 타원의 점선이나 곡률의 계산이나 복잡한 수학적 연산이 필요하지 않다. 하지만 이 방법은 타원의 장축을 모든 에지 점들의 쌍으로 계산해야 하기 때문에 HT 방법과 비교해 봤을 때 수행시간이 다소 오래 걸린다.

타원검출의 속도를 높이는 또 다른 방법으로 Elmowafy^[8]는 Fast Graphical Ellipse Detection (FGED) 알고리즘을 사용하였다. 먼저 타원의 중심점을 추정하기 위해서 RHT와 비슷하게 임의의 세 개의 에지 점을 선택하여 Graphical Validity Method (GVM)를 적용한다. GVM은 임의로 선택한 점들의 중점이 추정된 타원의 중심점과 점선들의 교점 사이에 존재하는 지를 체크한다. 다음으로 GVM을 만족하는 점들을 Mid Central Point Voting (MCPV)을 적용하여 타원의 중심점을 찾고 RHT와 같은 방법으로 나머지 변수들을 찾는다. 마지막으로 타원 경계에 속하는 점들과 추정된 타원의 원주의 길이 비를 계산하여 타원의 합당성을 판단하고 타원을 결정하게 된다. Uprwrite method^[9]의 경우는 타원 경계의 임의의 픽셀을 중심점으로 반지름이 r인 원형 이웃의 국부 모델을 계산하기 위해 Spot 알고리즘을 사용한다. 만약 이 반지름 r이 충분히 작다면 국부 모델의 값들은 직선으로 근사될 수 있다. 이 국부 모델들 중 같은 기하학적 특징을 갖는 국부 모델 집합을 만든다. 마지막으로 각각의 기하학적 특징을 Zernike moments를 이용해서 타원으로 특성화 한다.

이와 같이 다양한 타원검출 알고리즘들이 HT에 기반하고 있으며 근본적으로 HT가 정합을 통해 타원을 검출하는 기법이기에 때문에, 이로 인해 발생하는 속도와 정확도 문제를 해결하기 위한 방법들을 제안하고 있다. 본 논문에서도 입력 영상의 에지를 단일 세그먼트로 분리하고 같은 타원에 속하는 선분 세그먼트 간의 병합하는 방법을 사용하여 타원검출의 속도와 정확도를 향상시키는 방법을 제안한다. 세션화된 에지영상에서 타원 변수들의 결정은 크게 두 단계로 나눠 진행된다. 첫 번째 과정은 하나의 타원에만 귀속되고 한쪽 방향성만 갖는 연속된 화소들의 집합인 선분 세그먼트로 분리하는

과정을 수행한다. 변형된 라벨링 (Labeling) 기법을 이용하여 에지 분기점을 분리한다. 다른 타원에 속하지만 분기점이 존재하지 않아 아직 분리되지 않은 세그먼트들은 코너 패턴을 이용하여 선분 세그먼트를 완성하게 된다. 그리고 선분 세그먼트들 사이에 병합 가능성도 (Merging Probability Map, MPM)를 구성한다. 분리된 선분 세그먼트들을 하나의 타원에 속하는 선분 세그먼트끼리 병합하기 위하여 RHT로 타원 변수를 추정하고 추정된 타원 변수를 타원 결정 부등식으로 선분 세그먼트 사이의 병합 여부를 판단한다. 위 과정을 수행함으로써 얻어진 병합된 선분 세그먼트의 개수를 통해 입력 영상 내의 전체 타원의 개수를 정확하게 추정할 수 있으므로 기존 RHT의 문제점을 획기적으로 개선시킬 수 있다. 또한 전체 에지 화소들로 RHT를 수행하지 않고 분리된 선분 세그먼트 단위로 타원 변수를 결정하기 때문에 전체 수행시간을 크게 줄일 수 있다.

본 논문의 구성은 II장에서는 전체 시스템에 대한 설명과 전 처리 과정에 대해 설명한다. III장에서는 선분 세그먼트 검출을, IV장은 타원 결정을 위한 선분 세그먼트 병합 기법을 다룬다. 마지막으로 V장과 VI장에서 실험 및 결과와 결론 및 향후 연구방향을 제시한다.

II. 시스템 구성 및 전 처리

1. 전체 시스템 구성

본 논문에서 제안하는 알고리즘의 구성도는 그림 1과 같다. 우선 입력 영상 중에서 타원에 해당하는 부분을 추정하기 위해 입력 영상의 에지를 검출한다. 이 에지를 하나의 타원에만 속하는 단일 세그먼트로 분할하기 위해서 일차적으로 변형된 라벨링 기법을 이용하여 에지 분기점을 중심으로 분할하고 아직 분할되지 않은 세그먼트들을 코너패턴 검출 기법으로 분할한다. 분할된 선분 세그먼트들 중에서 두 개를 선정하고 RHT를 이용하여 이들의 타원 변수를 추정하고 추정된 변수가

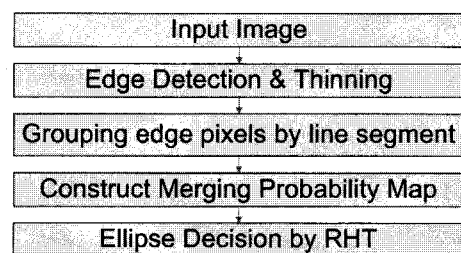


그림 1. 제안하는 알고리즘의 구성도
Fig. 1. Block diagram of the proposed algorithm.

타원 부등식에 만족하면 두 선분 세그먼트를 병합하는 방법을 계속적으로 반복함으로써 영상에 포함된 모든 타원을 검출할 수 있고 결과적으로 타원의 개수를 정확하게 결정하게 된다.

2. 전 처리 과정

입력영상이 주어지면 먼저 에지를 검출하고 세그먼트로 분리하기 위한 세선화 작업을 전처리과정으로 수행한다. 에지 검출기로는 세선화 작업에 유리한 Canny 에지 검출기를 사용한다. Canny 에지 검출기는 가우시안 필터를 사용하여 잡음을 제거하고, 각 화소의 1차 미분 취하여 구한 에지 강도와 방향 값으로 국부 최댓값을 에지점으로 추정하고 두 개의 임계값을 이용해 낮은 임계값으로 검출한 에지점 중에서 높은 임계값으로 검출한 에지점과 연결된 에지점들로 최종 에지를 얻어낸다. 그림 2(b)는 그림 2(a)의 입력 영상을 Canny 에지 검출기로 에지를 검출한 결과를 보여준다.

이와 같이 얻어진 에지영상에서 선분 세그먼트를 분리하기 위해 세선화를 수행한다. 그림 2(c)는 Canny 에지 검출기를 통과한 그림 2(b)의 한 부분을 확대한 그림으로 타원의 에지를 결정하는데 방해가 될 수 있는 불필요한 화소를 보여준다. 이렇게 불필요한 화소는 그림 3에서 보는 바와 같은 L자 모양의 패턴에서 나타난다. 이 패턴은 중심화소의 위쪽이나 아래쪽에 있는 화소 중 하나와 왼쪽이나 오른쪽에 있는 화소 중 하나에 에지점이 존재할 경우에 해당한다. 예를 들어 그림 3(a)의 패턴은 중심화소의 위쪽과 왼쪽에 에지점이 존재하는 경우이다. 이때 대각선 방향의 화소에 에지점의 존

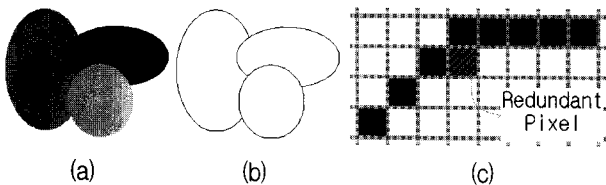


그림 2. (a) 입력 영상, (b) 에지 영상 (c) 불필요한 화소
Fig. 2. Origin image, (b) Edge image (c) Redundant pixel.

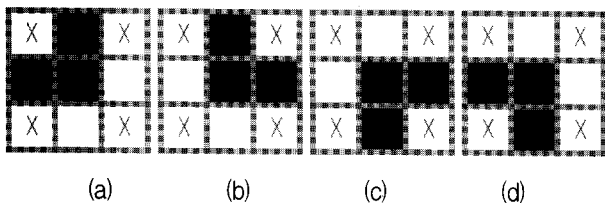


그림 3. 세선화 패턴
Fig. 3. Thinning pattern.

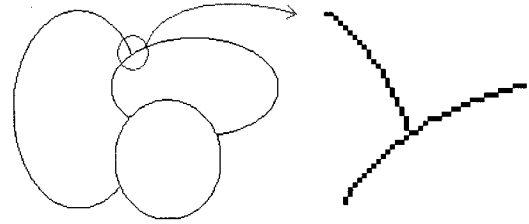


그림 4. 그림 2(b)의 세선화 결과
Fig. 4. Thinning of Fig. 2(b).

재 여부와는 무관하다.

그림 4는 그림 3의 네 개의 패턴을 이용하여 불필요한 화소를 제거하여 세선화한 결과를 보여준다. 이와 같은 세선화 작업의 결과로 에지점의 개수가 적어지게 되고 수행시간을 단축시킬 수 있게 된다.

III. 선분 세그먼트의 구성

RHT를 이용해 타원변수를 결정하기 위해서는 동일한 타원 상에 존재하는 세 점이 필요하다. 그림 2에 보이는 것 같이 영상 안에 두 개 이상의 타원이 존재한다면 이들 중에서 임의로 선택한 세 점을 선택하는 경우, 이들이 모두 동일한 타원에서 선택됐을 확률은 매우 적기 때문에 이와 같은 방법으로 모든 타원을 검출하면 오랜 수행시간이 필요할 뿐 아니라 타원 검출의 정확도도 떨어질 수밖에 없다. 따라서 본 논문에서는 그림 5의 과정을 통해서 에지점들을 선분 세그먼트 단위로 구성하고 RHT를 이용한 타원 검출에 적용하게 된다. 제안된 알고리즘을 그림 6의 입력영상에 적용하여 단계별로 살펴보기로 한다.

그림 6(a)의 에지 영상으로부터 그림 6(b)와 같이 선분 세그먼트를 찾기 위해서는 먼저 그림 6(a)의 분기점

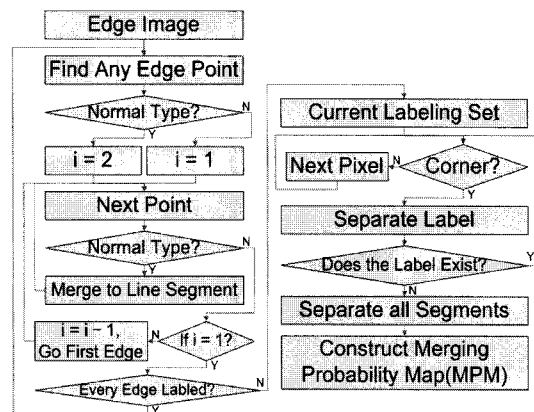


그림 5. 선분 세그먼트 분할 구성도
Fig. 5. Block diagram for constructing line segments.

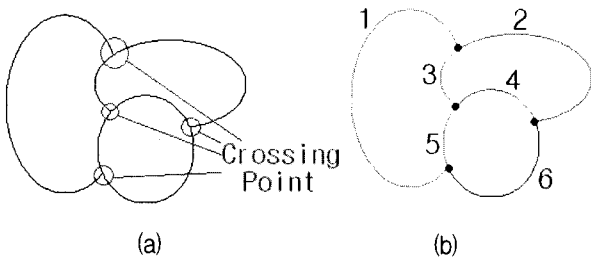


그림 6. (a) 에지의 분기점, (b) 분할된 에지
 Fig. 6. (a) Crossing point in edge, (b) Separated edge.

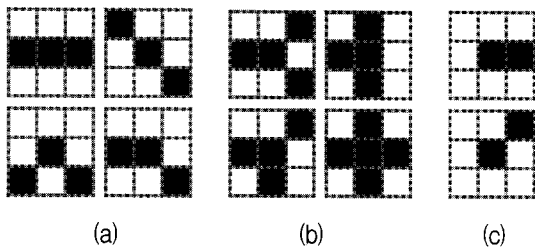


그림 7. 에지점 패턴 (a) 일반점, (b) 교차점, (c) 끝점
 Fig. 7. Pattern of edge point.
 (a)Normal point, (b)Crossing point, (c)End point.

을 이용한 라벨링 기법을 적용한다. 일반적인 라벨링 기법은 연결된 점들을 하나의 라벨로 묶기 때문에 약간의 변형이 필요하다. 일단 관심 에지점에 8방향 이웃 화소를 검사한다. 이 8 방향 화소 중 그림 7(a)처럼 두 개의 에지점이 존재하면 일반 에지점, 그림 7(b)같이 3 개 이상의 에지점이 존재하면 분기점으로 그림 7(c)같이 하나의 에지점이 존재하면 끝점으로 세 가지 패턴을 정의한다. 90도, 180도, 270도씩 회전된 모든 경우도 각 패턴에 속하게 된다. 기본적인 방법은 에지 영상의 좌상단부터 회전하며 에지점을 찾는다. 처음 찾아진 에지점에서 그림 7에 나타난 형태를 조사한다.

일반 에지점이라면 한쪽 방향으로 진행하여 분기점이나 끝점이 나타날 때까지 번호를 할당하며 조사한다. 결국 분기점이나 끝점을 만나고 현재 방향의 라벨링을 멈춘다. 그리고 다시 처음 찾은 일반 에지점으로 돌아가 같은 방법으로 반대 방향으로의 라벨링을 진행하여 하나의 세그먼트를 결정하게 된다. 만약 처음 찾은 에지점이 일반 에지점이 아닌 분기점이면 여러 방향 중 임의의 방향으로 라벨링을 진행하고 끝점이라면 하나의 방향만 존재하므로 그 방향으로 라벨링을 진행하여 다른 분기점이나 끝점을 만났을 경우 라벨링을 멈추고 하나의 세그먼트를 결정하게 된다.

또한 세그먼트 각각을 한쪽 끝점으로부터 화소단위로 저장하여 다음 알고리즘 적용에 사용한다. 이러한 방법을 윤곽선 영상 전체에 적용하면 그림 8과 같이 윤

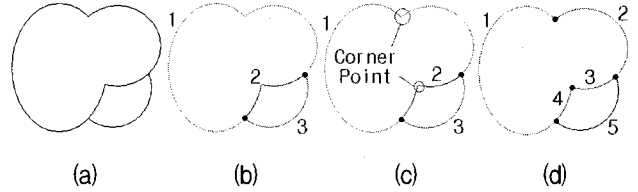


그림 8. (a) 에지 영상, (b) 분할 실패, (c) 교차점, (d) 분할 성공
 Fig. 8. (a) Edge image, (b) Separation failure, (c) Crossing point, (d) Separation success.

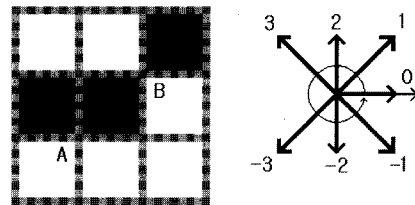


그림 9. 차분 체인코드
 Fig. 9. Difference chain code.

곽선을 분할할 수 있다.

하지만 라벨링만을 이용하여 선분 세그먼트를 분할하는 데는 그림 8의 예에서처럼 한계가 있다. 그림 8(a)의 윤곽선 영상을 라벨링만을 적용해서 나온 결과가 그림 8(b)에 나타난다. 그림 8(b)의 3번 선분 세그먼트는 두 타원이 겹쳐진 경우인데 불구하고 하나의 선분 세그먼트로 분할되었다. 이 선분 세그먼트들은 그림 8(c)의 코너점을 중심으로 두 개의 선분 세그먼트로 나누어야 완벽한 분할이 이루어 졌다고 할 수 있다.

코너점 검출을 위해 미리 코너점을 차분체인코드로 패턴화 하여 저장하고 각 에지세그먼트 별로 적용하여 그림 8(d)와 같이 코너점을 검출한다. 먼저 화소 단위로 저장된 선분 세그먼트를 차분 체인코드화 한다. 차분 체인코드는 그림 9과 같이 현재 화소와 이전화소로 만들어진 벡터(A)와 다음화소와 만들어진 벡터(B) 사이의 각도 차이를 부호화하여 나타낸다.

차분 체인코드는 그림 9에서 A 벡터를 기준으로 B 벡터와의 각도가 반 시계 방향일 때 양의 값을 시계 방향일 때 음의 값을 갖는다. 벡터를 정의할 때 8 방향의 이웃화소만 고려하기 때문에 벡터의 방향이 8개만 존재한다. 그리고 A 벡터와 B 벡터 사이에 각도는 90도가 될 수 없기 때문에 -3부터 3까지 일곱 개의 부호만 존재한다. 이를 이용하여 그림 10에 주어진 코너패턴을 정의할 수 있다.

부호화된 선분 세그먼트의 처음부터 끝까지 진행하면서 코너 패턴과의 정합과정을 거쳐 코너점을 검출한다. 차분체인코드를 사용하면 회전에 무관한 체인코드

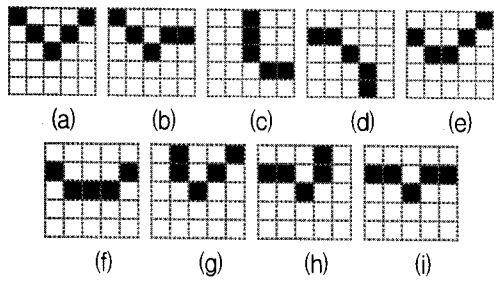


그림 10. 코너 패턴 (a) 020, (b) 02-1, 1-20, (c) 011, -1-10, (d) 101, -10-1, (e) 110, -1-10, (f) 101, -10-1, (g) 120, 0-2-1, (h) -121, -1-21, (i) -12-1, 1-21

Fig. 10. Corner pattern (a) 020, (b) 02-1, 1-20, (c) 011, -1-10, (d) 101, -10-1, (e) 110, -1-10, (f) 101, -10-1, (g) 120, 0-2-1, (h) -121, -1-21, (i) -12-1, 1-21.

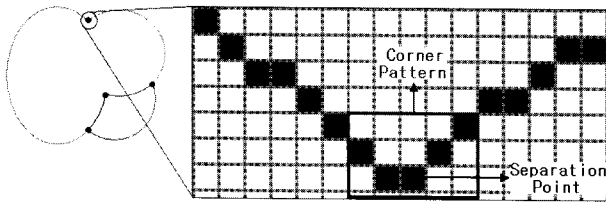


그림 11. 코너 패턴 검출 예, 01-10001100-110-1
Fig. 11. Example to detect corner pattern, 01-10001100-110-1.

를 생성할 수 있으므로 코너패턴의 개수가 일반 체인 코드에 비해 1/4개만 필요하여 빠르게 검출할 수 있다. 그림 11의 오른쪽 확대한 부분에서 코너 패턴 검출의 예를 들어보면 선분 세그먼트의 차분체인코드 01-10001100-110-1의 왼쪽부터 시작해서 세 개의 코드와 그림 10의 코너 패턴들과 비교하며 오른쪽까지 진행된다. 선분 세그먼트의 체인코드 중앙에 있는 110이 코너 패턴 중 존재하므로 1번 선분 세그먼트는 이 화소를 중심으로 두 개의 선분 세그먼트로 분리되어 그림 11 왼쪽과 같이 정확한 선분 세그먼트로 표현된다.

RHT를 이용한 타원 검출을 더욱 효과적으로 하기 위해 앞에서 언급한 병합 가능성도 (Merging Probability Map, MPM)를 정의한다. 인접한 선분 세그먼트끼리 RHT를 적용하여 불필요한 연산량을 줄이고 정확한 타원을 근사하기 위해 MPM을 구성한다. 그림 12는 분리된 선분 세그먼트를 보여준다.

MPM을 구성하는 방법은 선분 세그먼트의 끝점에서 일정 거리 안에 있는 라인 세그먼트들을 병합될 가능성이 있는 선분 세그먼트로 생각하고 그 항목들을 저장해 놓는 것이다. 예를 들어 그림 12에서 선분 세그먼트 1의 MPM을 구성한다고 하면 그림 12의 오른쪽 확대 영

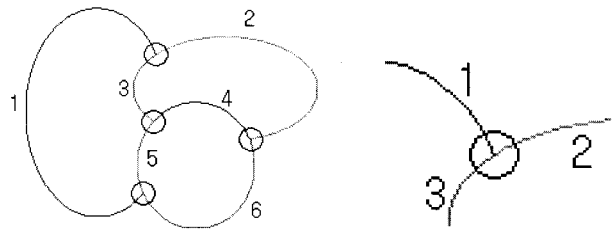


그림 12. 분리된 선분 세그먼트 및 확대 영상
Fig. 12. Separated line segment and megascopic image.

표 1. 그림 12의 선분 세그먼트 MPM
Table 1. MPM of line segment of Fig. 12.

| Line Segment | MPM |
|--------------|------------|
| 1 | 2, 3, 5, 6 |
| 2 | 1, 3, 4, 6 |
| 3 | 1, 2, 4, 5 |
| 4 | 2, 3, 5, 6 |
| 5 | 1, 3, 4, 6 |
| 6 | 1, 2, 4, 5 |

상에서 선분 세그먼트 1의 한 끝점의 일정 거리 안에 포함되는 선분 세그먼트 2와 3이 존재함을 알 수 있다. 이러한 방법으로 반대쪽 끝점에서도 인접 선분 세그먼트를 조사하면 라인세그먼트 1의 MPM을 선분 세그먼트 2, 3, 5, 6으로 구성할 수 있다. 표 1에 그림 12의 선분 세그먼트 각각의 MPM이 나타나 있다.

IV. 타원 결정

1. Randomized Hough Transform

HT은 유클리드 공간에서 계수 공간으로의 변환으로부터 시작된다. 일반적인 2차원 곡선 방정식은 식 (1)로 주어진다.

$$ax^2 + by^2 + cxy + dx + ey + 1 = 0 \quad (1)$$

이 방정식은 2차원 공간 내에서 표현할 수 있는 모든 종류의 곡선을 계수 (a,b,c,d,e)의 조합으로 나타낸다. 이 식은 중심점이 (p,q)인 타원의 식으로 다음과 같이 표현할 수 있다.

$$A(x-p)^2 + 2H(x-p)(y-q) + B(y-q)^2 = 1 \quad (2)$$

$$AB - H^2 > 0 \quad (3)$$

이 때 식 (3)의 부등식을 만족하는 경우에 식 (2)가 표현하는 2차 곡선이 타원이 된다. 일반적인 HT을 이용하여 타원 검출할 경우 (A,H,B,p,q) 다섯 개의 계수로

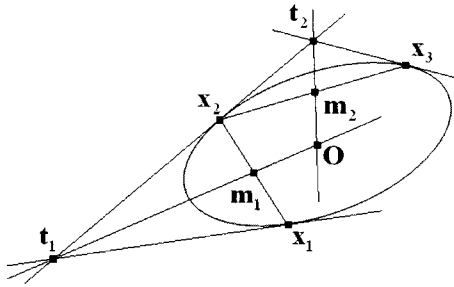


그림 13. 타원의 중심 추정
Fig. 13. Estimate center of ellipse.

이루어진 5차원 공간이 필요하므로 매우 많은 저장 공간과 긴 수행시간이 필요하다. 이 문제를 해결하기 위해 Yuen et al^[2]은 다음과 같은 방법을 제시했다. 타원에서 임의의 두 점(x1,x2)을 택하고 두 점 각각의 접선이 만나는 교점(t1)과 두 점의 중점(m1)으로 이루어진 직선은 항상 타원의 중심점(O)을 포함한다. 그림 13은 이 방법을 이용하여 타원의 중심점 추정을 보여준다.

타원에서 임의의 세 점 (x1,x2,x3)을 선택하고 두 점 (x1,x2)의 중점 (m1)과 접선의 교점 (t1)을 연결한 직선 $\overline{t_1m_1}$ 과 (x2,x3)의 중점 (m2)과 접선의 교점 (t2)을 연결한 직선 $\overline{t_2m_2}$ 의 교점 O를 타원의 중심점으로 추정한다. 식(2)의 (p,q)점이 추정된 타원의 중심이 되고 식 (2)를 원점으로 평행 이동시키면 식 (4)와 같이 표현될 수 있다.

$$Ax^2 + 2Hxy + By^2 = 1 \tag{4}$$

이로서 구해야 할 계수가 (A,H,B) 세 개로 줄어든다. 구해야 할 계수가 세 개이므로 타원에 속하는 임의의 세 점 $X_1 = (x_1, y_1), X_2 = (x_2, y_2), X_3 = (x_3, y_3)$ 이 주어지면 식 (5)를 풀어 (A,H,B)값을 추정할 수 있다.

$$\begin{pmatrix} x_1^2 & 2x_1y_1 & y_1^2 \\ x_2^2 & 2x_2y_2 & y_2^2 \\ x_3^2 & 2x_3y_3 & y_3^2 \end{pmatrix} \begin{pmatrix} A \\ H \\ B \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tag{5}$$

하지만 구해진 (A,H,B)값이 모두 타원의 계수가 되지 않는다. 앞에서 언급한 바와 같이 식 (3)에 만족하는 경우의 값들만 3차원 공간에 투표되고 가장 많이 투표된 (A,H,B)값으로 타원을 결정한다.

2. MPM을 이용한 선분 세그먼트 병합

여기에서는 완벽히 분할된 선분 세그먼트를 같은 타원에 존재하는 선분 세그먼트끼리 병합하는 작업이 설명 된다. 병합을 수행하는 방법은 그림 14와 같다.

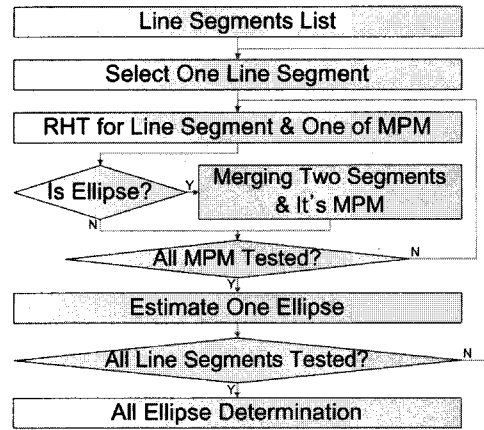


그림 14. 선분 세그먼트 병합 구성도
Fig. 14. Block diagram for segments merging.

먼저 분할된 선분 세그먼트 중 하나의 선분 세그먼트를 선택한다. 그리고 그 라인세그먼트의 MPM에 속하는 선분 세그먼트 중 임의의 하나와 같이 4.1절에서 설명한 RHT를 적용하여 타원을 추정하고 두 선분 세그먼트와 추정 타원 사이의 관계를 타원 정합비 (Ellipse Matching Ratio, EMR)로 정의한다.

$$EMR = \frac{N_c}{N_w} \tag{6}$$

식 (6)에서 N_w 는 두 선분 세그먼트에 속하는 모든 화소의 개수이고 N_c 는 선분 세그먼트에 속하는 모든 화소 중 추정 타원과 근접한 화소의 개수이고 다음과 같이 정의 된다.

$$N_c = \sum_{i=1}^{N_w} Near(i) \tag{7}$$

$$Near(i) = \begin{cases} 1 & \text{if } \overline{P_iQ_i} < th \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

식 (8)의 $\overline{P_iQ_i}$ 는 선분 세그먼트에 속하는 임의의 점 Q_i 와 추정 타원의 중점 O와 Q_i 로 만들어진 직선과 추정 타원이 만나는 점 P_i 사이의 거리이고 그림 14(a)와 같이 나타난다. 그리고 th 는 실험에 의해 결정된다. 식 (6)-(8)로 계산된 EMR 값에 의해 두 세그먼트의 병합을 판단하게 된다. EMR 값은 선분 세그먼트로 추정된 타원의 정확도를 측정하는 척도가 되기 때문에 높은 임계값으로 설정해야 한다. 본 논문에서는 EMR을 0.95로 놓았다. 만약 임의의 두 선분 세그먼트의 EMR이 0.95를 넘는다면 두 세그먼트를 병합하고 각각 나누어져 있던 MPM들을 하나로 통합하게 된다.

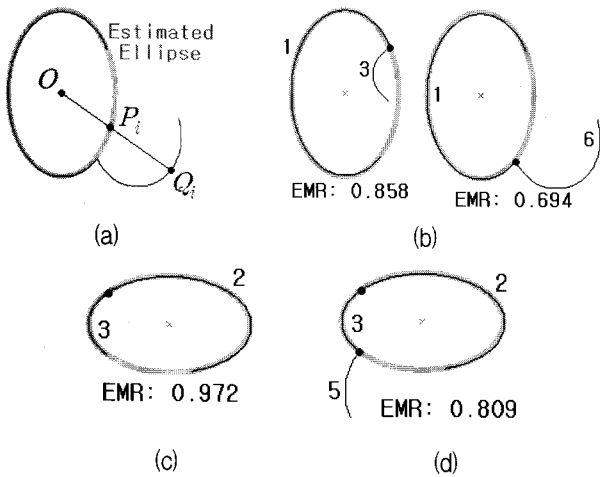


그림 15. 병합 결정, (a) EMR, (b) 선분 세그먼트 1과 MPM의 EMR, 병합 불가능, (c) 선분 세그먼트 2와 MPM의 EMR, 병합 가능, (d) 선분 세그먼트 2, 3과 MPM의 EMR, 병합 불가능

Fig. 15. Merging decision, (a) EMR, (b) EMR of line segment 1 and it's MPM, Merging impossible, (c) EMR of line segment 2 and it's MPM, Merging possible, (d) EMR of line segment 2, 3 and it's MPM, Merging impossible.

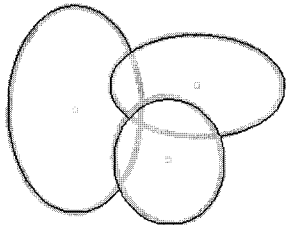


그림 16. 그림 12에서의 타원 추출
Fig. 16. Ellipse extraction of Fig. 12.

그림 12의 에지 영상에서 예를 들어 보면 선분 세그먼트 1의 MPM(표 1)은 4개의 선분 세그먼트 2, 3, 5, 6으로 구성되어 있다. MPM 중 선분 세그먼트 3과 6의 경우 각각 타원을 추정하여 EMR을 구하면 그림 15(b)와 같이 각각 0.858과 0.694로 계산된다. 두 경우는 사람이 판단하기에도 다른 타원에 속하는 선분 세그먼트이고 EMR 값도 우리가 정한 임계값보다 낮기 때문에 병합되지 않는다. 나머지 두 선분 세그먼트 2, 5의 경우도 마찬가지다. 선분 세그먼트 2의 MPM(표 1)에 속하는 선분 세그먼트 1, 3, 4, 6 중 그림 15(c)에 나타난 선분 세그먼트 3의 경우 계산된 EMR이 0.972로 임계값보다 높기 때문에 두 선분 세그먼트 2, 3은 병합되고 각각의 MPM(표 1)은 선분 세그먼트 1, 4, 5, 6을 갱신된다. 위와 같은 방법으로 모든 선분 세그먼트에서 타원 검출을 수행하면 그림 16과 같이 에지 영상에서 3개의 타원을 추출할 수 있다.

V. 실험 및 결과

본 장에서는 기존의 HT 중 RHT^[5]과 Yonghong Xie^[7], Fast Graphical Ellipse Detection (FGED)^[8] 세 가지 알고리즘과 제안하는 방법의 타원 검출 수행 시간 및 정확성에 대한 성능 비교와 타원 개수 추정을 인조 영상에서 비교하고 실제 영상에 대한 제안하는 알고리즘의 성능 결과를 보여준다. 인조 영상의 경우 1개부터 9개의 타원이 존재하는 영상 각각을 10종류씩 총 90개의 영상에 대하여 각각 100번을 실험하여 결과를 산출하였다. 실제 영상의 경우 Bin Picking에 사용되는 산업 물체를 여러 가지 경우로 쌓아 놓고 적용했다.

그림 17에 실험한 인조 영상 중 하나의 예가 있다. 그림 17(a)의 원 영상에는 서로 크기가 다른 4개의 타원이 존재한다. 회색의 굵은 타원이 원 영상으로부터 추정된 타원이다. 그림 17(b)-(e)는 원 영상에 RHT, Yonghong Xie의 알고리즘, FGED, 제안하는 방법을 적용한 결과이다. RHT를 적용한 그림 17(b)의 결과를 보면 좌측 상단의 작은 타원을 검출하지 못했고 정확성이 비교적 떨어진다. 또한 voting 임계치 이상의 모든 타원이 검출되기 때문에 중복된 타원을 검출하는 경우도 있다. Yonghong Xie의 알고리즘을 적용한 그림 17(c)의 결과는 3개의 큰 타원에 대해서는 정확히 검출했지만 RHT와 마찬가지로 좌상단의 작은 타원은 검출하지 못

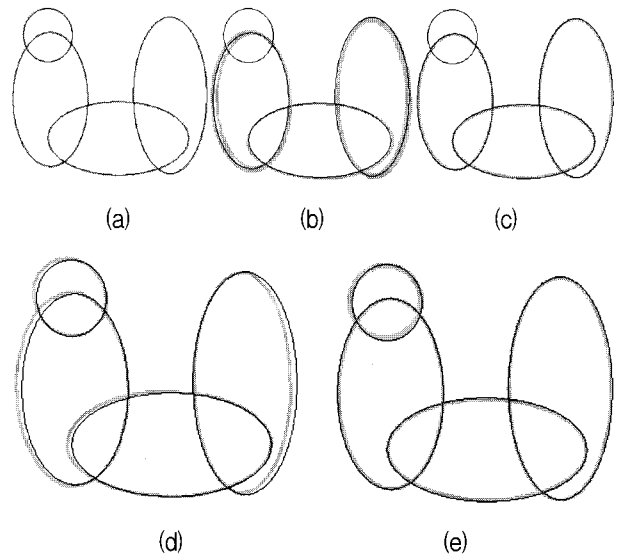
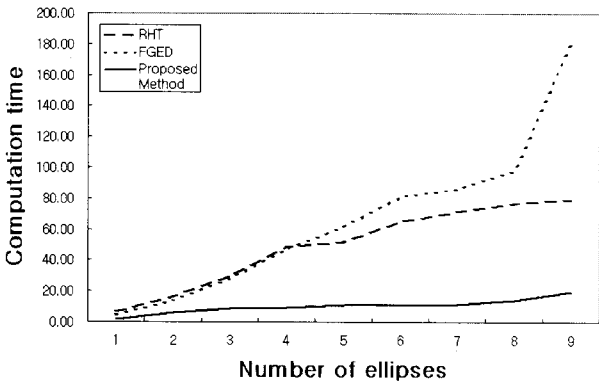
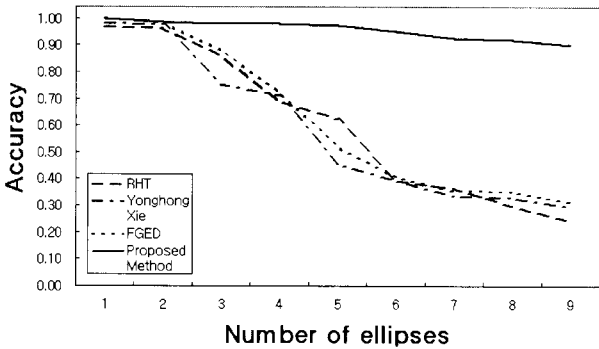


그림 17. 인조 영상에서 타원 추출 예, (a) 원 영상, (b) RHT, (c) Yonghong Xie, (d) FGED, (e) Proposed method

Fig. 17. Example of ellipse extraction in synthetic image, (a) Origin image, (b) RHT, (c) Yonghong Xie, (d) FGED, (e) Proposed method.



(a)



(b)

그림 18. RHT와 Yonghong Xie의 알고리즘, FGED, 제안하는 방법 간의 성능 비교, (a) 타원의 개수에 대한 계산 시간, (b) 타원의 개수에 대한 정확도

Fig. 18. Comparing the performance of RHT, Yonghong Xie's algorithm, FGED and proposed method, (a) Computation time versus number of ellipse, (b) Accuracy versus number of ellipses.

했다. FGED를 적용한 그림 17(d)의 결과는 4개의 타원을 모두 검출했지만 정확도가 다른 알고리즘에 비해 가장 떨어진다. 제안하는 방법을 적용한 그림 17(e)의 결과는 원 영상에 존재하는 4개의 타원을 모두 검출했다.

모든 인조 영상에 대한 성능비교가 그림 18과 표 2에 나와 있다. 수행시간의 경우 영상 내의 타원의 개수가 적은 경우 Yonghong Xie의 알고리즘을 제외한 나머지 알고리즘들과 제안하는 방법이 약간의 차이를 갖는다. 하지만 영상 내의 타원의 개수가 증가할수록 수행시간은 현격히 차이가 나게 된다. 제안하는 방법을 제외한 나머지 방법들은 타원의 개수가 많아질수록 수행시간이 급격히 증가하지만 제안하는 방법의 수행시간은 완만히 증가하였다. 표 2의 수행시간 비교 결과를 보면 제안하는 방법이 다른 방법에 비해 월등히 빠르게 타원을 검출함을 알 수 있다.

그림 18(b)의 그래프는 제안하는 방법과 다른 방법들

표 2. RHT와 Yonghong Xie의 알고리즘, FGED, 제안하는 방법 간의 성능 비교

Table 2. Comparing the performance of RHT, Yonghong Xie's algorithm, FGED and proposed method.

| | Computation time (sec) | | | | Accuracy | | | |
|---|------------------------|--------------|-------|-----------------|----------|--------------|-------|-----------------|
| | RHT | Yonghong Xie | FGED | Proposed Method | RHT | Yonghong Xie | FGED | Proposed Method |
| 1 | 6.77 | 51.503 | 4.131 | 1.702 | 0.97 | 0.981 | 0.998 | 1.000 |
| 2 | 16.5 | 83.910 | 13.62 | 5.945 | 0.96 | 0.977 | 0.982 | 0.997 |
| 3 | 29.4 | 287.06 | 27.54 | 8.718 | 0.86 | 0.753 | 0.879 | 0.990 |
| 4 | 48.2 | 1386.5 | 46.69 | 9.398 | 0.69 | 0.713 | 0.725 | 0.988 |
| 5 | 51.4 | 1386.5 | 61.29 | 10.74 | 0.62 | 0.456 | 0.520 | 0.973 |
| 6 | 64.9 | 1419.9 | 81.20 | 10.95 | 0.39 | 0.390 | 0.409 | 0.953 |
| 7 | 71.4 | 1452.3 | 85.94 | 11.60 | 0.36 | 0.334 | 0.355 | 0.925 |
| 8 | 76.5 | 1985.8 | 97.85 | 14.36 | 0.30 | 0.329 | 0.351 | 0.920 |
| 9 | 79.4 | 2105.1 | 181.0 | 19.56 | 0.24 | 0.293 | 0.312 | 0.901 |

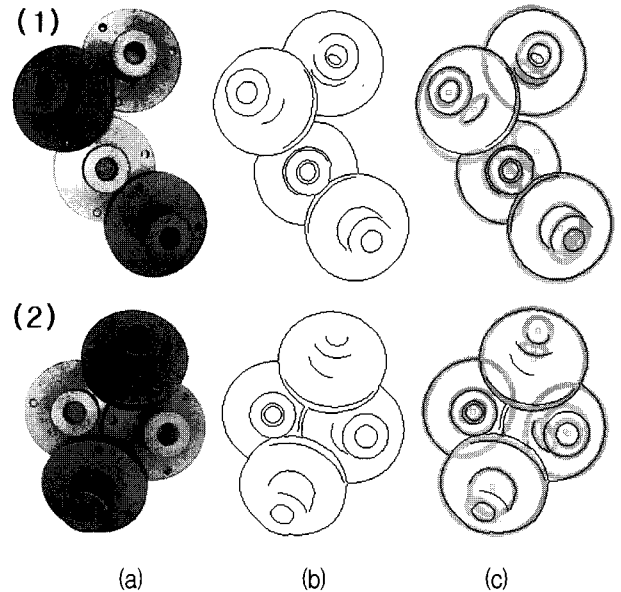


그림 19. 실제 영상에서의 타원 검출 (a) 실제 영상, (b) 에지 영상, (c) 제안하는 방법의 타원 검출

Fig. 19. Ellipse detection in real-world images, (a) Real-world image, (b) Edge image, (c) Ellipse extraction by proposed method.

의 타원 검출 정확도를 비교한 것이다. 영상 내의 타원의 개수가 1개나 2 개로 적을 때는 모든 방법이 비교적 정확한 타원을 검출하지만 타원의 개수가 많아지면 제안하는 방법을 제외한 나머지 방법들의 정확도는 20%까지 떨어지게 되어 타원 검출에 어려움이 있다. 하지만 표 2의 정확도 비교 결과를 보면 제안하는 방법은 영상 내의 타원이 많아져도 90%이상 정확하게 타원을 검출할 수 있다.

그림 18(a)의 그래프는 입력 영상 내 타원의 개수가

표 3. 그림 18(1)의 실제영상에서 검출된 타원 변수
Table 3. Parameter values of ellipses detected in Fig. 18(1).

| Center | Major axis | Minor axis | Orientation |
|------------|------------|------------|-------------|
| (170, 85) | 51 | 47 | 42.03297 |
| (146, 238) | 51 | 47 | -59.8352 |
| (84, 199) | 54 | 47 | 81.59341 |
| (66, 210) | 24 | 20 | -64.7802 |
| (98, 120) | 34 | 20 | -54.8901 |
| (150, 243) | 14 | 10 | -76.6484 |
| (180, 76) | 24 | 20 | 56.86813 |
| (66, 210) | 14 | 10 | 53.9011 |
| (122, 137) | 24 | 24 | 89.5055 |
| (172, 85) | 24 | 20 | 33.13187 |
| (130, 137) | 27 | 24 | -5.43956 |
| (140, 227) | 17 | 10 | -31.1538 |
| (124, 137) | 10 | 10 | -45.989 |
| (182, 74) | 14 | 10 | 43.02198 |
| (90, 194) | 14 | 7 | 49.94506 |
| Time (sec) | | | 12.227 |

표 4. 그림 18(2)의 실제영상에서 검출된 타원 변수
Table 4. Parameter values of ellipses detected in Fig. 18(2).

| Center | Major axis | Minor axis | Orientation |
|-----------------------|------------|------------|-------------|
| (181, 127) | 50 | 44 | 71.47059 |
| (61, 165) | 50 | 44 | 79.94118 |
| (132, 195) | 22 | 22 | 77.82353 |
| (130, 195) | 44 | 38 | -14.2941 |
| (108, 103) | 44 | 41 | -25.9412 |
| (112, 109) | 22 | 22 | 77.82353 |
| (132, 197) | 13 | 9 | -61.9412 |
| (112, 109) | 13 | 9 | -46.0588 |
| (134, 175) | 16 | 9 | 30.17647 |
| (196, 127) | 16 | 9 | 57.70588 |
| (205, 117) | 9 | 9 | 56.64706 |
| (68, 155) | 13 | 6 | 49.23529 |
| (37, 169) | 13 | 13 | -71.4706 |
| (112, 111) | 9 | 9 | 43.94118 |
| Processing time (sec) | | | 9.503 |

증가함에 따라 타원 검출 정확도를 나타낸다. 영상 내 타원의 개수가 적을 때는 기존의 두 방법과 제안하는 방법 모두 높은 정확도를 나타내지만 영상 내 타원의 개수가 증가함에 따라 타원 검출 정확도가 현격히 벌어짐을 알 수 있다. 두 방법을 그림 18(b)의 그래프와 함께 비교해 보면 그 차이를 정확히 알 수 있다.

그림 18(b)의 그래프는 각 영상에서 추정된 타원의 개수를 나타낸다. 기존의 두 방법의 경우 검출된 타원의 개수가 입력 영상 내 타원의 개수와 많은 차이를 보이고 이것은 잘못된 타원을 추정했다고 이해 할 수 있다. 반면에 제안하는 방법의 경우 검출된 타원의 개수와 입력 영상 내 타원의 개수가 정확하게 일치하므로

기존의 방법에 비해 타원을 정확하게 검출했다고 판단할 수 있다.

그림 19는 실제 산업 부품을 임의로 쌓아 놓은 영상에서 제안하는 방법을 이용한 타원 검출을 보여준다. 실제 영상은 인조영상과 다르게 내부에 많은 에지들이 존재하기 때문에 물체의 외형뿐만 아니라 내부에 존재하는 타원도 모두 찾아야 한다. 표 3과 표 4에 각각의 실제 영상에서 검출된 타원들의 정보가 나와 있다.

표 3과 4의 결과를 보면 실제 영상에 존재하는 대부분의 타원을 매우 빠른 시간에 찾아내었다.

인조 영상과 실제 영상에서의 실험에 대한 결과는 제안하는 알고리즘이 기존의 방법에 비해 빠르고 정확하므로 타원검출에 매우 효과적임을 보여준다.

VI. 결 론

본 논문에서는 선분 세그먼트기반의 randomized Hough transform의 타원 검출방법을 제안 하였다. 본 논문은 기존의 Hough transform을 이용한 타원 검출 방법 중 RHT^[5] 및 Yonghong Xie^[7], FGED^[8]의 알고리즘과 성능을 비교하였다. 제안하는 방법은 기존의 방법에서 모든 에지점을 사용하여 타원을 추정한 것과 다르게 연속된 에지점의 모임을 하나의 선분 세그먼트로 정의하여 선분 세그먼트를 이용한 방법을 적용하였다. 전체 에지를 세그먼트 단위로 분할한 후 세그먼트 사이의 RHT를 이용한 타원 추정으로 세그먼트간의 병합을 통해 하나의 타원을 이루는 집합을 형성한다. 이로써 정확한 타원을 검출할 수 있었으며 타원 검출에 필요한 반복 수행시간도 크게 개선할 수 있었다. 또한 RHT뿐만 아니라 대부분의 HT를 이용한 타원 검출에서의 문제점인 타원의 개수 추정을 정확하게 해결할 수 있었다.

참 고 문 헌

- [1] P. V. C. Hough, "Method and Means for Recognizing Complex Patterns," U. S. Patent 3069654, Dec. 18 1962.
- [2] H. K. Yuen, J. Illingworth, and Kittler, "Detecting partially occluded ellipses using the Hough Transform," Image and Vision Computing, vol. 7, no. 1, pp. 31-37, Feb. 1989.
- [3] N. Kiryati, Y Eldar, and A. M. Bruckstein, "A probabilistic Hough Transform," Pattern

- Recognition, vol. 24, no. 4, pp. 303-316, 1991.
- [4] Sheng-Ching Jeng, Wen-Hsuang Tsai, "Scale and orientation-invariant generalized Hough Transform-A new approach," Pattern Recognition, vol. 24, no. 11, pp. 1034-1051, 1991.
- [5] Lei Xu, Erkki Oja, and Pekka kultanena, "A new curve detection method: Randomized Hough Transform (RHT)," Pattern Recognition Letters, vol. 11, no. 5, pp. 331-338, May. 1990.
- [6] Robert A. McLaughlin, "Randomized Hough Transform: better ellipse detection," Digital Signal Processing Applications, vol. 1, pp. 409-414, Nov. 1996.
- [7] Yonghong Xie, Qiang Ji, "A new efficient ellipse detection method," Pattern Recognition, vol. 2, pp. 957-960, Aug. 2002.
- [8] Elmowafy, O.M., Fairhurst, M.C., "Improving ellipse detection using a fast graphical method," Electronics Letters, vol. 35, no 2. pp. 135-137, Jan. 1999.
- [9] McLaughlin, R.A., Alder, M.D, "The Hough transform versus the UpWrite," Pattern Analysis and Machine Intelligence, IEEE Transactions on vol. 20, no. 4, pp. 396-400, Apr. 1998.

 저 자 소 개



한 광 수(학생회원)
 2006년 숭실대학교 정보통신전자
 공학부 학사 졸업.
 2006년~현재 숭실대학교
 전자공학과 석사과정.
 <주관심분야 : 컴퓨터 비전, 영상
 처리>



한 영 준(정회원)
 1996년 숭실대학교 전자공학과
 학사 졸업
 1998년 숭실대학교 전자공학과
 석사 졸업
 2003년 숭실대학교 전자공학과
 박사 졸업
 2003년~현재 숭실대학교 정보통신전자공학부
 조교수
 <주관심분야 : 로봇트 비전, 영상처리, 비주얼서
 보잉>

한 현 수(정회원)
 대한전자공학회 논문지
 제38권 SP편 제 6호 참조