

동영상에서 움직임을 이용한 빠른 허프 원 찾기

Fast Hough circle detection using motion in video frames

원 혜 민* 이 경 미**
Hye-min Won Kyoung-Mi Lee

요 약

영상에서 원을 찾을 때 정확도가 높은 일반화 허프 변환 (Generalized Hough Transform: GHT) 알고리즘이 가장 많이 사용된다. 그러나 GHT는 영상의 모든 가능한 픽셀에 대해 가능한 모든 원의 크기를 확인해야하기 때문에 많은 계산 시간을 요구한다는 단점을 가지고 있다. 또한 동영상에서 원을 찾는다면, 매 프레임마다 이러한 GHT를 실행해야하므로 계산 속도는 더욱 느리게 된다. 본 논문에서는 동영상의 연속된 프레임 사이에서 변화되는 부분이 적고 대부분 이전 영상과 유사하다는 점을 이용하여, GHT의 정확도를 유지하면서 계산 속도를 줄이는 알고리즘을 제안한다. 제안하는 알고리즘은 동영상의 현재 프레임과 이전 프레임을 비교하여 움직임이 크면 본래의 GHT를 수행하는 변화 기반 방법과 움직임이 작으면 이전 프레임에서 검출한 원의 정보로 GHT의 후보영역과 후보 원들만을 이용하여 GHT를 수행하는 경로 기반 방법을 사용하였다. 이는 동영상에서의 GHT 수행에 영향을 미치는 프레임 수, 에지 수, 원의 개수를 줄임으로써 속도를 향상시킬 수 있다. 실험결과는 제안하는 알고리즘이 고정된 카메라나 이동 카메라에 대해 정확도의 손실이 없으면서, 수행 시간을 줄이고 있음을 보여준다.

ABSTRACT

The Generalized Hough Transform(GHT) is the most used algorithm for circle detection with high accuracy. However, it requires many computation time, because many different templates are applied in order to find circles of various size. In the case of circle detection and tracking in video, the classical approach applies GHT for each frame in video and thus needs much high processing time for all frames. This paper proposes the fast GHT algorithm in video, using two consecutive frames are similar. In the proposed algorithm, a change-driven method conducts GHT only when two consecutive frames have many changes, and trajectory-based method does GHT in candidate areas and with candidate radius using circles detected in a previous frame. The algorithm can reduce computation time by reducing the number of frames, the edge count, and the number of searching circles, as factors which affects the speed of GHT. Our experimental results show that the algorithm successfully detects circles with less processing time and no loss of accuracy in video acquired by a fixed camera and a moving camera.

☞ keyword : Hough Transform, Circle detection, Real-time processing, 허프 변환, 원 찾기, 실시간 처리

1. 서 론

영상에서 원을 찾는 것은 많은 응용에 필요한 영상 인식 기술이다. 김중선 등은 인간의 얼굴영역 크기를 찾을 때의 오차를 줄이기 위해 원형 탐색을 통해 어깨선을 찾았고[1], Gupta 등은 동공의

경계와 홍체의 외곽 부분을 찾기 위해 허프 변환을 이용한 원 찾기를 수행했으며[2], Pei 등은 스포츠 방송 영상에서 룰 기반 알고리즘을 이용하여 축구공을 찾았다[3]. 또한 실시간 카메라 영상에서 인터페이스로써 원을 찾는 응용 프로그램들도 있고[4], 김상희 등은 원 형상의 일부만 존재하더라도 원 찾기를 수행했다[5].

원을 찾는 많은 알고리즘 중 가장 많이 사용되는 것은 허프 변환(Hough Transform)이다. 그 중 일반화 허프 변환 (Generalized Hough Transform: GHT)은 디지털 영상 내에서 임의의, 또는 미리

* 준 회 원 : 덕성여자대학교 대학원 전산정보통신학과 석사과정
haemini86@duksung.ac.kr

** 정 회 원 : 덕성여자대학교 컴퓨터학과 교수
kmlee@duksung.ac.kr

[2010/08/05 투고 - 2010/08/07 심사(2010/10/11 2차) - 2010/10/29 심사완료]

알려진 형태를 찾을 수 있는 매우 강력하고 잘 알려진 템플릿 매칭 알고리즘이다[6]. GHT는 잡음이 있거나 다중 물체인 경우에 형태를 분석하는데 유용하고 정확도는 높으나, 영상의 모든 가능한 픽셀에 대해 가능한 모든 원의 크기를 확인해야하기 때문에 많은 계산 시간을 요구한다는 단점이 있다. 더욱이 동영상에서 원을 찾아 추적하는 경우라면, 매 프레임마다 이러한 허프 변환을 실행해야하므로 계산 속도는 더욱 느리게 된다. 따라서 동영상에서 허프 변환 시 수행 속도에 대한 개선이 필요하다.

본 논문에서는 실시간 카메라 동영상의 연속된 프레임 사이에서 변화되는 부분이 적고 대부분 이전 프레임과 유사하다는 점을 이용하여, 허프 변환 계산 속도를 줄이는 방법을 제안한다. 즉, 모든 프레임에 대해 허프 변환을 실행하는 대신에 움직임을 활용하여 프레임 사이의 중복을 없애는 방법과 각 프레임의 전체 영역 대신에 이전 프레임으로부터 원의 위치 및 크기를 추측하여 프레임 내 중복을 없애는 방법을 제안한다.

본 논문의 2장에서는 연구에 대한 배경으로 일반화 허프 변환과 동영상에서의 빠른 원 찾기에 대한 내용을 다룬다. 3장에서는 동영상에서 움직임을 이용한 빠른 원 찾기를 위한 알고리즘을 제안한다. 또 실험 환경과 실험 결과 내용을 4장에서 기술하고, 5장에서 결론을 맺는다.

2. 배경

2.1 일반화 허프 변환 (Generalized Hough Transform : GHT)

일반화 허프 변환 (GHT)은 원하는 형태의 위치를 찾는 투영 기반 물체 알고리즘으로[6], 에지를 검출한 후 모든 연결 가능한 포인트에 직선을 그어 높게 투표된 부분을 원의 원점의 후보로 놓고 원점과 각 포인트 간의 거리를 이용하여 원의 반지름을 찾는다[7]. GHT의 (x, y) 로 표시된 영상 공간과 일치하는 누산기의 집합으로 나누어진 매개

공간에서 영상 내 각 에지 화소는 매개변수에 연결된 누산기 내 셀을 결정하는 매핑에 의해 변환된다. 선택된 셀의 값은 누적되고, 이후 영상 내 모든 화소가 처리되며, 누산기 내 국부 최대치가 특정 형태의 매개변수에 대응하게 된다.

GHT를 이용하여 원을 찾기 위해서는 찾고자 하는 원의 크기가 미리 정의되어야 한다. R-테이블로 불리는 원의 검색 테이블은 기울기 방향과 에지 화소로부터 치환사이의 관계로 구성될 수 있다. 더욱이, 찾고자 하는 원의 크기가 다양하고 미리 알려지지 않은 경우에는, 주로 GHT 과정에서 입력 형태의 가능한 모든 크기가 강제적으로 계산하게 된다. 이는 수행시간이 매우 급격하게 증가하게 만든다.

영상에서 원을 찾기 위한 GHT의 수행 복잡도는 $O(n_r n_i \frac{n_t}{R})$ 이다. 여기서 n_r 은 다양한 크기의 원의 개수, n_i 는 영상 내 에지 픽셀의 개수이고, n_t 는 템플릿 내 에지 픽셀의 개수로 크기에 따라 다르다. R은 R-테이블의 해상도이다.

2.2 동영상에서 빠른 원 찾기

정지 영상에서 원을 찾을 때 정확성의 이유로 GHT를 많이 사용한다. 동영상에서 원을 찾을 때, 가장 전통적인 방법은 매 프레임마다 GHT를 실행하는 것인데, 계산 속도는 매우 느리다.

이러한 문제점을 해결하기 위해 여러 방법들이 제안되었다. 우선 GHT 자체의 속도를 향상시키는 연구가 있었는데, Ma와 Chen은 기울기와 곡률을 이용한 알고리즘을 제안하였다(SC-GHT) [8]. SC-GHT는 α 와 β 가 각각 템플릿과 프레임의 기울기 각도라고 가정하고, $\tan(\alpha - \beta) = \tan(\Theta)$ 를 만족하는 템플릿에 대한 프레임의 회전의 각도 Θ 를 계산한다. Franke 등은 허프로 빠른 원 검출을 위해 찾고자 하는 공의 크기의 개수 n_r 를 최대한 줄였다[9].

또한, 동영상의 움직임을 이용하여 속도를 향상 시킨 연구도 이루어졌다. Aizawa 등은 이미지

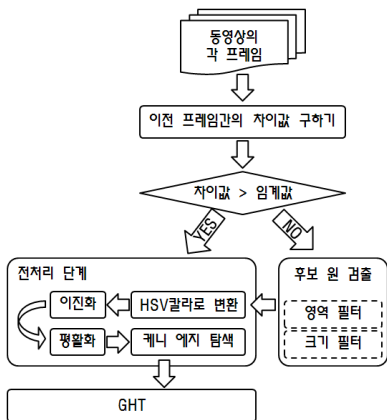
(표 1) 동영상에서 원 찾기에 대한 기존 연구

	전통적인 방법			움직임 이용
	GHT[6]	SC-GHT[8]	Franke 등[9]	Pardo 등[12]
수행 시간	$O(fn_r n_i \frac{n_t}{R})$	$O(fn_i n_i)$	$O(fn_r n_i \frac{n_t}{R})$	$O\left(n_r n_i \frac{n_t}{R} + (f-1)wh\right)$
장점	정확도가 높음	속도가 빠름	속도가 빠름	속도가 빠름
단점	속도가 느림	잡음에 민감함	원의 크기를 미리 알아야 함 적은 개수의 원에만 유효함	잡음에 민감함 동영상 내 원의 개수가 바뀌면 검출 어려움

센서를 이용해 카메라에서 전달된 데이터의 양을 감소시켰고[10], 이런 데이터의 감소는 이미지 처리의 속도를 향상시키는데 사용되었다[11]. Pardo 등은 하드웨어와 센서가 아닌 소프트웨어를 사용한 방식을 제안하였는데, 잡음에 민감한 임계값을 이용하여 허프 변환으로 원을 찾은 후 에지의 움직임 상태를 통해 원을 추적함으로써 동영상에서 원 찾기의 계산 시간을 줄였다[12].

표 1은 전통적인 방법과 움직임을 이용한 방법의 수행시간, 장단점을 보여주고 있다. 여기서 f 는 동영상 내 총 프레임 수, w 와 h 는 프레임의 가로, 세로 길이이다.

3. 움직임을 이용한 빠른 원 찾기



(그림 1) 전체 알고리즘

본 논문에서는 동영상에서 허프 변환의 속도 개선을 위해 동영상의 움직임을 이용하여 원을 찾는 방법을 제안한다. 변화(Change) 기반 방법은 동영상의 각 프레임이 획득되면 이전 프레임간의 차이값을 구하고, 차이값이 클 경우에만 허프 변환 계산을 수행한다. 경로(Trajectory) 기반 방법은 차이값이 작을 경우에 이전 프레임에서 찾은 원 정보를 이용하여 허프 변환의 매개공간 내 탐색 영역 및 탐색할 원의 개수를 줄인다. 그림 1은 제안하는 알고리즘의 전체 순서도를 보여주고 있다.

3.1 이전 프레임 간 차이값 구하기

변화 기반 방법은 이전 프레임과의 차이값을 이용하여 GHT의 수행 여부를 결정한다. 프레임 간의 차이값은 이전 프레임 I^{t-1} 과 현재 프레임 I^t 의 차이로 다음과 같이 계산된다:

$$\text{차이값} = \frac{\sum_{i=0}^{wh-1} (I_i^t - I_i^{t-1})}{wh}$$

여기서 I_i^{t-1} 과 I_i^t 는 각 프레임의 i 번째 픽셀을 말한다. 즉, 차이값은 이전 프레임과의 픽셀당 평균 차이값을 말하며, 대부분 0에서 5이하의 값이다. 본 논문에서는 임계값을 0에서 5까지의 값으로 설정하고 차이값이 임계값 이상일 경우에만 GHT를 수행했다.

3.2 GHT의 전처리 단계

GHT를 수행하기 전에 전처리 단계가 프레임 (그림 2(a))에 적용된다. 전처리 단계는 우선 입력 프레임을 아래와 같이 HSV(Hue, Saturation, Value) 칼라 변환 형식으로 변환한다.

$$H = \begin{cases} \theta & \text{if } B \leq G \\ 360 - \theta & \text{if } B > G \end{cases}$$

$$\theta = \cos^{-1} \left\{ \frac{1}{2} \frac{(R-G) + (R-B)}{\sqrt{(R-G)^2 + (R-B)(G-B)}} \right\}$$

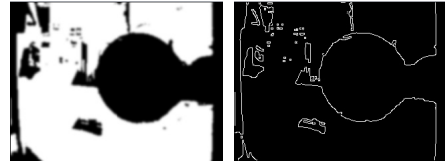
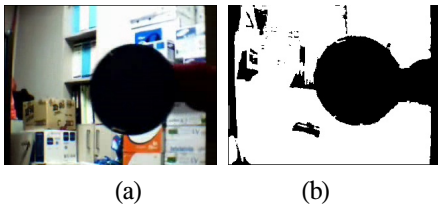
$$S = 1 - \frac{3}{R+G+B} \min(R, G, B)$$

$$V = \frac{1}{3}(R+G+B)$$

프레임을 H, S, V의 각 채널(색상)으로 분리한 후, 프레임에서 의미있는 정보만을 처리하기 위해 분리한 세 채널 중 H채널 이미지의 픽셀값이 임의의 임계값 이하이면 0, 이상이면 1로 변환시킨다. 그림 2(b)는 그림 2(a)를 HSV 칼라 변환으로 변화한 후, 이진화 임계값을 70으로 하여 이진화시킨 결과이다.

이진화 후 프레임에 존재할 수 있는 잡음 또는 손상으로 인한 잘못된 결과를 줄이기 위해 그림 2(c)와 같이 프레임을 평활화한다. 본 논문에서는 많은 평활화 방법 중에 가장 보편적이며 많이 쓰이는 가우시안 필터를 사용했다.

다음은 그림 2(d)와 같이 그레이스케일 이미지의 에지를 찾기 위해 x방향과 y방향으로의 1차 미분을 구하여 네 개의 방향 도함수를 구해 이 방향 도함수의 국지적 최대값에 존재하는 점이



(그림 2) 원영상과 전처리 영상 : (a) 원영상, (b) 이진화 영상, (c) 평활화 영상, (d) 에지 영상

나중에 에지로 판단될 후보 픽셀이 되는 케니 에지 탐색을 수행한다. 케니 에지 탐색을 수행할 때 픽셀의 차가 클 때 단일 임계치 값을 사용하여 에지가 제거되는 것을 막기 위해 이중 임계값(hysteresis threshold)을 사용하여 각각의 에지 후보 픽셀들을 외곽선 형태로 모은다.

3.3 변화 기반(Change-based) 방법

변화 기반 방법은 동영상이 연속된 프레임에서는 변화가 적어 현재 프레임의 GHT 계산 결과가 이전 프레임에서 계산한 GHT의 결과와 거의 같다는 것을 이용한다. 3.1장에서 설명한 이전 프레임과의 차이값을 이용하여 움직임이 임계치 이상인 경우에만 GHT를 실행한다. 움직임이 임계값보다 크면 현재 프레임을 3.2장에 소개한 전처리 과정에 따라, HSV 채널로 변경하고 이진화와 평활화, 에지 검출을 거쳐 GHT를 수행하게 된다.

그림 3(a)와 같이 119번째 프레임에 GHT를 수행하고, 다음 프레임의 움직임이 그림 3(b)와 같이 큰 경우에만 GHT를 수행한다. 즉, 그림 3(b)는 임계값이 2일 때, 135번째 프레임에서 GHT를 수행함을 보여주고 있다. 119번째 프레임과 135번째 프레임 사이에 있는 프레임은 움직임이 작다고 판단하여, 경로 기반 방법을 이용한다.



(a) 119번째 프레임 (b) 135번째 프레임
(그림 3) 변화 기반 방법

3.4 경로 기반(Trajectory-based) 방법

경로 기반의 방법은 프레임에서 에지의 수(n_i)와 탐색할 원의 개수(n_r)가 GHT의 속도에 영향을 미친다는 점을 이용한 방법이다. 에지의 수와 원의 개수를 줄이기 위해, 본 논문에서는 프레임 내 원은 이전 프레임에서 검출한 원의 근처에 존재할 가능성이 높다는 점을 이용하여 후보 원 검출을 수행한다.

먼저, 프레임에서의 에지의 수를 줄이기 위해 아래와 같이 영역 필터를 적용하여 후보 영역 A^t 을 구한다:

$$A^t = C^{t-1}(x,y) \pm (r^{t-1} + c_1)$$

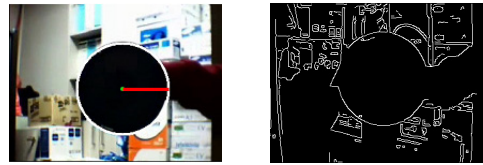
여기서 $C^{t-1}(x,y)$ 는 이전 프레임에서 검출한 원의 중심, r^{t-1} 는 이전 프레임에 검출된 원의 반지름, c_1 는 예상되는 움직임의 정도를 말한다. 본 논문에서는 c_1 의 값으로 5를 사용하였다.

다음으로 탐색할 원의 개수를 줄이기 위해 아래와 같이 크기 필터를 적용하여 후보 원 반지름을 구한다:

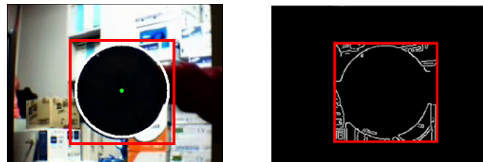
$$\{r^{t-1} - c_2, \dots, r^{t-1}, \dots, r^{t-1} + c_2\}$$

여기서 c_2 는 예상되는 크기의 변화 정도를 말한다. 본 논문에서는 c_2 의 값으로 5를 사용하였다.

그림 4(c)는 이전 영상인 그림 4(a)에서 구한 원의 위치와 크기를 이용하여 GHT를 수행한다. 그림 4(b)는 그림 4(a)의 에지 영상이고, 그림 4(d)는 그림 4(c)의 후보 영역 내 에지 영상으로 그림 4(b)보다 에지의 수가 감소되기 때문에 계산 속도를 향상시킬 수 있다.



(a) 119번째 프레임 (b) 119번째 에지 영상



(c) 120번째 프레임 (d) 120번째 프레임의 후보 영역의 에지 영상

(그림 4) 경로 기반 방법

4. 실험 결과

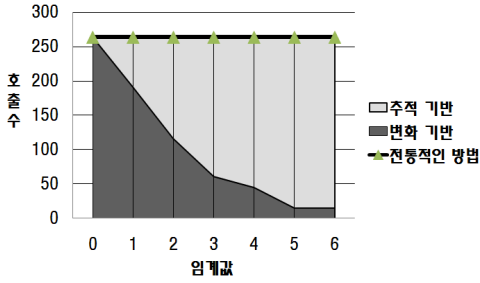
4.1 실험 환경

제안된 알고리즘은 Pentium-4 3.00GHz CPU와 3.00GB 메모리 사양의 Microsoft Windows XP 환경에서 OpenCV언어로 개발되었다. 평활화에는 3×3 의 가우시안 필터가 사용되었고 케니 에지의 임계값은 100과 150이 사용되었다.

본 논문에서는 가장 전통적인 방법인 모든 프레임에 대해 GHT 변환을 실행하는 대신에, 움직임을 활용하여 프레임 사이에 중복을 없애고 움직임이 큰 프레임에 대해서만 GHT를 수행한다. 또한, 프레임 내 움직임이 있는 후보 영역과 탐색할 후보 원의 크기를 추정하여 GHT를 실행하는 방법도 제안하였다.

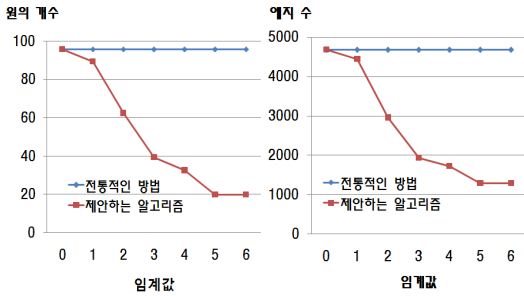
4.2 실험 결과 : 고정 카메라

고정 카메라 실험은 카메라가 고정된 상태에서 하나의 움직이는 원형의 물체를 촬영한 총 264개의 프레임으로 이루어진 동영상을 사용했다.



(그림 5) 임계값에 따른 호출 수

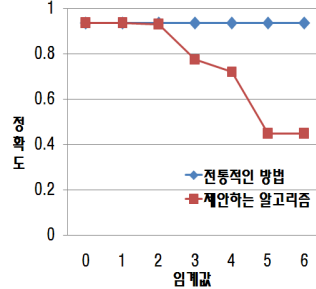
그림 5는 이전 프레임과의 차이값에 대한 임계값에 따른 호출 수를 나타낸 것이다. 전통적인 방법은 임계값에 관계없이 프레임 수 만큼의 GHT를 호출한다. 그러나, 움직임을 이용하는 변화 기반 방법은 임계값이 커질수록 허프 변환의 호출 수가 급격히 줄어드는 것을 볼 수 있다. 이는 임계값이 클수록, 이전 영상과 차이가 크더라도 유사한 영상으로 판단하여 허프 변환의 호출 수가 줄어드는 것이다. 추적 기반 방식은 줄어드는 탐색 영역에서 GHT를 수행함으로써 수행시간을 줄인다.



(그림 6) 임계값에 따른 프레임 당 평균 에지 수 및 원의 개수

그림 6은 변화와 경로 기반한 방법을 이용한 제안하는 알고리즘과 전통적인 방법에서 사용된 프레임 당 평균 에지 수와 탐색된 원의 개수를 보여주고 있다. 이 그림다. 이이전 프레임과의 차이값에 대한 임계값레임두값에 대한제안하는 알고리즘에서 사용된 에지 수와 원의 개수가 전통적인 방법에서 사용한 에지수보다 줄어들었음을 확인할 수 있다. 임계값이 6이상이 되면 경우엔 값

에 변화가 없이 일정하다.



(그림 7) 임계값에 따른 정확도

알고리즘의 검출 성능을 평가하기 위해서 정확도를 계산하였다. 정확도는 검출된 원 중 실제 원일 확률이다:

$$\text{정확도} = \frac{TP}{TP + FP}$$

여기서 TP (true positive)는 동영상의 원 중에서 제안된 알고리즘에 의해 검출된 원의 개수를 나타내고, FP (false positive)는 동영상에서 검출되지 않은 원의 개수를 의미한다.

그림 7은 임계값에 따른 정확도를 나타내고 있는데, 임계값 1,2에 대해서는 전통적인 방법의 정확도와 거의 같게 나오다가 임계값이 높아질수록 정확도가 떨어짐을 보여주고 있다. 임계값이 클수록 이전 영상과 차이가 있더라도 유사하게 보게 되어, 후보 영역의 에지와 원의 크기에 대한 예측 정확도가 낮아 정확한 원 검출이 어렵게 된다.

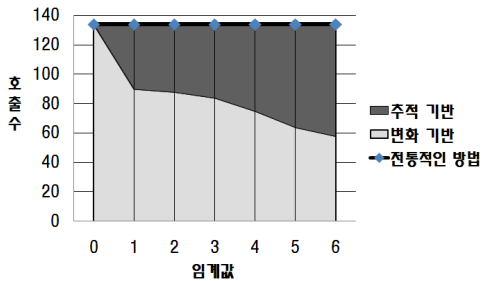
(표 2) 고정카메라에 대한 성능 비교

	SC-GHT [8]	Franke 등 [9]	Pardo 등 [12]	제안하는 알고리즘
속도 향상	약 5.8배	약 1.3배	약 1.5배	약 1.8배
정확도	62%	93.5%	93.5%	93%

다른 알고리즘과의 올바른 비교를 위하여, GHT의 정확도와 같은 약 93%의 관점에서 속도

향상을 비교하였다. 즉, Pardo 등 방법은 그들의 임계값 기준 1일 때, 제안하는 방법은 임계값이 2 일 때를 대상으로 하였다. 또한, GHT에서는 반지름 5에서부터 100까지의 원을 찾으며, 동영상 내 원의 반지름은 42에서 98까지 다양하게 존재한다. 속도 향상은 SC-GHT가 가장 뛰어나지만, 정확도가 다른 세 알고리즘에 비해 떨어진다. Franke 등의 방법은 동영상에 존재하는 모든 원을 찾을 수 있는 반지름 값을 미리 주었으며, 원 크기의 다양성에 따라 속도에 영향을 준다. Pardo 등의 방법은 작은 원이 포함된 경우엔 비교적 높은 속도 향상을 이루나, 반지름이 60 이상인 원이 다수 포함된 경우엔 GHT보다 수행시간이 늘어나기도 한다.

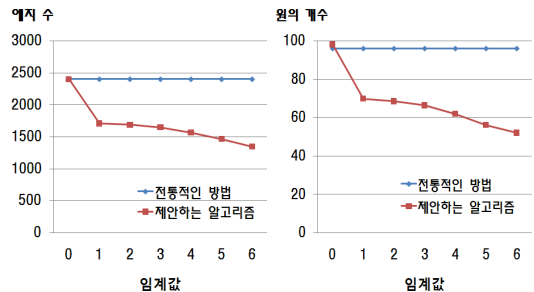
4.3 실험 결과 : 이동 카메라



(그림 8) 임계값에 따른 호출 수

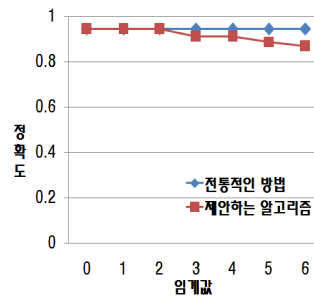
이동 카메라 실험은 두 개의 원형 물체를 등장, 이동, 퇴장, 재등장하도록 움직이는 카메라로 촬영한 총 145개의 프레임으로 이루어진 동영상을 사용했다. 그림 8은 이전 프레임과의 차이값에 대한 임계값에 따른 호출 수를 나타낸 것으로, 변화 기반 방법은 임계값이 커질수록 GHT 변환의 호출 수가 줄어든다. 그러나 고정 카메라 실험에서는 임계값이 커짐에 계속해서 호출수가 급격히 줄어든 반면, 이동 카메라 실험에서는 처음 대폭 줄어든 이후에 약간의 감소만을 보여주고 있다. 이는 상대적으로 고정 카메라의 영상에 비해 이동 카메라의 영상이 큰 움직임들을 주로 포함하

여 임계값에 대부분이 반응하기 때문이다.



(a) 에지 수 (b) 탐색할 원의 개수
(그림 9) 임계값에 따른 프레임 당 평균 에지 수 및 원의 개수

그림 9는 이동 카메라를 이용한 영상에 대해 전통적인 방법과 제안하는 알고리즘에서 사용된 프레임 당 평균 에지 수와 탐색된 원의 개수를 보여주고 있다. 이 그림에서는 제안하는 알고리즘에서 사용된 에지 수와 원의 개수가 전통적인 방법에서 사용된 것보다 줄어들었음을 확인할 수 있다. 임계값이 6 이상이 되면 약간의 감소를 보인 후 일정한 값을 유지하게 된다.



(그림 10) 임계값에 따른 정확도

그림 10은 이동 카메라를 이용했을 때의 임계값에 따른 정확도를 나타낸 것이다. 고정 카메라 실험과 마찬가지로 임계값이 높아질수록 허프 변환의 정확도는 떨어지는 경향이 있지만, 대체로 기존 GHT의 정확도에 근사한 값을 획득하였다.

(표 3) 이동카메라에 대한 성능 비교

	SC-GHT [8]	Franke 등 [9]	Pardo 등 [12]	제안하는 알고리즘
속도 향상	약 6.2배	약 1.0배	약 1.9배	약 1.3배
정확 도	76%	94%	63%	94%

이동 카메라로 촬영한 실험 영상에 대해 다른 알고리즘과의 비교했을 때, SC-GHT가 가장 뛰어난 속도 향상을 보이지만 정확도가 여전히 떨어진다. 원의 반지름의 종류에 영향을 많이 받는 Franke 등의 방법은 이동 카메라가 비슷한 반지름을 가진 원만을 포함하지 않는 한 속도 향상을 기대하기 어렵다. F. Pardo 등의 방법은 검출된 원의 움직임만을 이용해 추적하므로, 원이 등장, 퇴장, 재등장, 큰 이동을 하는 경우엔 정확도가 낮아진다.

5. 결론

본 논문에서는 영상에서 고정 카메라와 이동형 카메라로 실시간으로 원을 찾을 때의 시간 효율적인 GHT 알고리즘을 제안하였다. 제안된 알고리즘 중 변화 기반의 방법은 현재 프레임과 이전 프레임을 비교하여 변화가 큰 프레임만 본래의 GHT를 수행하고, 경로 기반의 방법은 움직임이 작은 프레임에 대해 이전 프레임에서 검출한 원의 정보로 원의 경로를 예측하여 후보 영역과 후보 원들만을 이용하여 GHT를 계산함으로써 수행속도를 줄였다.

동영상에서 원을 찾기 위한 전통적인 방법의 수행 복잡도는 $O(fn_r n_i \frac{n_t}{R})$ 인데, 제안하는 알고리즘은 프레임 수 f 와 영상 내 에지 수 n_i , 탐색할 원의 개수 n_r 을 줄임으로써 속도향상을 이루는 것이다. 실험을 통해 GHT와 비슷한 정확도인 측면에서 고정 카메라 실험에서는 44.45%, 이동 카메라 실험에서는 23.08%까지 수행 속도를 줄임

으로써, 실시간 동영상에서 강건한 원 검출을 위한 속도 향상을 이루었다.

또한, 제안하는 알고리즘은 고정 카메라와 이동 카메라에서 촬영한 동영상에서 원을 검출하는 실험을 수행하였다. 카메라가 고정되어 있을 때는 물론 이동형 카메라에서의 영상 인식 기술에도 제안한 알고리즘이 속도 개선에 효과가 있음을 보여주었다. 이는 제안하는 알고리즘이 카메라 기반 인터페이스에서 실시간으로 빠르고 강건하게 상호작용하는 응용 프로그램에서 중요한 기술로서 신뢰할 수 있는 방법임을 보여주고 있다.

이동 카메라를 이용할 시에는 배경이 계속 변화하기 때문에 물체를 찾을 때의 인식 속도가 매우 중요한 요소가 된다. 이동 카메라를 이용한 실험의 경우 본 논문에서 수행한 원 찾기 뿐만 아니라 다른 다양한 형태의 물체를 찾는 것에도 응용이 가능하다. 또한, 이동 카메라로 보다 다양하게 움직이는 도형을 실시간으로 추출 및 추적하는 연구를 수행하는 것이 가능하다.

참고문헌

- [1] 김종선, 박진배, 주영훈, “하이브리드 실루엣 기반 인간의 강인한 특징 점 추출”, 제어·로봇·시스템학회논문지 제15권, 제4호, pp. 433-438, 2009.
- [2] P. Gupta, H. Mehrotra, A. Rattani, A. Chatterjee and A.K. Kaushik, “Iris recognition using corner detection”, in proc. of international biometric conference, 2006.
- [3] C. Pei, S. Yang, L. Gao and W. Ma, “A real time ball detection framework for soccer video”, in proc. of IWSSIP, pp. 1-4, 2009.
- [4] D. Bradley and G. Roth, “Natural interaction with virtual objects using vision-based six DOF sphere tracking”, in proc. of ACE, pp. 19-26, 2005.
- [5] 김상희, 최형일, “개선된 2D 허프 변환

- (Hough Transform)을 이용한 원 형상 검출 방법”, 한국정보과학회 가을 학술발표논문집, 제2호(A), pp. 233~237, 2008.
- [6] D.H. Ballard, “Generalizing the Hough transform to detect arbitrary shapes”, Pattern Recognition, Vol. 13, No. 2, pp. 111-122, 1981.
- [7] D. Ioannou, W. Huda, A. F. Laine, “Circle recognition through a 2D Hough Transform and radius histogramming”, Image and Vision Computing, Vol. 17, No. 1, pp. 15-26, 1999.
- [8] D. Ma and X. Chen, “Hough transform using slope and curvature as local properties to detect arbitrary 2D shapes,” in proc. of ICPR, pp. 511-513, 1988.
- [9] U. Franke and S. Heinrich, “Fast obstacle detection for urban traffic situations”, IEEE transactions on Intelligent Transportation Systems, Vol. 3, No. 3, pp. 173-181, 2002.
- [10] K. Aizawa, Y. Egi, T. Hamamoto, M. Hatori, M. Abe, H. Maruyama, and H. Otake, “Computational image sensor for on sensor compression”, IEEE transactions on Electron Devices, Vol. 44, No. 10, pp. 1724-1730, 1997.
- [11] T. Hamamoto, R. Ooi, Y. Ohtsuka, and K. Aizawa. “Real-time image processing by using image compression sensor”, in proc. of ICIP, Vol. 3, pp. 935-939, 1999.
- [12] F. Pardo, J.A. Boluda, X. Benavent, J. Domingo, and J.C. Sosa, “Circle detection and tracking speed-up based on change-driven image processing”, in proc. of International Conference on Graphics, Vision and Image Processing, pp. 131-136, 2005.

● 저 자 소개 ●



원혜민

2010년 덕성여자대학교 인터넷정보공학과 졸업(공학사)
 2010년~현재 덕성여자대학교 대학원 전산정보통신학과 석사과정
 관심분야 : 영상처리, 동작인식, HCI etc.
 E-mail : haemini86@duksung.ac.kr



이경미

1993년 덕성여자대학교 전산학과(이학사)
 1996년 연세대학교 전산과학과(이학석사)
 2001년 아이오와 주립대학교 전산학과(이학박사)
 2003년~현재 덕성여자대학교 컴퓨터학과 교수
 관심분야 : 영상처리, 패턴인식, 멀티미디어, HCI
 E-mail : kmlee@duksung.ac.kr