

사각형 마커 검출 및 인식 시스템 개발

강 선 경*, 이 상 설**, 정 성 태**

Development of a Detection and Recognition System for Rectangular Marker

Sun-Kyung Kang*, Sang-Seol Lee**, Sung-Tae Jung**

요 약

본 논문에서는 카메라 영상으로부터 사각형 형태의 마커를 검출하고 인식하는 방법을 제안한다. 본 논문에서는 사각형 형태의 마커 검출을 위하여 입력 영상을 이진 영상으로 변환하고 객체들의 윤곽선을 추출한 다음에 윤곽선을 선분으로 근사화 한다. 근사화된 선분으로부터 기하학적 특징을 이용하여 사각형을 찾는다. 마커의 사각형 영역을 찾은 다음에는 워핑 기법을 이용하여 사각형 마커 영상을 정사각형 형태로 정규화한다. 마커 영상을 정규화한 다음에는 주성분 분석을 통하여 마커 영상으로부터 특징 벡터를 추출하고 표준 마커에 대한 특징 벡터와의 최소 거리법에 의해 마커의 종류를 인식한다. 인식 실험 결과 마커의 종류가 50개일 때에 최대 98%의 인식률을 얻을 수 있었고 입력 영상에 11개의 마커가 있는 경우에 초당 11.1 프레임의 수행 속도를 얻을 수 있었다.

Abstract

In this paper, we present a method for the detection and recognition of rectangular markers from a camera image. It converts the camera image to a binary image and extracts contours of objects in the binary image. After that, it approximates the contours to a list of line segments. It finds rectangular markers by using geometrical features which are extracted from the approximated line segments. It normalizes the shape of extracted markers into exact squares by using the warping technique. It extracts feature vectors from marker image by using principal component analysis. It then calculates the distance between feature vector of input marker image and those of standard markers. Finally, it recognizes the marker by using minimum distance method. Experimental results show that the proposed method achieves 98% recognition rate at maximum for 50 markers and execution speed of 11.1 frames/sec for images which contains eleven markers.

▶ Keyword : Marker Detection(마커 검출), Marker Recognition(마커 인식), Principal Component Analysis(주성분 분석)

• 제1저자 : 강선경

• 접수일 : 2006.08.11, 심사일 : 2006.08.16, 심사완료일 : 2006.09.20

* 원광대학교 컴퓨터공학과 대학원 ** 원광대학교 전기전자 및 정보공학부 교수

※ 본 논문은 정보통신연구진흥원의 정보통신선도기반기술개발사업 과제 A1100-0501-0044에 의해 지원되었음.

1. 서론

의미가 부여된 마커를 카메라 영상으로부터 검출하고 인식하여 증감 현실(augmented reality), 로봇의 경로 이동, 객체 인식 등 다양한 분야에 활용하는 연구가 수행되고 있다. 증감 현실에서는 마커를 검출하고 인식함으로써 자신의 위치를 파악하거나, 사용자의 시선 방향을 인식하거나, 카메라에 보이는 영상에 추가하여 3차원 객체를 디스플레이하거나 텍스트 정보를 출력하는데 활용한다. 사용되는 마커의 종류도 다양하다. 대부분의 경우에 평면에 그려진 사각형 또는 원형 물체를 가진 그림을 이용하는데, 참고 문헌 [1]의 시스템에서는 사람 눈에는 보이지 않고 적외선 카메라로만 감지할 수 있는 마커를 사용함으로써 보다 깔끔한 주변 환경을 제공하는 방법을 제안하였다. 참고 문헌 [2]에서는 발광 다이오드를 사용해서 마커의 내용을 변화시킬 수 있게 함으로써 보다 동적인 시스템을 구축하는 방법을 제안하였다.

그림 1에는 몇몇 시스템에서의 마커의 예가 나타나 있다. 그림 1에서 ARToolkit[3], ARTag[4], SCR[5], HOM[6]은 증감 현실에 사용되는 마커 시스템이다. ARTag, SCR, HOM의 마커는 2차원 바코드처럼 격자 형태의 구조에 데이터를 코드화한 형태를 취하고 있으며 인식의 견고성을 더 높일 수 있고 많은 수의 마커를 쉽게 생성할 수 있는 장점을 가진다. 그림에 나타나 있는 ARTag의 마커는 숫자 1을 나타내는데, 사람이 봐서는 어떤 의미를 가진 마커인지 인지하기는 거의 불가능하다.

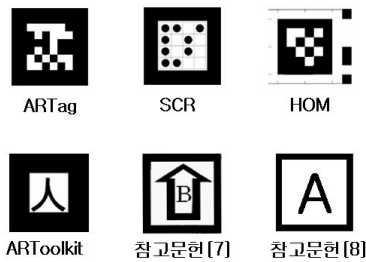


그림 1. 기존 시스템의 마커 예
Fig. 1. Marker examples previous marker system

ARToolkit, 참고문헌 [7, 8] 등에서는 알파벳, 숫자, 기호 등 특정한 구조를 가지지 않는 마커를 사용하였다. ARToolkit는 착용 가능한 컴퓨터를 입은 사용자가 낚신 건

물에서 마커를 인식함으로써 자신의 위치를 파악하고 원하는 장소에 찾아가기 위한 정보를 얻는 등의 증감 현실을 위한 여러 가지 용도로 활용되었다[7,9,10,11]. 참고 문헌 5에서는 원격 교육을 위한 촬영에서 마커를 추적함으로써 카메라를 자동으로 조정하는 용도로 활용하였다. 참고 문헌 6에서는 로봇이 마커를 인식함으로써 자신의 위치를 파악하는데 활용하였다. 격자 구조를 갖는 마커를 격자형 마커라고 하고 그렇지 않은 마커를 일반형 마커라고 하자.

본 논문에서는 <그림 2>와 같은 일반형 마커를 검출하고 인식하는 방법을 제안한다. 마커의 가장자리는 정사각형이고 내부에 단순한 형태의 도형을 포함한다. 배경은 흰색이고 마커의 색은 검정색이다.

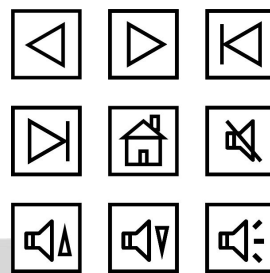


그림 2. 제안된 마커 예
Fig. 2. Proposed marker examples

본 논문에서 사용한 마커들에는 보다 사람이 인지 가능한 마커들이 포함되어 있다. 이러한 마커는 사용자 인터페이스에 활용될 수 있을 것이다. 예를 들어, 사용자가 착용 가능한 컴퓨터를 입고 있고 또한 카메라가 달린 HMD(Head Mounted Display)를 착용하고 있다고 하자. 그리고 사용자의 주변 환경에 있는 객체들은 어떤 제어 장치에 의해 동작이 원격으로 조절될 수 있다고 하자. 사용자가 HMD에 디스플레이 되는 주변 환경의 영상에서 어떤 객체를 동작시키거나 한다면 그 객체를 인식하고 제어장치에 원하는 장치와 동작을 요구해야 할 것이다. 이때에 영상으로부터 직접 객체를 인식할 수도 있지만 마커를 객체 주위에 붙여 놓고 마커를 인식하면 보다 용이할 것이다. 또한 그 객체가 수행할 수 있는 동작도 마커로 형상화해서 붙여놓고 그 마커를 인식하면 해당 동작을 간단하게 제어 장치에게 요구할 수 있을 것이다. 예를 들어 <그림 2>의 두 번째 줄 맨 오른쪽 마커는 스피커의 소리를 나지 않게 하는 것을 나타내며, 세 번째 줄의 첫 번째 마커는 볼륨을 높이고, 두 번째는 볼륨을 내리고, 세 번째는 스피커 소리가 나게 하는 동작을 나타낸다. 이러한 마커가 스피커 옆에 있을 때에 마커를 인식

함으로써 스피커에 대한 원하는 동작을 제어 장치에 요구할 수 있을 것이다.

본 논문의 방법에서는 사각형 형태의 마커 검출을 위하여 입력 영상을 이진 영상으로 변환하고 객체들의 윤곽선을 추출한 다음에 윤곽선을 선분으로 근사화한다. 주변 환경의 밝기에 따라 영상의 명암이 크게 달라지는데, 본 논문에서는 여러 임계값을 사용하여 이진 영상을 구함으로써 조명 변화에 영향을 받지 않도록 했다. 근사화된 선분으로부터 기하학적 특징을 이용하여 사각형을 찾는다. 마커의 사각형 영역을 찾은 다음에는 위핑 기법을 이용하여 사각형 마커 영상을 정사각형 형태로 정규화한다. 마커 영상을 정규화한 다음에는 주성분 분석을 통하여 마커 영상으로부터 특징 벡터를 추출하고 표준 마커에 대한 특징 벡터와의 최소 거리 법에 의해 마커의 종류를 인식한다. 인식 실험 결과 마커의 종류가 50개일 때에 최대 98%의 인식률을 얻을 수 있었고 입력 영상에 11개의 마커가 있는 경우에 초당 11.1 프레임의 수행 속도를 얻을 수 있었다.

일반형 마커와 격자형 마커는 인식 방법에 있어서 차이가 있으므로 본 논문의 방법은 일반형 마커 인식 방법하고 비교될 수 있다. 즉, 격자형 마커에서는 마커 영상을 분석하여 어떤 코드를 나타내는지 해독함으로써 마커를 인식하지만 일반형 마커에서는 미리 정의된 마커 부류에 대하여 입력 영상의 마커를 비교하여 인식하는 방법을 사용한다. ARToolkit에서는 이진화 영상에 대해 마커 영역을 검출한 다음에 마커 영상에 대한 기하학적 불변량(geometrical invariants)을 구하여 미리 정의된 값들과 비교하여 마커를 인식한다. 기하학적 불변량이란 어떤 영상에 회전, 확대, 축소, 이동 등의 변환을 적용하더라도 계속해서 변하지 않는 값들을 의미한다. ARToolkit는 기하학적 불변량을 사용함으로써 수행속도가 아주 빠르지만 마커를 잘못 인식하는 경우가 많다는 문제를 가지고 있다. 참고문헌 [7]의 마커 인식은 ARToolkit의 방법을 사용한 것이다. 참고 문헌 [8]에서는 마커 인식에 인공 신경망을 이용하였고 97%의 인식률을 얻었다고 되어 있다. 본 논문의 방법은 ARToolkit 방법 보다는 보다 조명 변화에 견고하고 인식률도 앞선다. 참고 문헌 [8]의 방법에 비해서는 보다 간단한 인식 방법을 사용하면서도 인식률은 비슷한 결과를 얻었다고 할 수 있다. 본 논문의 2장에서는 마커 검출 방법에 대해서 설명하고 2장에서는 마커 인식 방법을 설명하며 4장에서는 실험 결과를 설명하고 5장에서 결론을 맺는다.

II. 마커 검출

본 논문에서 사용하는 마커의 원래 형태는 정사각형이지만 마커가 있는 평면과 카메라의 상대적인 위치에 따라서 정사각형 형태로 보이지 않을 수 있다. <그림 3>에는 마커를 비스듬한 방향에서 촬영된 모습이 나타나 있다. 이 영상에서 마커는 각 변의 길이도 다르고 네 정점의 사이 각이 90도가 아닌 일반 사각형 형태이다.



그림 3. 마커 촬영 예

Fig 3. Captured image of a sample marker

2.1 사각형 검출

본 논문의 사각형 마커 검출 과정이 <그림 4>에 나타나 있다.

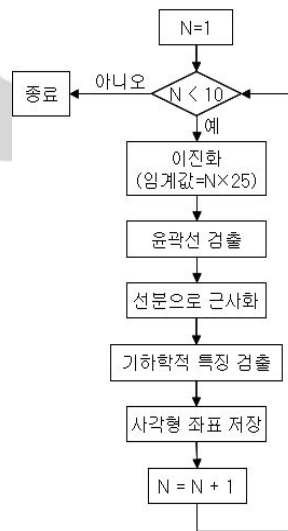


그림 4. 마커 검출 단계

Fig 4. The procedure of marker detection

본 논문에서는 사각형 마커를 검출하기 위해 먼저 입력 영상을 이진 영상으로 변환하고 윤곽선을 검출한다. 여기에서

검출된 윤곽선은 연결된 픽셀들로 구성되어 있는데, 사각형 검출을 위해 윤곽선을 선분으로 근사화한다. 선분으로 근사화한 다음에는 윤곽선을 이루는 선분의 수, 윤곽선 내부의 면적, 블록 객체 여부 등의 기하학적 특징을 검출하여 이로부터 윤곽선이 사각형인지 아닌지를 판단한다.

본 논문에서는 처리 방법을 단순하게 하기 위해 컬러 영상인 카메라의 입력 영상을 그대로 사용하지 않고 배경색은 255이고 마커 색은 0인 이진 영상으로 변환해서 사용한다. 컬러 입력 영상을 이진 영상으로 변환하기 위해 먼저 256가지의 명암도를 갖는 흑백 영상으로 변환한다. 흑백 영상을 이진 영상으로 변환하기 위해서는 임계값을 정하여 임계값보다 크면 배경색으로 변환하고 그렇지 않으면 마커 색으로 변환하는 방법을 사용하면 된다. 사각형 마커는 흰색 배경에 검정색으로 되어있지만 카메라로부터 촬영된 입력 영상에서는 주변 환경의 빛의 밝기에 따라 배경과 마커의 명암도가 많은 영향을 받으므로, 이진화를 위한 적절한 임계값을 정하는 문제는 그리 간단한 문제가 아니다. 본 논문에서는 흑백 영상이 가질 수 있는 명암도 값인 0에서 255 사이에 일정 간격으로 9개의 임계값을 지정하여 이진 영상 9개를 생성하고 각 이진 영상으로부터 사각형을 검출하는 방법을 사용한다. <그림 5>에는 그림 3에 대한 이진화 영상 예가 나타나 있다. 실험 결과에 의하면 대부분의 경우에 사각형 마커가 검출되는 이진 영상은 3개 내지 4개 정도이었다.



(a) 임계값 = 100 (b) 임계값 = 125
그림 5. 이진화 영상 예

Fig 5. Binary images of the input image

이진화를 수행한 다음에는 윤곽선 검출을 수행한다. 윤곽선이란 객체의 가장자리 선을 말하는데, <그림 6> (a)와 같이 객체 내부에 비어있는 공간이 없는 경우에는 바깥쪽 윤곽선만 존재하고 <그림 6> (b)의 사각형 마커의 외곽선처럼 객체 내부에 비어있는 부분이 있으면 객체의 바깥쪽 윤곽선과 안쪽 윤곽선이 존재한다.

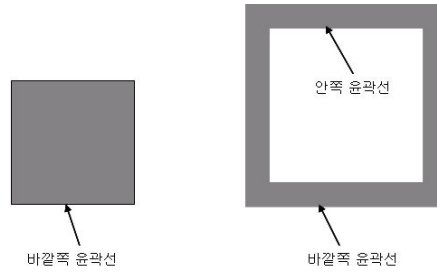


그림 6. 바깥쪽 윤곽선과 안쪽 윤곽선
Fig 6. Outer contour and inner contour

본 논문에서는 객체의 가장자리에 있는 한 픽셀로부터 시작하여 가장자리에 있는 픽셀을 계속 따라가다가 시작점에 도착하면 추적을 멈추는 방법[12]을 사용하였다. 윤곽선 검출의 시작점은 영상의 맨 윗줄의 좌측에 있는 픽셀부터 시작하여 오른쪽방향으로 스캔하면서 다음 조건을 만족하는 픽셀 $P(x, y)$ 를 선택한다.

$$P(x, y) = 0 \text{ 그리고 } (P(x, y-1) = 255 \text{ 또는 } y = 0)$$

윤곽선 검출의 시작점으로부터 물체의 가장자리 픽셀을 한 방향으로 따라가다가 원래 자리에 돌아오면 탐색이 종료된다. 픽셀 $P(x, y)$ 가 가장자리 픽셀일 경우에 인접한 가장자리 픽셀은 픽셀 $P(x, y)$ 주위의 8개 픽셀 중에 존재하는데, 이를 찾는 방법은 다음과 같다.

- 1) 인접 픽셀중의 한 픽셀에서 검사를 시작한다.
- 2) 픽셀값이 배경색이면 계속해서 다음 픽셀을 검사한다. 픽셀값이 물체색이면 그 픽셀을 인접한 가장자리 픽셀로 정하고 검사를 멈춘다.

다시 말하면, 인접한 8개의 픽셀을 순서대로 검사하면서 물체색을 가지는 첫 번째 픽셀을 만나면 그 픽셀을 인접한 가장자리 픽셀로 정하는 것이다. 가장자리 픽셀 $P(x, y)$ 주위의 8개 픽셀에 그림 7과 같이 시계 방향으로 번호를 붙이기로 하자.

7	0	1
6	$P(x, y)$	2
5	4	3

그림 7. 가장자리 픽셀 $P(x,y)$ 의 인접 픽셀 번호
Fig 7. Adjacent pixels of a boundary pixel

인접 픽셀 검사의 시작 위치는 현재까지 가장자리 픽셀을 탐색해온 방향에 따라 다르게 결정된다. 먼저 윤곽선 검출의 시작점에서는 1번 픽셀부터 검사를 시작하면 된다. 그리고 윤곽선을 따라가는 도중에는 현재의 탐색에서 가장자리 픽셀로 선택된 픽셀 번호와 그에 따른 다음의 탐색에서 인접 픽셀의 검사를 시작할 픽셀 번호가 <표 1>에 나타나 있다.

표 1. 가장자리 픽셀의 위치와 다음 검사 시작 위치
Table 1. The location of a boundary pixel and the start location of the next probe

가장자리 픽셀 발견 위치	다음 검사 시작 위치
0	7
1	7
2	1
3	1
4	3
5	3
6	5
7	5

가장자리 픽셀을 탐색해 나가다가 더 이상 연결된 가장자리 픽셀이 없는 경우 또는 탐색의 시작점을 만나는 경우에는 탐색을 종료한다.

윤곽선 검출에 의해 구한 물체의 가장자리 점들을 선분으로 근사화한다. 윤곽선을 구성하는 픽셀들을 선분으로 근사화하는 방법으로 본 논문에서는 Douglas-Peucker 알고리즘 [13]이라 불리는 방법을 사용한다. 이 방법은 다음과 같은 단계로 구성된다.

- 1) 윤곽선의 점들 중에서 가장 먼 거리에 있는 두 점을 발견
- 2) 가장 먼 거리에 있는 두 점을 기점으로 윤곽선을 두 개의 조각으로 분할하여 스택에 저장
- 3) 스택이 비워질 때까지 다음 과정을 반복
 - i) 스택으로부터 윤곽선 조각 하나를 가져온 다음에 그 길이를 검사
 - ii) 윤곽선 조각을 구성하는 점의 수가 2개이면 더 이상 그 윤곽선 조각은 선분으로 출력
 - iii) 윤곽선 조각을 구성하는 점의 수가 2개 보다 많은 경우에는 그 조각의 점들 중에서 그 조각의 양 끝점 A와 B를 잇는 선분 AB로부터 가장 먼 거리에 있는 점 C를 발견

- iv) 선분 AB와 점 C 사이의 거리가 임계값 이하이면 A와 B 사이의 윤곽선을 선분 AB로 근사화하고 그렇지 않으면 점 C를 기점으로 윤곽선을 2개로 분할하여 스택에 저장

선분으로 근사화한 다음에는 사각형인지 아닌지를 판단하기 위하여 기하학적 특징을 이용한다. 기하학적 특징으로는 윤곽선 내부의 면적, 각 선분 사이의 각도, 블록 객체 여부, 윤곽선을 구성하는 선분의 수, 선분의 길이를 이용한다. 윤곽선 내부 면적은 다각형 면적을 계산하는 공식을 이용하면 간단하게 구할 수 있다. N-각형의 각 정점의 좌표가 연결되어 있는 순서대로 $(x_0, y_0), (x_1, y_1), (x_2, y_2), \dots, (x_{N-1}, y_{N-1})$ 이라 할 때 면적은 다음의 식으로 구할 수 있다. 여기에서 $x_N = x_0, y_N = y_0$ 라고 가정한다.

$$A = \frac{1}{2} \sum_{i=0}^{N-1} (x_i y_{i+1} - x_{i+1} y_i)$$

윤곽선의 이웃한 선분 사이의 각도는 두 선분을 벡터로 표현한 다음에 벡터 사이의 각도 계산 방법을 활용하여 구한다. 사이각을 구하고자 하는 두 선분이 <그림 8>과 같다고 하자.

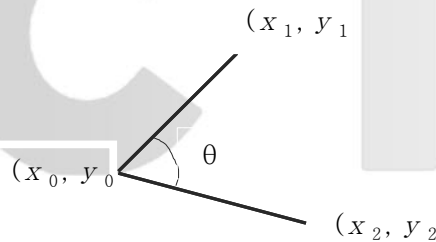


그림 8. 각을 구하고자 하는 두 선분
Fig 8. Two line segments and the included angle

두 선분이 그림 8과 같이 주어질 때 두 선분 사이의 각도 θ 를 구하는 식은 다음과 같다.

$$\theta = \cos^{-1} \left(\frac{(x_1 - x_0)(x_2 - x_0) + (y_1 - y_0)(y_2 - y_0)}{\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} \sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2}} \right)$$

블록 객체란 객체 내부의 임의의 두 점을 연결하는 선을 그었을 경우에 객체 외부로 선이 나가지 않는 객체를 의미하는데 <그림 9> (a)의 객체는 블록 객체이고 <그림 9> (b)의 객체는 오목 객체이다.

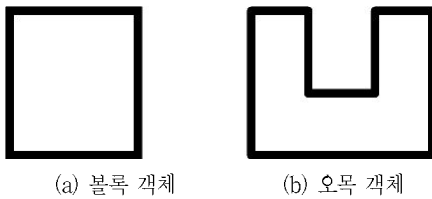


그림 9. 볼록 객체와 오목 객체에
Fig 9. Convex object and concave object

다각형이 볼록한지는 다각형을 구성하는 모든 이웃한 두 선분에 대하여 다음 부등식의 값이 일정한지 검사하면 된다. 이웃한 두 선분의 세 정점이 (x_{i-1}, y_{i-1}) , (x_i, y_i) , (x_{i+1}, y_{i+1}) 라고 하자. 볼록 객체의 경우에 아래의 부등식이 일정한 값을 갖게 되는 이유는, 볼록 객체의 경우에 정점의 진행 방향에 규칙성이 있기 때문이다. 즉, 볼록 객체의 정점을 시계 방향으로 따라갈 때에 이웃한 두 변에서 x 좌표 값이 증가하기 시작하면 y 좌표 값이 작아지기 시작하고 x 좌표 값이 감소하기 시작하면 y 좌표 값은 커지기 시작한다. <그림 9> (b)의 오목 다각형의 경우에는 x 좌표 값이 증가하기 시작한 다음에 y 좌표 값이 작아졌다 커지는 것을 볼 수 있다.

$$(y_{i+1} - y_i)(x_i - x_{i-1}) \geq (x_{i+1} - x_i)(y_i - y_{i-1})$$

본 논문에서는 선분으로 근사화된 윤곽선이 사각형 마커의 윤곽선인지를 판별하는 조건으로 윤곽선을 구성하는 선분의 수가 4개이고 윤곽선 내부 면적이 100이상이고 이웃한 선분 사이각이 45도에서 135도 사이이며 볼록 다각형이어야 하도록 했다. 윤곽선 내부 면적을 검사하는 것은 마커를 인식하기 위해 최소한 마커 영상의 크기가 가로 세로 각각 10 픽셀씩은 되어야 할 것으로 판단했기 때문이다. <그림 10>에는 그림 3의 영상에 대한 윤곽선 검출 결과가 나타나 있다.



그림 10. 마커 검출 결과
Fig 10. Marker detection result

2.2 마커 영상 추출

입력 영상의 사각형 마커를 인식하기 위해서는 검출된 사각형 마커의 영상을 추출한 다음에 정사각형 형태의 표준 마커 영상 형태로 변환을 해야 한다. <그림 10>의 영상에 대하여 사각형 마커에 해당하는 영역의 영상만 추출한 결과가 <그림 11>에 나타나 있다.



그림 11. 추출된 마커 영상
Fig 11. Extracted marker image

<그림 11>의 마커 영상과 같이 추출된 마커 영상이 정사각형이 아닌 경우에는 이를 정사각형 형태로 변환을 해주어야 한다. 이를 위하여 본 논문에서는 먼저 워핑(warping)기법 [14]을 이용하여 마커 영상을 <그림 12>와 같이 직사각형 형태로 변환한다.



그림 12. 직사각형 형태로 워핑된 결과 영상
Fig 12. Marker image warped into a square

직사각형 형태로 워핑을 수행한 다음에는 확대 또는 축소 연산을 이용하여 표준 크기의 정사각형 형태로 변환한다. <그림 13>에는 <그림 12>의 영상을 정사각형 형태로 변환한 결과가 나타나 있다.



그림 13. 정사각형 형태로 변환된 마커 영상
Fig 13. Marker image warped into an exact square

정사각형 형태로 변환한 다음에는 이진화를 수행하는데, 이진화에 사용하는 임계값은 사각형 검출시에 적용되었던 임계값과 같은 값을 사용하면 된다. 그런데, 여러 가지 임계값에 대하여 이진화를 수행하고 마커를 검출하였으므로 같은 마커가 여러 번 검출될 수 있다. 같은 마커들이 여러 번 검출된 경우에는 그중에서 임계값이 중간인 경우에 검출된 마커를 선택하여 사용한다. <그림 3>의 영상에서는 임계값이 50, 75, 100, 125인 경우에 마커가 검출되었는데, 이 중에서 임계값이 75를 사용하여 이진화를 수행한다. 이진화 결과는 <그림 14>에 나타나 있다.



그림 14. 이진화된 마커 영상
Fig 14. Binary marker image

III. 마커 인식

본 논문에서는 마커 인식을 위해 최소 거리 식별법을 사용한다. 패턴 인식에서 인식 대상이 되는 각 부류를 대표하는 패턴을 표준 패턴이라 하는데, 표준 패턴은 각 부류마다 하나 또는 여러 개 존재 할 수 있다. M개의 부류

W_1, W_2, \dots, W_M 가 있고 부류 W_i 를 대표하는 표준 패턴을 y_i 라 하고 입력 패턴을 x 로 나타내기로 하자.

각 패턴을 N차원의 특징 벡터로 나타내면 $y_i = (y_{i1}, y_{i2}, \dots, y_{iN})$, $x = (x_1, x_2, \dots, x_N)$ 이 될 것이다. 그리고 패턴 x 와 패턴 y_i 의 거리

$D(x, y_i)$ 는 다음 식으로 계산할 수 있다. 최소 거리 식별법에서는 입력 패턴 x 를 $D(x, y_i)$ 의 값이 최소가 되는 부류 W_i 로 인식한다.

$$D(x, y_i) = \sqrt{\sum_{k=1}^N (x_k - y_{ik})^2}$$

앞 절에서 설명한 방법에 의하여 입력 영상으로부터 구한 정사각형 마커 영상의 크기는 64 x 64이다. 영상의 픽셀

값을 그대로 특징 벡터로 사용하면 특징 벡터의 차원은 4096이 되고 최소거리 식별법에서 많은 양의 연산을 필요로 하게 된다. 따라서 인식의 수행 속도를 높이기 위해 특징 벡터의 차원을 줄이는 방법이 널리 사용되는데, 본 논문에서는 주성분 분석을 이용하여 특징 벡터의 차원을 줄인다. 주성분 분석은 고차원 특징 벡터를 몇 개의 주성분 값으로 나타내어 저차원의 특징 벡터로 표현해 주는 방식이다. 특징 벡터가 N-차원일 경우에는 각 특징 벡터는 N개의 기준 축을 기준으로 표현된다. 따라서 특징 벡터의 차원을 줄인다는 것은 기준 축을 줄이는 것으로 생각할 수 있다. 예를 들어 X축과 Y축으로 구성된 2차원 평면에서 특징 벡터가 <그림 11.10>(a)와 같이 대각선 방향으로 길게 퍼져있다고 하자. 여기에서는 각 특징 벡터를 하나의 점으로 표시하였다. <그림 11.10>(a)와 같은 분포에서 대각선 방향과 그에 수직 방향인 X' 축과 Y' 축 관점에서 특징 벡터를 살펴 볼 때에는 특징 벡터가 X'축을 중심으로 넓게 퍼져 있고 Y' 축을 중심으로 좁게 모여 있다. 따라서 특징 벡터를 구분할 때에 X' 축 상에서 어느 위치에 있느냐가 가장 중요한 요소가 되고 Y' 축 상에서는 어느 위치에 있느냐 하는 것이 그 다음으로 중요하게 된다. 이와 같이 특징 벡터의 분포에 따른 새로운 기준 축을 구한 다음에는 기준 축 중에서 중요한 것만 남기고 나머지는 버림으로써 데이터의 차원을 줄일 수 있다. <그림 11.10>(a)의 경우에 X' 축이 더 중요하므로, <그림 11.10>(b)에 나타나 있듯이 특징 벡터를 X' 축에 투영하여 Y' 축 성분 값은 버리고 X' 축 성분 값만 남도록 함으로써 2차원의 특징 벡터를 1차원의 특징 벡터로 줄여서 표현할 수 있다.

주성분 분석을 위해서는 마커 영상들이 필요하다. 그림 15에는 주성분 분석에 사용된 마커 영상의 일부가 나타나 있다. 이들은 앞 절에서 설명한 마커 검출 방법에 의하여 카메라 영상으로부터 추출된 영상이다.

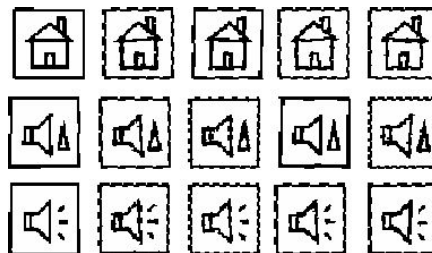


그림 15. 주성분 분석에 사용된 마커 영상 예
Fig 15. Images for principal component analysis

주성분 분석을 위해서는 먼저 각 영상을 열벡터인 하나의 특징 벡터로 나타낸다. 그리고 전체 영상에 대한 특징 벡터들을 모아 하나의 행렬 S 를 생성한다. 그리고 행렬 S 에 대한 공분산 행렬을 구한 다음에 공분산 행렬에 대한 고유 벡터[15]와 고유 값[15]을 구한다. 고유 벡터들이 주성분 축이 되며 고유 값이 클수록 해당 주성분 축의 중요도가 높아진다. 한 영상의 픽셀수가 N 이고 주성분 분석에 사용된 영상의 수가 M 개이면, 생성된 행렬 S 의 크기는 $N \times M$ 이 된다. Φ_i 를 i 번째 영상에 대한 특징 벡터라 하고 영상의 수가 M 개일 때 행렬 S 는 다음과 같이 나타낼 수 있다.

$$S = [\Phi_1 \ \Phi_2 \ \dots \ \Phi_M]$$

행렬 S 에 대한 공분산 행렬을 구하기 위해 먼저 평균 벡터 Ψ 를 다음 식과 같이 구한다.

$$\Psi = \frac{1}{M} \left[\sum_{i=1}^M \Phi_i \right]$$

평균 벡터 Ψ 를 구한 다음에는 행렬 S 의 각 열벡터로부터 평균 벡터를 빼줌으로써 행렬 A 를 생성한다.

$$A = [\Phi_1 - \Psi \ \Phi_2 - \Psi \ \dots \ \Phi_M - \Psi]$$

행렬 S 의 공분산 행렬은 다음과 같이 구할 수 있다. 행렬 A 의 크기는 $N \times M$ 이고 행렬 A^T 의 크기는 $M \times N$ 이므로 공분산 행렬 C 의 크기는 $N \times N$ 이 된다.

$$C = AA^T$$

위와 같이 공분산 행렬을 구한 다음에 고유 벡터와 고유 값을 구함으로써 주성분 축을 구할 수 있다. 그런데, 마커 영상의 경우와 같이 N 의 값이 아주 큰 경우에는 공분산 행렬에 대한 고유 벡터를 구하는 것은 고성능 PC의 경우에도 여러 날 이상이 소요된다. 고유 벡터를 빠르게 구하기 위한 방법으로는 다음과 같이 행렬 L 을 생성한 다음에 L 에 대한 고유 벡터를 구하고 행렬 A 를 고유 벡터에 투영하는 방법이 있다. 행렬 A^T 의 크기는 $M \times N$ 이고 행렬 A 의 크기는 $N \times M$ 이므로 행렬 L 의 크기는 $M \times M$ 이 된다.

$$L = A^T A$$

행렬 L 에 대해 구한 i 번째 고유 벡터를 \mathbf{x}_i 라 하고 고유 값을 λ_i 라 하자. 고유 벡터를 구한 다음에는 다음과 같이 고유 벡터를 열로 하는 행렬을 생성한다. 고유 벡터들은 고유 값에 따라 정렬하여 행렬의 좌측으로부터 우측으로 고유 값이 점점 작아지도록 하자.

$$E = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_M]$$

행렬 L 에 대한 고유 벡터로부터 행렬 C 에 대한 고유 벡터를 다음 식과 같이 구할 수 있다.

$$V = AE$$

행렬 V 를 구한 다음에는 좌측의 열부터 원하는 수만큼 주성분 축으로 사용하면 된다. 주성분 축을 구한 다음에는 주성분 축에 입력 영상을 투영함으로써 입력 영상들에 대해서 차원이 축소된 특징 벡터를 구할 수 있다.

마커 인식을 위해서는 먼저 주성분 분석에 사용된 마커 영상 $\Phi_1, \Phi_2, \dots, \Phi_M$ 에 대해서 특징 벡터 Z_1, Z_2, \dots, Z_M 를 구한다. 그리고 인식할 입력 영상이 들어오면 그에 대한 특징 벡터를 구하고 이 벡터와 Z_1, Z_2, \dots, Z_M 벡터 사이의 거리를 구해서 가장 가까운 거리에 있는 벡터를 찾으면 된다.

영상 Φ 를 주성분 축에 투영하여 특징 벡터 Z_i 를 구하기 위해서는 다음 식과 같이 먼저 영상 Φ 로부터 주성분 분석에 사용된 평균 얼굴 벡터 Ψ 를 뺀 다음에 고유 벡터 행렬에 대한 전치 행렬과 곱하면 된다. 행렬 V 로부터 K 개의 열만 선택하고 나머지는 버림으로써 V 의 크기가 $N \times K$ 가 되었을 경우 행렬 V^T 의 크기는 $K \times N$ 이고 행렬 I 의 크기는 $N \times 1$ 이므로 행렬 Z_i 의 크기는 $K \times 1$ 이 된다. 이렇게 함으로써 특징 벡터의 크기를 N 에서 K 로 줄어든게 되는 것이다. 인식 대상 영상에 대해서도 아래의 식을 동일하게 적용하여 특징 벡터를 구할 수 있다.

$$I = \Phi_i - \Psi$$

$$Z_i = V^T I$$

IV. 실험 결과

본 논문에서는 두 가지로 나누어 실험을 진행하였다. 하나는 인식 성능을 측정하기 위하여 디지털 카메라로 촬영된 영상에 대해 인식 실험을 수행하였고 다른 하나는 인식 속도를 측정하기 위해 PC 카메라에서 실시간으로 캡처된 영상에 대한 인식 실험을 수행하였다. 실험에 사용된 전체 마커의 종류는 모두 50개이고 그림 16에 나타나 있다.

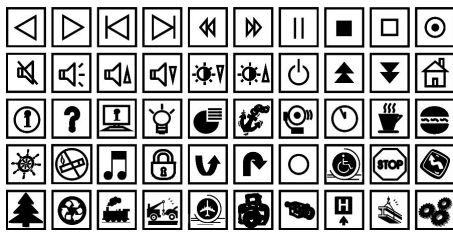


그림 16. 마커의 종류
Fig 16. Marker classes

디지털 카메라 영상에 이용한 실험에서는 마커가 10 종류, 20 종류, 30 종류, 40 종류, 50 종류일 때로 구분해서 실험함으로써 마커 수의 증가에 따른 인식률의 변화를 살펴본다. 또한 주성분 분석 결과로 구한 특징 벡터의 크기도 변화시키면서 인식률의 변화를 살펴보았다. 각 마커에 대하여 10 장의 영상을 촬영하였으며 5장의 영상은 주성분 분석에 사용하였고 각 마커에 대한 표준 패턴으로 사용하였다. 나머지 5장의 영상은 인식실험에 사용하였다. 마커의 촬영은 <그림 17>과 같이 거리와 카메라 방향등을 변화시켜 가면서 촬영하였다. 50개의 마커에 대해 10장씩 모두 500장의 영상으로부터 마커를 검출한 결과 모든 영상에서 마커 영상을 올바르게 검출할 수 있었다.

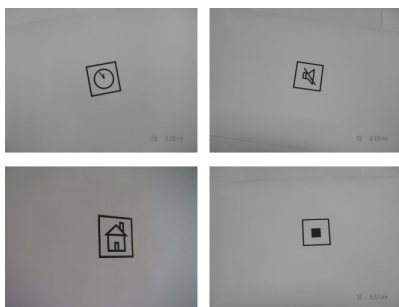


그림 17. 마커 촬영 영상
Fig 17. Captured marker images

<표 2>에는 인식 실험 결과가 나타나 있다. 표에서 마커이 수가 10개일 때는 특징 벡터의 크기가 50까지만 나타나 있다. 이는 특징 벡터의 최대 크기는 마커의 수에 5를 곱한 값과 같기 때문이다. 마커가 10개일 때는 특징 벡터의 최대 크기가 50이고 마커가 50개일 때에는 특징 벡터 최대 크기가 250이다. 표에 나타나 있듯이 대체적으로 특징 벡터의 크기가 증가하면 인식률도 같이 증가하다가 어느 정도 증가하면 그 수준을 유지하거나 더 낮아지는 것을 볼 수 있다. 특징 벡터의 크기가 대략 마커의 수에 20 또는 30을 더한 수와 같아질 때에 인식률이 최고가 되는 것을 볼 수 있다. 인식률은 마커의 수가 10개에서 20개로 되면서 증가하였고 30에서 다시 감소하였다가 40, 50이 되면 다시 증가하였다. 이러한 결과가 발생한 이유는 맨 처음 사용한 10개의 마커 안에 서로 비슷한 마커가 있었고 이들이 서로 혼동되어 잘못 인식되었기 때문이다. 마커의 수가 20개로 늘어나도 추가된 마커들에서는 오류가 없었고 10개일 경우에 틀렸던 것들이 계속해서 틀린 것이다. 따라서 틀린 개수는 같으면서 마커의 수가 늘어나 인식률이 높아진 것이다. 마찬가지로 30개가 되면서 비슷한 마커들이 추가되어서 인식률이 떨어진 것이다. 이로부터 마커의 수가 증가하더라도 인식률에는 큰 영향이 없다는 것을 알 수 있었고 비슷한 마커가 얼마나 있는 지가 인식률에 더 큰 영향을 미치는 것을 알 수 있었다.

표 2 인식 결과
Table 2. Recognition results

마커 수	영상 수	특징 벡터의 크기에 따른 인식률									
		10	20	30	40	50	60	70	80	90	
10	50	86	98	98	98	98					
20	100	91	97	98	99	99	98	98	98	98	
30	150	93.3	92.7	95.3	95.3	96.7	96	96	96	95.3	
40	200	92.5	93.5	95.5	96	96.5	97.5	97.5	97	97	
50	250	93.2	96	96	95.6	96.4	97.6	97.6	98	97.6	

특징 벡터로부터 영상을 복원할 수도 있는데, 마커의 수가 20개인 경우에 여러 가지 특징 벡터의 크기에 대하여 복원된 영상과 원래의 마커 영상이 <그림 18>에 나타나 있다. 그림에서 나타나 있듯이 특징 벡터의 크기가 작을수록 복원된 영상의 질도 상대적으로 떨어지는데, 이로부터 특징 벡터의 크기가 충분하지 않으면 영상을 충실하게 표현하지 못하게 되고 그에 따라 인식률도 떨어지게 되는 것을 알 수 있다.

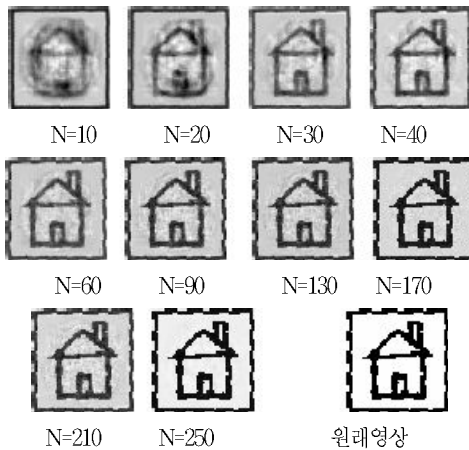


그림 18. 특징벡터 크기 N에 따른 복원 영상
 Fig 18. Recovered images with different size of feature vector, N

PC 카메라 영상으로부터 실시간으로 캡처한 영상에 대한 실험에서는 캡처된 영상에 존재하는 모든 마커를 검출하여 마커 위치를 표시하였다. 사용자가 키보드를 이용하여 인식하고자 하는 마커를 선택하고 인식 명령을 선택하면 인식 수행이 되도록 하였다. <그림 19>에는 7개의 마커가 검출되어 표시되어 있다. 가운데 위에 있는 오른쪽 방향의 마커는 다른 마커와는 다른 색으로 표시되어 있는데, 이는 현재 선택된 마커를 나타낸다.



그림 19. 다중 마커 검출 예
 Fig 19. Multiple marker detection

영상에 포함된 마커의 수를 변화시켜 가면서 마커 검출에 소요된 시간을 측정 한 결과가 <표 3>에 나타나 있다. 마커의 수가 늘어남에 따라 마커 검출 시간이 늘어나기는 하지만 11개까지 마커 수를 늘려도 초당 11프레임의 실시간 수행 속도를 얻을 수 있었다.

표 3. 마커 검출 시간
 Table 3. Execution time for marker detection

마커의 수	마커 검출 시간(ms)	프레임/초
1	76.9	13.0
3	82.2	12.2
5	82.6	12.1
7	83.8	11.9
9	84.1	11.9
11	89.9	11.1

V. 결론

본 논문에서는 카메라 영상으로부터 사각형 형태의 마커를 검출하고 인식하는 방법을 제안하였다. 본 논문에서는 여러 단계의 임계값을 사용하여 입력 영상을 이진 영상으로 변환하고 객체들의 윤곽선을 추출한 다음에 윤곽선의 기하학적 특징을 이용하여 사각형을 찾고 워핑 기법을 이용하여 사각형 마커 영상을 정사각형 형태로 정규화한 다음에 주성분 분석을 통하여 마커 영상으로부터 특징 벡터를 추출하고 표준 마커에 대한 특징 벡터와의 최소 거리법에 의해 마커의 종류를 인식하는 방법을 제안하였다. 마커의 종류를 변화시켜가면서 인식 실험을 해본 결과 마커의 종류가 증가하더라도 신뢰할만한 인식률을 얻을 수 있었고 실시간에 가까운 수행 속도를 얻을 수 있었다. 본 논문에서 제안된 방법은 향후에 착용형 컴퓨팅 환경 등에서 활용될 수 있을 것으로 기대된다.

참고문헌

[1] Y. Nakazato, M. Kanbara, N. Yokoya, "Wearable augmented reality system using invisible visual markers and an IR camera," Proc. of International Symposium on Wearable Computers, pp. 198-199, 2005

[2] Y. Kishino, M. Tsukamoto, Y. Sakane, S. Nishio, "Realizing a visual marker using LEDs for wearable computing environment," In Proc. of International Conference on Distributed Computing Systems Workshops, pp. 314-319, 2003

[3] I. Poupyrev H. Kato, M. Billinghurst. ARToolkit User Manual, Version 2.33. Human

Interface Technology Lab, University of Washington, 2000.

[4] Fiala, M., "ARTag, a fiducial marker system using digital techniques," In Proc. of Conference on Computer Vision and Pattern Recognition, pp. 590 - 596, 2005.

[5] X. Zhang, Y. Genc, and N. Navab, "Mobile computing and industrial augmented reality for real-time data access," In Proc. of IEEE Int'l Conference on Emerging Technologies and Factory Automation, 2001

[6] M. Appel and N. Navab, "Registration of technical drawings and calibrated images for industrial augmented reality," In IEEE Workshop on Applications of Computer Vision, 2000

[7] N. Osawa, K. Asai, "Distributed automatic camera control system tracking markers for distance education," In Proc. of International Conference on Information Technology: Research and Education, pp. 377-381, 2005

[8] G. Adomi, G. Destri, M. Mordonini, F. Zanichelli, "Robot self-localization by means of vision", In Proc. of Euromicro Workshop on Advanced Mobile Robot, , pp. 160-165, 1996

[9] H. Slay, B. Thomas, R. Venik, "Using ARToolkit for passive tracking and presentation in ubiquitous workspaces," In Proc. of International Augmented Reality Toolkit Workshop, pp. 46-53, 2003

[10] A. Woolard, V. Lalioti, N. Hedley, J. Julien, M. Hammond, N. Carrigan, "Using ARToolKit to prototype future entertainment scenarios," In Proc. of International Augmented Reality Toolkit Workshop, pp. 69-70, 2003

[11] W. Piekarski, B.H. Thomas, "Using ARToolKit for 3D hand position tracking in mobile outdoor environments", In Proc. of International Augmented Reality Toolkit, 2002

[12] I. Pitas, "Digital Image Processing schemes and Application", New York, John Wiley and Sons Inc., 2000.

[13] J. Hershberger and J. Snoeyink, "Speeding up the Douglas-Peucker line simplification

algorithm," In Proc. 5th Intl. Symp. Spatial Data Handling. IGU Commission on GIS, pp 134-143, 1992

[14] R. Crane, "A simplified approach to image processing," Prentice Hall, 1997

[15] D.S. Watkins, "Fundamentals of Matrix Computation," A John Wiley and Sons, Inc. 2002

저자 소개



강선경

2000년 2월 : 원광대학교 전기·전자공학부 졸업
 2004년 2월 : 원광대학교 정보·컴퓨터 교육석사 졸업
 2006년 2월 ~ 현재 : 원광대학교 컴퓨터 공학과 박사과정
 관심분야 : 휴먼 컴퓨터 인터페이스, 영상처리, 영상인식, 임베디드시스템



이상설

1984년 2월 고려대학교 전자공학과 졸업
 1989년 2월 한국과학기술원 전기및전자공학과 석사학위 취득
 1994년 2월 한국과학기술원 전기및전자공학과 박사학위 취득
 1994년~현재 원광대학교 전기전자 및 정보공학부 교수
 관심분야 : 병렬컴퓨터구조, SoC, 영상 및 통신 VLSI, 임베디드시스템



정성태

1987년 2월 서울대학교 컴퓨터공학과 (공학사)
 1989년 2월 서울대학교 컴퓨터공학과 (공학석사)
 1994년 8월 서울대학교 컴퓨터공학과 (공학박사)
 1995년 3월~현재 원광대학교 전기전자 및 정보공학부 교수
 관심분야 : 영상인식, 영상처리, 컴퓨터 그래픽스