

증강현실 구현을 위한 Multi-Marker에 관한 기법

김 학 희*

Techniques on Multi-Marker for the Implementation of Augmented Reality

Hag-Hee Kim*

요 약

본 논문에서는 가상과 현실을 이어주는 하나의 방법인 마커(Marker)인식방법에서 멀티마커방법을 사용하여 복잡한 환경이나 또는 손으로 가려지는 현상에서도 추적이 가능한 기법을 제안한다. 기존의 마커인식 방법은 한 개의 마커에만 의존하기 때문에 마커가 가려짐 현상에서 객체가 깜박이는 현상을 볼 수 있는데 이 시스템에서 마커가 가려지더라도 여러 개의 마커가 가려진 마커를 대신함으로써 증강된 객체가 나타남을 확인 할 수 있다. 이러한 기술을 구현하기 위해 관점 매트릭스(perspective matrix)를 이용한 멀티 마커 추정 방법으로 다른 연산이 없기 때문에 빠르게 추정 할 수 있는 것이 특징이다. 구현된 기술은 영상으로부터 다양한 환경에서 마커를 인지하고 위치를 추정 하는 기술로써 다양한 환경에서 3차원 객체를 표현 할 수 있다.

Abstract

This study is intended to propose a technique that can trace objects by utilizing the multi-marker method in the Marker Recognition which is a bridge way connecting virtuality and reality in complex environments or in a condition hands become obstacles. In the existing marker method, the object becomes blinking when the maker is blocked since this method relies only on single marker. However, the researcher was able to confirm the appearance of augmented object when the marker was blocked for the multiple markers replaced the blocked one. In order to implement such technique, multi-marker estimation was utilized with perspective matrix. The advantage is the fast estimation process for there is no need of other calculation. The implemented technique can detect markers and locate their positions, represent 3-D object in various circumstances.

▶ Keyword : Marker, Multi-Marker, Augmented Reality

• 제1저자 : 김학희
• 투고일 : 2010. 09. 29, 심사일 : 2010. 10. 21, 게재확정일 : 2010. 10. 25.
* 한국기술교육대학교 컴퓨터공학

I. 서론

증강현실(AR; Augmented Reality)은 실세계의 영상과 가상세계의 데이터들이 상호작용하여 이루어진 공간을 말한다. 무엇보다 직관적 인터페이스(perceptual interface)와 유비쿼터스 컴퓨팅(ubiquitous computing) 등에 대한 응용 가능성으로 다양한 형태의 증강현실 기반 프로그램들이 선보이고 있다. 이런 증강현실 시스템을 구축하기 위해서는 객체의 위치를 찾는 마커 검출과 정합, 그리고 추적과 가상물체를 합성하는 렌더링 기술이 필요하다.

이러한 증강현실을 구현하기 위해서는 실제 상황 즉, 카메라가 받은 영상의 어느 곳에 물체를 증강하는가가 가장 중요시 된다. 객체 추적하는 방식에 따라 마커인식 방식과, 마커리스인식 방식이 있는데 마커리스 방식은 특징점을 추출하여 많은 연산을 해야 함으로 영상처리에 많은 시간과 노력이 따른다. 또한 마커인식 방법은 가상과 현실을 이어주는 하나의 방식으로 마커라는 것을 사용하는 것인데 이 방식은 간편하고 쉽게 처리가 가능하기 때문에 많이 사용된다. 하지만 이러한 마커인식방법은 장면 내에 마커가 존재하지 않거나, 마커의 일부가 가려짐 등으로 인해서 마커의 정보 추출이 불가능한 경우 시스템 구동이 불가능한 단점을 가지고 있다.

본 논문에서는 이러한 문제점을 보완하기 위해서 마커인식 방법을 멀티마커(Multi-Marker)인식을 제안한다. 제안된 방법은 마커가 일부 가려지거나 화면에서 벗어난 경우 또한 복잡한 환경에서의 경우 마커의 위치를 추정 추적하여 증강된 객체의 정합을 유지하는 장점을 가지고 있다.

II. 관련연구

2.1 단일 마커 인식

2.1.1 이진화

일반적인 마커는 검은색과 흰색의 단색으로 된 객체기 때문에 이진화 만으로 큰 효율을 볼 수 있다. 하지만 이진화에 서 일반적으로 사용되는 임계값을 주는 형태는 조명이나 마커 위에 그림자가 생기는 등의 변화가 생길 때 마커의 테두리 또는 마커가 가려지는 현상이 나타 날수 있다. 이러한 특성을 고려하여 적응 이진화를 실시한다.

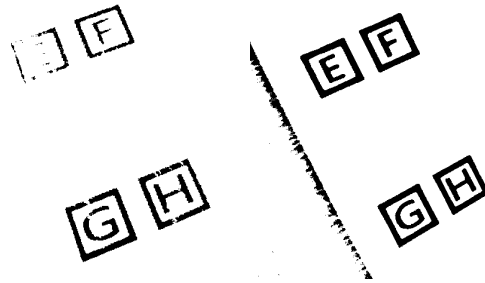


그림 1. 임계값 80에 의한 이진화 모습
Fig. 1. Binarization threshold of 80 appearance

그림 2. 적응 이진화 모습
Fig. 2. Adaptive binarization appearance

2.1.2 윤곽선 추적

윤곽선이란 물체의 가장자리 선을 말하는데 물체 내부에 비어 있는 공간이 없는 경우에는 바깥쪽 윤곽선만 존재하고 도너츠 형태의 물체처럼 물체 내부에 비어 있는 부분이 있으면 물체의 바깥쪽 윤곽선과 안쪽 윤곽선이 존재한다고 말한다. 윤곽선 추적 방법으로는 여러 가지 방법이 있는데 여기서는 물체의 가장자리에 있는 한 픽셀로부터 시작하여 가장자리에 있는 픽셀을 계속 따라가다가 시작점에 도착하면 멈추는 방법을 사용하였다.

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

그림 3. 윤곽선 추적 방법 테이블
Fig. 3. How to trace the outline table

위의 그림 3 모양처럼 인접 픽셀을 0부터 7번순으로 검색하면서 참조한다.

이 시스템에서는 이진화로 전처리된 영상을 Canny Edge 검출기를 통과하여 선들만 남기게 되고 이 영상을 윤곽선 추적을 통해 부분화 시켰다.

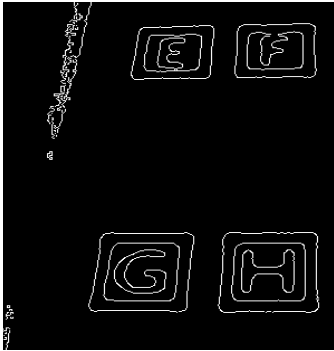


그림 4. Canny edge 검출 이후 윤곽선 추출
Fig. 4. Canny edge detection since the contour extraction

2.1.3 선분 근사화

윤곽선 검출에 의해 구한 물체의 가장자리 점들은 선분으로 근사화하면 윤곽선의 표현이 보다 간결해지고 데이터의 양도 줄어든다. 또한 선분으로 근사화한 다음에는 간단한 도형을 훨씬 더 용이하게 검출할 수 있다. 이를 통해서 사각형의 각 코너점을 구할 수 있게 된다. 선분 근사화 방법에는 여러 가지가 있는데 이번에 사용된 알고리즘은 Douglas-peucker 알고리즘이라고 불리는 방법이다. 이 알고리즘은 다음과 같은 단계로 구성된다.

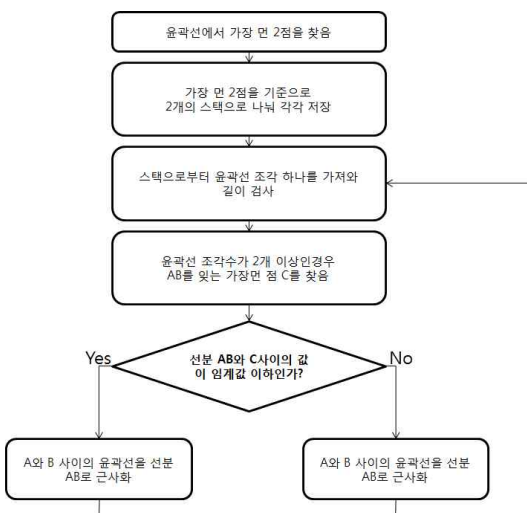


그림 5. 더글라스-파우커 알고리즘
Fig. 5. Douglas-peucker algorithm

AB와 C사이의 값이 임계값 이상인지를 판단할 때는 아래와 같은 공식을 사용하여 판단한다.

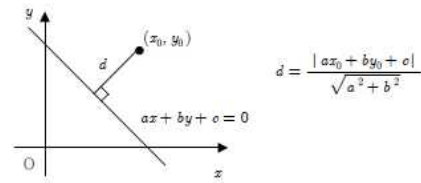


그림 6. 점과 직선에 관한 식
Fig. 6. Points on a straight line equation

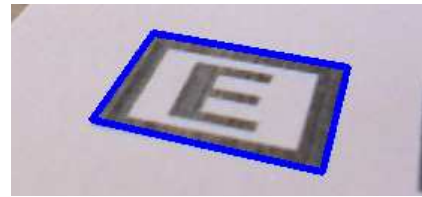


그림 7. 근사화 후 각 포인트를 연결한 모습
Fig. 7. Approximation to look after each point connected

2.1.4 Sub-pixel Corner

마커의 실제움직임은 화소 단위로 이루어지지 않는다. 그러므로 정화소 단위의 움직임 벡터는 정확도가 매우 떨어진 다. 마커의 위치를 추정하기 위해서는 정확한 각도 정보가 필요로 하는데 이를 해결하기 위해서 반화소 또는 1/4, 1/8 단위로 움직임을 추정하게 된다. Sub-pixel 단위로 움직임을 추정하기 위해서는 일반적으로 두 단계의 과정을 거친다. 첫 번째로는 탐색영역에서 정화소로 움직임을 추정하기 위해 정화소 움직임 벡터를 얻는다. 두 번째 단계에서는 정화소 움직 임 벡터를 중심으로 Sub-pixel 단위로 움직 임 벡터를 예측하여 최종 움직임 벡터를 얻는다.

표 1. 마커의 위치를 추정 테이블
Table 1. Estimation of the location of the marker table

Frame	코너 포인트	입력된 코너	Sub-Pixel 연산후
70	1	(x:444 y:291)	(x:443.3931 y:291.9692)
	2	(x:598 y:328)	(x:598 y:328)
	3	(x:616 y:259)	(x:615.7433 y:259.2557)
	4	(x:486 y:235)	(x:486 y:235.2332)
71	1	(x:444 y:291)	(x:444.5379 y:290.0936)
	2	(x:600 y:327)	(x:600.3248 y:327.3249)
	3	(x:616 y:259)	(x:615.7433 y:259.2557)
	4	(x:486 y:235)	(x:485.6139 y:235.3853)

위의 표 1 에서 보게 되면 입력된 코너의 점을 가지고 각 포인트들 간의 움직임 벡터를 추정하면 오른쪽과 같이 Sub-pixel

Corner점들을 발견하게 되는데 기존의 정수형태가 아니라 실수형태로 변화된 모습을 볼 수 있다.

이렇게 실수형태로 변화된 데이터를 나중에 멀티 마커의 위치를 추정하는데 사용된다.

2.1.5 사각형 인식

사각형 인식 방법은 위에서 해온 전처리 과정인 선분근사화와 서브픽셀코너를 거친 점들을 가지고 사각형 인지를 판단하게 된다. 처음으로 선분근사화시 사각형의 특징적으로 4점을 가짐을 알 수 있다. 이렇게 구해진 점을 이용하여 각 점들간의 각도를 계산하여 그 계산량이 90도를 기준으로 일정량 이상이거나 혹은 이하의 데이터를 가질 경우 사각형으로 인식하였다. 그리고 원형도 공식을 통해서 사각형인지를 검증하게 된다. 아래는 원형도 구하는 식과 각 모양마다의 원형도 값을 표시한다.

$$e = \frac{4\pi A}{l^2}$$

	원	정사각형	삼각형
원형도	0.92	0.80	0.57

2.1.6 Normalize

입력 영상의 사각형 마커를 인식하기 위해서는 검출된 사각형 마커의 영상을 추출하여 정사각형 또는 직사각형의 표준형으로 바꿔줘야 한다. 현재의 마커에서는 24 X 24의 정사각형을 기준으로 한다. 입력영상의 사각형의 꼭지점과 저장된 사각형 패턴의 4점을 이용하여 기준 패턴의 형태로 변경 해 줄 수 있다. 입력 패턴의 한 꼭지점의 좌표를 (x_c, y_c) 이라고 하고 저장된 사각형 패턴의 대응되는 한 꼭지점을 (x_w, y_w) 라 한다면, 다음과 같은 결과를 가질 수 있다.

$$\begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \\ p_7 & p_8 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix}$$

위와 같은 식에 각각의 입력영상의 사각형 꼭지점과 저장된 사각형 꼭지점을 넣어주면 $p_i, i = 1, 2, \dots, 8$ 를 구할 수 있다. 이를 풀어 쓰면 아래와 같다.

$$\begin{bmatrix} x_w & y_w & 1 & 0 & 0 & 0 & -x_c x_w - x_c y_w \\ 0 & 0 & 0 & x_w & y_w & 1 & -y_c x_w - y_c y_w \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \\ p_8 \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \end{bmatrix}$$

이렇게 구해진 값을 이용하여 우리가 원하는 정사각형 패턴 또는 직사각형 패턴을 만들 수 있다.



그림 8. 왼쪽은 원본영상 오른쪽은 Normalize된 영상
Fig. 8. Normalize the original image as a left-right image

2.1.7 Template Matching

템플레이트 매칭이란 검사할 영상이 주어졌을때 미리 주어진 템플레이트 영상을 이용하여 검사할 영상 내부에 있는 유사한 영상패턴을 찾아내는 기법이다. 이때 템플레이트는 일종의 모델 영상이 된다.

$$MAD = \frac{1}{MN} \sum_{i=0}^M \sum_{j=0}^N |T(x_i, y_j) - I(x_i, y_j)|$$

위는 템플릿 매칭에서 사용되는 연산식이다 템플레이트 T와 검사할 영상의 겹쳐진 영상부분 I를 서로 겹쳐놓고 대응되는 픽셀의 밝기값을 서로 비교해서 차이를 더하는 방식이다.

현재 우리의 시스템에서 사용되는 모델은 알파벳 형태의 문자 모델을 사용한다. Normalize된 이미지를 이것을 인식하기 위해서 90도의 회전을 시켜가면서 매칭을 하고 그와 유사도를 측정한다.

2.2 Camera Calibration

마커위에 객체를 증강하기 위해서는 평면 영상에서 3차원의 위치 좌표를 추정 하는 것이 매우 중요하다. 여기서는 핀홀 카메라 모델의 원리를 이용하여 사각형의 꼭지점 벡터를 통해 3차원 좌표를 추정하고, 동차좌표계 행렬을 얻는 알고리즘을 사용하였다.

2.2.1 핀홀 카메라 모델

3차원 실세계 객체 위의 한 점에서 빛이 발산되면 그 중 하나의 광선만이 핀홀을 통과하고, 핀홀을 통과한 광선은 영상 평면에 투영된다. 결과적으로 영상 또는 투영 평면이라고 불리는 곳에 영상이 맺히게 되는데, 객체가 영상에서 얼마만큼 한 크기로 상이 맺히는 지는 핀홀 카메라[4]의 초점 거리에 의해 결정된다.

이상적인 핀홀 카메라의 경우, 핀홀 구멍으로부터 영상 평면까지의 거리가 초점 거리가 된다.

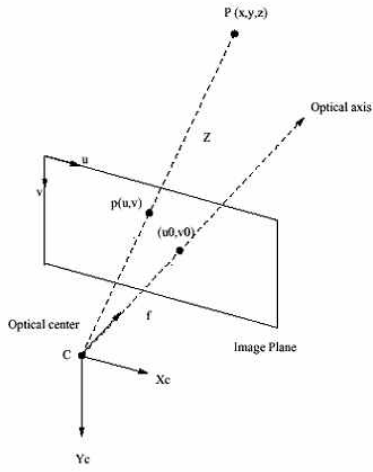


그림 9. 핀홀 카메라 영상 모델
Fig. 9. Pinhole Camera Model

이 그림에서 f 는 핀홀 카메라의 초점 거리고, Z 는 카메라와 객체 사이의 거리를 나타낸다.

2.2.2 벡터를 이용한 3차원 포지션 측정

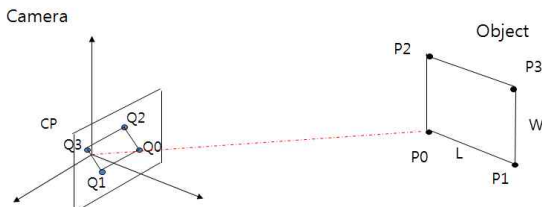


그림 10. 벡터를 이용한 3차원 포지션 측정
Fig. 10. Measured using a three-dimensional position vector

벡터를 이용한 3차원 포지션 측정하는 방법[4]은 입력 받은 화면상의 사각형 4점의 꼭지점 Q_0, Q_1, Q_2, Q_3 에 일정한 K_0, K_1, K_2, K_3 상수를 곱해서 원하는 3차원 공간의 좌표를 추출한다. 수식으로 표현하면 아래와 같이 된다.

$$P_i \text{ Cam} = k_i \times Q_i \text{ Cam} \quad (i = 1, 2, 3, 4)$$

위 수식을 이용하여 각 벡터를 표현 하면 아래 그림과 같다.

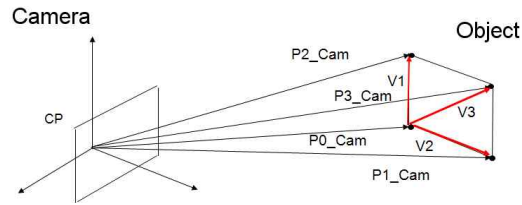
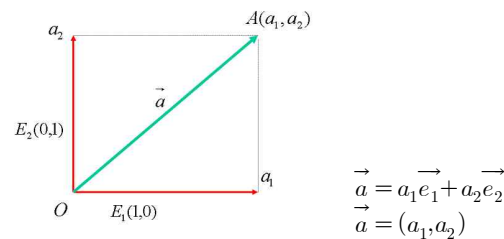


그림 11. 수식에 의한 벡터 표현 결과
Fig. 11. Vector representation of the formula results

이 수식을 이해하기 위해서는 벡터의 합을 이용한다. 벡터의 합은 다음과 같이 표현된다.



벡터의 특성중 각 벡터 2개의 합으로 원하는 좌표 값을 추출 할 수 있다는 것을 이용한다. 이 방법을 통해서 다음과 같은 수식을 세울 수 있다.

$$(P_1 \text{ Cam} - P_0 \text{ Cam}) + (P_2 \text{ Cam} - P_0 \text{ Cam}) = (P_3 \text{ Cam} - P_0 \text{ Cam})$$

위의 수식을 보면 벡터 $\vec{V_1} + \vec{V_2} = \vec{V_3}$ 를 이용한다는 것을 수 있다. 위의 수식을 간략히 하면

$$\begin{aligned} (k_1 \times Q_1 \text{ Cam} - k_0 \times Q_0 \text{ Cam}) + (k_2 \times Q_2 \text{ Cam} - k_0 \times Q_0 \text{ Cam}) \\ = (k_3 \times Q_3 \text{ Cam} - k_0 \times Q_0 \text{ Cam}) \\ (k_1 \times Q_1 \text{ Cam}) + (k_2 \times Q_2 \text{ Cam} - k_0 \times Q_0 \text{ Cam}) = (k_3 \times Q_3 \text{ Cam}) \end{aligned}$$

여기서 $k_i = k_i/k_3$ 라고 가정하면

$$(k_1' \times Q_1 \text{ Cam}) + (k_2' \times Q_2 \text{ Cam} - k_0' \times Q_0 \text{ Cam}) = Q_3 \text{ Cam}$$

라는 식을 얻을 수 있다.

여기서 k_0', k_1', k_2' 의 값은 선형 방정식을 통해서 구할 수 있다.

실제 화면상의 투영되는 대각선의 길이는 실제 마커의 대각선의 길이와 같다는 것을 이용하여 k_3 의 값을 유도한다. 실제 k_3 의 값이 유도되면 k_0', k_1', k_2' 을 구할 수 있다.

$$\begin{aligned} |P_3 Obj - P_0 Obj| &= |P_3 Cam - P_0 Cam| \\ &= |k_3 \times Q_3 Cam - k_0 \times Q_0 Cam| \\ k_3 &= |P_3 Obj| / |k_0 \times Q_0 Cam - Q_3 Cam| \end{aligned}$$

이렇게 구해진 P_0, P_1, P_2, P_3 으로 동차(homogeneous) 좌표계를 생성 할 수 있는데 식은 아래와 같다.

$$\begin{aligned} Unit_x &= (P_1 Cam - P_0 Cam) / |P_1 Cam - P_0 Cam| \\ Unit_y &= (P_2 Cam - P_0 Cam) / |P_2 Cam - P_0 Cam| \\ Unit_z &= Unit_x \times Unit_y \end{aligned}$$

$Unit_z$ 의 경우에는 $Unit_x$ 와 $Unit_y$ 의 외적의 성질을 이용하여 구할 수 있게 된다.

이렇게 생성된 값을 이용하여

$$H = \begin{pmatrix} Unit_x^T & Unit_y^T & Unit_z^T & P_0 Cam^T \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

의 동차 좌표계를 얻을 수 있다.

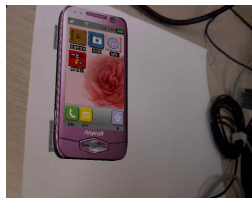


그림 12. X축 회전 변화
Fig. 12. X-Axis Rotation

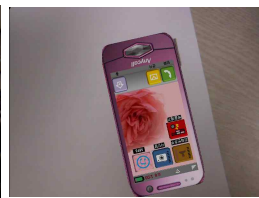


그림 13. Z축 회전 변화
Fig. 13. Z-Axis Rotation

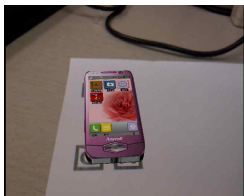


그림 14. Y축 회전변화
Fig. 14. Y-axis Rotation



그림 15. 이동에 따른 변화
Fig. 15. Move and Changes

III. 구현

이 논문에서는 perspective matrix를 이용한 멀티 마커 추정 방법을 소개 한다. perspective matrix를 앞에서 normalize 에서 사용한 행렬을 뜻한다. 다른 연산이 없기 때문에 빠르게 추정 할 수 있는 것이 특징이다. perspective matrix를 이해 하기 위해서는 3가지의 변화에 따른 matrix가 필요하다.

3.1 제안된 알고리즘

전체 시스템의 프로세서는 그림 16 과 같다.

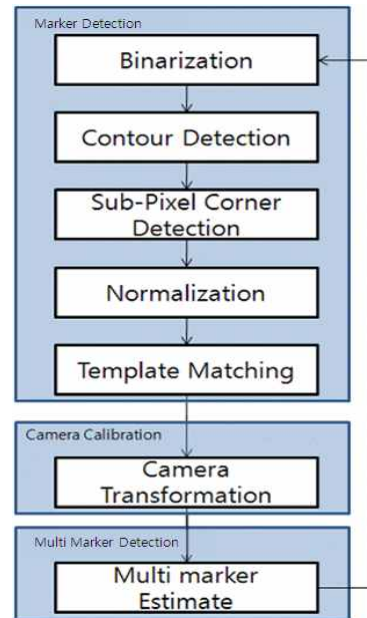


그림 16. 전체 시스템의 프로세서
Fig. 16. All the system's processor

비디오 영상으로부터 얻어진 데이터에서 이진화 및 윤곽선 검출을 통해서 사각형을 찾아내고 이미 기록된 마커의 ID와 템플링 매칭 기법을 사용하여 마커의 ID를 판단하고, 이렇게 구해진 화면상에 마커를 가지고 마커의 각 꼭지점에 대한 3차원 공간을 판단하여 멀티 마커의 위치를 판단한다.

3.2 Isometries transformations

평면에서의 각도의 변화를 뜻하는 매트릭스로, x,y의 각도 변화량을 매트릭스화 시킨 것이다.

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \epsilon \cos \theta & -\sin \theta & t_x \\ \epsilon \sin \theta & \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\epsilon = \pm 1$$

$$X' = H_s X = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} X$$

평면에서는 1개의 회전과 2개(x,y)의 이동의 자유도를 가진다.

3.3 Similarity transformations

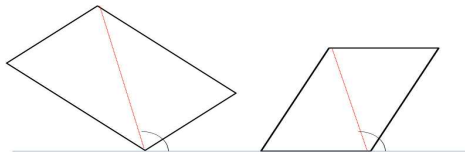
$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} s \cos \Theta - s \sin \Theta t_x \\ s \sin \Theta \quad s \cos \Theta \quad t_y \\ 0 \quad 0 \quad 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$X' = H_s X = \begin{bmatrix} s R t \\ 0^T 1 \end{bmatrix} X$$

Similarity transformations은 Isometries transformation에서 Scaling값을 더한 것으로 4개의 자유도를 가지게 된다.

3.4 Affine transformations

Affine 변환은 아래 그림에서 보이듯이 한축을 기준으로 양쪽의 점들이 대칭을 이룬다는 것을 기본으로 한다. 식으로 표현하게 되면 다음의 식이 된다.



$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

이것을 간단히 표현하게 되면 아래와 같다.

$$X' = H_A X = \begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix} X$$

$$A = R(\Theta) R(-\Theta) D R(\Theta)$$

D는 고유값을 나타낸다. 이것을 이용하여 얼마의 각도가 휘어 있는지도 판단하게 된다. Affine Matrix인 경우 6개의 자유도를 가지게 된다.

3.5 Perspective transform

위의 3.3.2, 3.3.2, 3.3.3의 식들을 한 대 묶으면 아래와 같은 식을 구할 수 있게 된다.

$$H = H_s H_a H_p \begin{bmatrix} s R t \\ 0^T 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0^T 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ v^T & v \end{bmatrix} = \begin{bmatrix} A & t \\ v^T & v \end{bmatrix}$$

$$A = s R K + t v^T$$

perspective matrix를 살펴 보게 되면 각도의 변화 및 스케일을 가지는 Similarity matrix와 상수 K 그리고, 왜곡변수 v의 값으로 이루어진다.

3.6 Multi Marker의 위치 추정

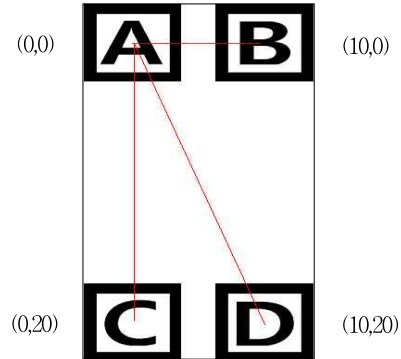


그림 17. 마커의 절대좌표
Fig. 17. Absolute position of the marker

위의 그림처럼 실제 사용되는 멀티 마커로 현실세계에서 이미 마커의 상대가 되는 절대좌표를 이미 알고 있다. 절대좌표를 알기 때문에 절대 좌표를 기준으로 상대적으로 원하는 좌표를 빼주게 되면 추정 하고 싶은 상대 좌표를 구할 수 있게 된다. 이렇게 구해진 값을 이용하여 조금 전 구한 perspective matrix에 각 마커를 기준으로 한 상대 좌표만 곱해준다면 우리가 원하는 추정 좌표를 구할 수 있게 되는 것이다.

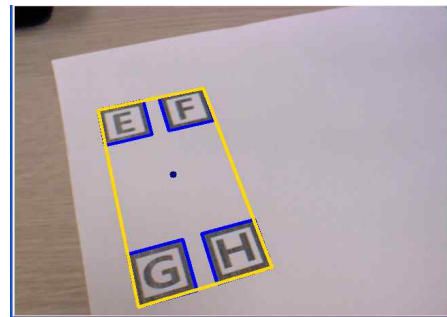


그림 18. 실제 추정 화면
Fig. 18. The actual estimated surface

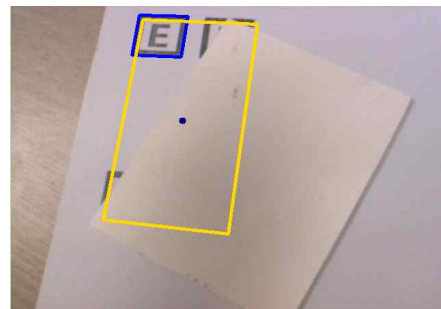


그림 19. 세 개가 가려졌을 때
Fig. 19. Posted three to go

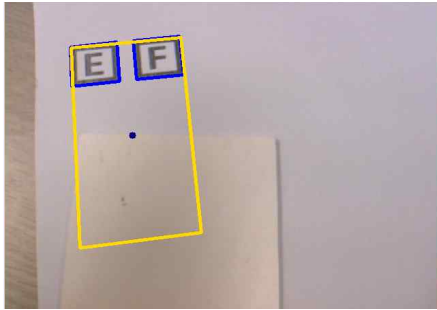


그림 20. 두 개가 가려졌을 때
Fig. 20. Posted two to go

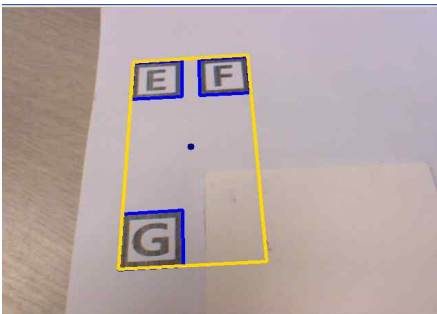


그림 21. 한 개가 가려졌을 때
Fig. 21. Posted one to go

IV. 결 론

본 논문의 제안 시스템은 배경 및 밝기 값에 적응적인 임계값을 사용함으로써 음영지역에도 정확하게 마커를 획득할 수 있다. 또한 기존의 마커인식 방법은 한 개의 마커에만 의존하기 때문에 마커가 일부 또는 마커 전체가 가려짐 현상에서 증강된 객체가 움직이지 않거나, 깜박이는 현상을 볼 수 있는데 본 연구된 시스템에서 마커가 가려지더라도 여러 개의 마커가 가려진 마커를 대신함으로써 증강된 객체가 나타나고 상호 인터페이스가 될 수 있는 것을 확인 할 수 있었다.

본 논문에서 구현된 기술은 영상으로부터 다양한 마커를 인지하고 위치를 추정하는 기술로써 다양한 환경에서 3차원 객체를 표현 할 수 있다. 따라서 본 시스템 이용하면 실제 영상에서 사용자가 조작 객체를 이용하여 다양한 3차원 가상 객체를 쉽게 합성할 수 있으며, 조절할 수 있기 때문에 방송, 게임, 영화, 특수효과, 교육 등의 여러 분야에서 널리 응용될 것으로 기대된다. 본 논문은 멀티마커 기반으로 가상 객체를 증강하는 것에 초점을 두었다. 그러나 향후 연구 과제로써 많은 객체에 대한 마커의 수를 줄여나가는 것과 실제 환경에서의 상호 조명 효과에 관한 연구 등으로 진행되어야 할 것이다.

참고문헌

- [1] Kato H., Billingham M., Blanding B., May R., "ARToolkit," Technical Report, Hiroshima City University, December, 1999.
- [2] M. Fiala, "ARTag, a Fiducial Marker System Using Digital Techniques," Proc. IEEE Intl. Computer Vision and Pattern Recognition Conference, pp 590-596, 2005.
- [3] E. Umlauf, H. Piringer, G. Reitmayr, D. Schmalstieg, "ARLib: The Augmented Library," Poster paper in Proc. ART'02, Darmstadt, Germany, Sept. 29, 2002.
- [4] R.Y. Tsai, "A Versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," IEEE Journal of robotics and Automation, Vol. RA-3, No. 4, Page(s): 323-344, 1987
- [5] Y. Hung, P. Yeh, D. Harwood "Passive Ranging to Known Planar Point Sets," IEEE Int'l Conf Robotics and Automation, 1985.
- [6] K. Kutulakos and J. Vallino, "Calibration-Free Augmented Reality," IEEE Trans. Visualization and Computer Graphics, Vol. 4, No. 1, pp. 1-20, 1998.
- [7] G. Simon, M. Berger, "Registration with a Moving Zoom Lens Camera for Augmented Reality Applications," ECCV (2), pp. 578-594, 2000.

저 자 소 개



김 학 희

2000년: 충북대학교 이학사

2008년: 한국기술교육대학교 공학석사

2009년~현재: 한국기술교육대학교

컴퓨터공학 박사과정

관심분야: 영상처리, 컴퓨터비전

멀티미디어