심플 프레임 마커: 마커 내부 이미지 및 문자 패턴의 인식 및 추적 기법 구현

Simple Frame Marker: Implementation of In-Marker Image and Character Recognition and Tracking Method

김혜진, Hyejin Kim*, 우운택, Woontack Woo**
광주과학기술원, U-VR 연구실

요약 본 논문에서는 증강현실에서 마커에 포함된 이미지뿐만 아니라 문자 인식을 지원하기 위한 심플 프레임 마커를 제안한다. 마커 내부에 임의의 패턴 대신에 문자를 삽입하고 문자 인식 알고리즘(Optical Character Recognition)을 사용하여 인식하면 실행 전 학습과정이 필요 없을 뿐만 아니라 문자의 친숙함 때문에 시각적 장애요인도 줄일 수 있다. 따라서 기존의 마커 방식인 이미지뿐만 아니라 문자도 인식하기 위해서 제안된 심플 프레임 마커는 정의된 마커의 가로세로 비율에 따라 이미지타입의 마커(Square SFMarker)인지 문자타입의 마커(Rectangle SFMarker)인지를 구별하고 각기 다른 인식 알고리즘을 적용한다. 또한 문자 인식을 위한 전처리 과정을 줄이기 위해 디자인 단계에서 마커 테두리에 방향정보를 삽입하고, 인식 단계에서는 이 방향 정보를 추출하여 문자 인식을 빠르고 정확하게 수행한다. 마지막으로 매 프레임 문자를 인식하는 알고리즘을 수행 시 추적 속도가 저하되므로, 프레임간 변화량이 적을 때는 이전 프레임의 인식 결과 정보를 사용하여 수행 속도를 높인다.

Abstract In this paper, we propose Simple Frame Marker(SFMarker) to support recognition of characters and images included in a marker in augmented reality. If characters are inserted inside of marker and are recognised using Optical Character Recognition(OCR), it doesn't need marker learning process before an execution. It also reduces visual disturbance compared to 2D barcode marker due to familarity of characters. Therefore, proposed SFMarker distinguishes Square SFMarker that embeds images from Rectangle SFMarker with characters according to ratio of marker and applies different recognition algorithms. Also, in order to reduce preprocessing of character recognition, SFMarker inserts direction information in border of marker and extracts it to execute character recognition fast and correctly. Finally, since the character recognition for every frame slows down tracking speed, we increase the speed of recognition process using the result of character recognition in previous frame when frame difference is low.

핵심어: Simple Frame Marker, Image Recognition, Character Recognition, Mobile Augmented Reality, Fiducial Marker, Marker Tracking

본 연구는 문화체육관광부 및 한국문화콘텐츠진흥원의 문화콘텐츠기술연구소육성사업의 연구결과로 수행되었음.
*주저자 : 광주과학기술원 정보통신공학과 박사과정 e-mail: hjinkim@gist.ac.kr
**교신저자 : 광주과학기술원 정보통신공학과 교수 e-mail: wwoo@gist.ac.kr

1. 서론

기존의 증강 현실 연구에서는 마커의 구별을 위해서 내부의 이미지를 인식하기 때문에 해당 마커를 증강할 대상으로 사용하기 전 학습과정이 별도로 필요하다.[1] 이러한 문제점을 해결하기 위해 사용 전 학습이 필요 없는 2D 바코드 형태의 마커가 제안되었다.[2] 하지만 이 경우 2D 바코드 형태가 부자연스러운 시각적 장애 요인이 된다. 한편, 문자를 마커의 내부에 삽입하여 문자 인식 알고리즘으로 인식하면 학습과정이 필요 없을 뿐만 아니라 문자의 친숙함 때문에 시각적 장애요인도 줄일 수 있다.

본 논문에서는 기존의 마커 내부의 이미지뿐만 아니라 문자의 인식을 지원하기 위해 심플 프레임 마커를 제안한다. 심플 프레임 마커는 다음과 같은 세 가지 특징을 지닌다. 첫째, 마커 내부의 이미지뿐만 아니라 문자를 인식하기 위해 마커의 가로, 세로의 비율에 따라 이미지를 포함하는 마커인지 문자를 포함하는 마커인지 구별하고 그에 따른 인식 알고리즘을 적용한다. 둘째, 문자 인식을 위한 문자 포함 영역 추출과 문자 시작점 설정 등의 전처리 과정을 줄이기 위하여 마커 영역과 테두리에 삽입된 방향정보를 이용한다. 추출된 방향정보로 문자 영역의 시작점을 설정함으로써 문자 인식을 빠르고 정확하게 수행할 수 있다. 셋째, 매 프레임 문자를 인식하고 자세 추정을 하게 되면 속도가 저하되기 때문에 프레임 변화량이 적을 때 이전 프레임에서 얻은 문자 인식의 결과 정보를 사용함으로써 문자가 포함된 마커의 추적 속도를 높일 수 있다.

2. 심플 프레임 마커

2.1 심플 프레임 마커 정의

그림 1은 심플 프레임 마커 타입을 보여주고 있는데 마커 내부에 이미지가 포함된 경우에는 정사각 형태로 문자가 포함된 경우에는 직사각 형태로 디자인 한다. 따라서 수식 (1)과 같이 두 종류의 마커로 인식하게 된다. 또한 문자 인식 시 전처리 과정을 줄이기 위해 마커 테두리 중 한 변에 방향 정보를 삽입하게 되는데 이때 방향 정보는 수식 (2)와 같이 동, 서, 남, 북의 네 가지로 구분된다.

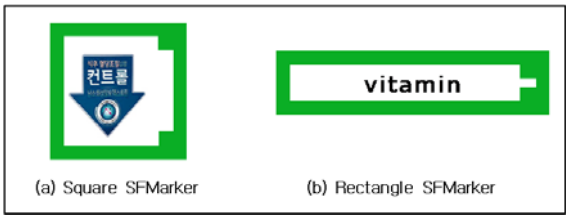


그림 1. 심플 프레임 마커 타입

$$type \in \{Square\ SFMarker, Rectangle\ SFMarker\} \quad (1)$$

$$dir \in \{East,\ West,\ South,\ North\} \quad (2)$$

심플 프레임 마커는 사각형 마커 내부에 이미지와 문자를 포함하는 마커로 내부의 인식을 위해 수식 (3)과 같이 이미지와 문자 타입의 마커로 구별하여 각각 내부에 맞는 알고리즘–이미지 매칭 모듈이나 문자 인식 모듈 등–을 적용한다. 여기에서 p는 $flag$로 마커가 $Square\ SFMarker$로 판단되면 1이 된다. 그리고 $Template\ Matching$(이미지 매칭)이 수행된다. 만약 마커가 $Rectangle\ SFMarker$로 판단되면 p는 0이 되어 $Character\ Recognition$(문자 인식: Optical Character Recognition)이 수행된다. 따라서 D^*의 출력 값은 $type$(마커타입), dir(방향정보)과 id(이미지인 경우) 또는 $string$(문자인 경우)이 된다.

$$D^* = p \times \arg max_{dir,id}(Template\ Matching) + (1-p) \times \arg max_{dir,string}(Character\ Recognition)$$

$$p = flag(Square\ SFMarker)$$

$$D^* \in \{(x, y, z) / x = type, y = dir, z \in \{id, string\}\} \quad (3)$$

2.2 심플 프레임 마커 처리과정

심플 프레임 마커의 처리과정은 그림 2와 같이 1)마커 검출 전처리, 2)마커 타입 구별, 3)마커 방향 추출로 심플 프레임 마커 타입 결정, 4)심플 프레임 마커 내부 인식 및 추적으로 이루어진다. 프로그램이 시작되면 이진화를 위한 임계값을 설정하고 charFlag를 0으로 초기화한다. charFlag는 이전 프레임의 문자 인식 결과를 이용하기 위한 플래그 변수이다. 이후, 이미지를 포함한 마커인 경우 템플릿 매칭을 위한 이미지 모델을 등록한다. 이미지 모델의 등록은 기존의 이미지 기반 마커에서 필요한 과정으로 프로그램 외부에서 먼저 등록할 이미지 모델을 학습한다.

카메라로부터 이미지가 획득되면 영상을 이진화하고 라벨링으로 영역을 획득한다. 그리고 경계선 추적 과정을 거쳐 추출된 영역에서 꼭지점을 검사하여 사각형 영역인지 판단한다. 영역이 사각형으로 판단되면 해당 객체를 호모그라피와 샘플링을 통하여 그림 3 (a), (b)의 좌상단의 작은 템플릿과 같이 정사각형 비율로 조정한다. 이때 마커 프레임 테두리의 가로, 세로 비율을 구하고 수식 (4)와 같이 임계값과 비교하여 정사각형 타입인지 직사각형 타입인지 결정한다. R은 정사각형 비율로 조정된 이미지에서 테두리의 가로, 세로 너비의 비율이고, Th는 타입을 구별하기 위한 가로, 세로 비율의 임계값이다. 본 논문에서는 Th를 1로 설정하여 마커타입을 정사각형과 직사각형으로 구분하였다. 만약 다른

값으로 설정한다면 그에 따라 비율이 다른 사각형을 구별하게 된다.

· $\frac{1}{Th} \le R \le Th$: SquareSFM (Less than $1:Th$)

· $R < Th$ or $Th < R$: RectangleSFM (More than $1:Th$) (4)

이후, 꼭지점의 형태를 비교하여 정사각형과 직사각형 타입에 따른 방향정보를 추출하고 방향정보가 추출되면 각각 정사각형 심플 프레임 마커와 직사각형 심플 프레임 마커로 결정하게 된다. 인식과 추적의 단계에서는 정사각형 심플 프레임 마커인 경우, 방향 정보를 이용해 등록한 이미지 모델과 템플릿 매칭과 자세 추정을 수행하여 콘텐츠를 렌더링한다. 직사각형 심플 프레임 마커인 경우, charFlag의 값이 초기화한 0이면 방향 정보를 이용해 문자 인식과 자세 추정을 수행하고 charFlag의 값을 1로 변경하여 콘텐츠를 렌더링한다. 만약 charFlag의 값이 1이면 문자인식을 수행하지 않고 이전 프레임에서 획득한 문자 인식 결과를 이용하여 추적 속도를 높인다. 이후, charFlag의 값은 매 프레임마다 프레임간 변화량이 특정 값 이상으로 커지면-즉, 마커가 사라졌다가 나타나게 되면-새로운 마커가 나타난 것으로 가정하고 0으로 초기화되어 문자 인식을 다시 수행하게 된다.

3. 구현 및 실험

그림 3은 정사각형과 직사각형 타입의 심플 프레임 마커 인식 및 추적 결과의 예를 각각 보여준다. 구현을 위해 사용한 소프트웨어는 OpenCV 1.0[3], osgART 1.1[4]과 문자인식을 위한 Tesseract[5]이다. 다음 3.1~3.3에서는 본 논문의 특징에 대해 검증하는 실험을 설명한다. 실험은 심플 프레임 마커의 테두리 크기를 0.5cm로 하고 약 20cm이내의 동작 거리에서 진행하였다. 심플 프레임 마커의 인식과 추적을 수행하기 위한 모바일 장치로는 Intel(R) Core2 Duo CPU 2.40GHz의 Notebook과 Sony VAIO의 VGN-UX17LP UMPC를 각각 사용하였다.

3.1 심플 프레임 마커 타입 구별 시 처리시간과 정확도

심플 프레임 마커를 정사각형과 직사각형 타입으로 구별하는데 걸리는 처리시간과 정확도는 표 1과 같다. 마커 타입을 구별하는 과정은 기존의 마커 인식보다 부가적인 계산을 필요로 하지만 평균 4~6ms에 수행됨을 알 수 있다. 따라서 기존의 이미지 인식 및 사각형 추적 알고리즘과 통합했을 때의 속도가 약 12~15fps를 나타냄으로써 기존 알고리즘과 연계가 가능함을 확인할 수 있다. 또한, 정확도는 앞서 언급한 테두리의 크기와 거리에서 총 20회 동안 임의의 각도에서 심플 프레임 마커를 바라보았을 때 구별에 성공한 횟수의 백분율로 나타내었다. 약 20cm이내에서는 정확도가 95% 이상이지만 20cm이상의 거리에서는 꼭지점의 방향정보의 해상도가 떨어져 정확도가 떨어짐을 확인할 수 있었다.

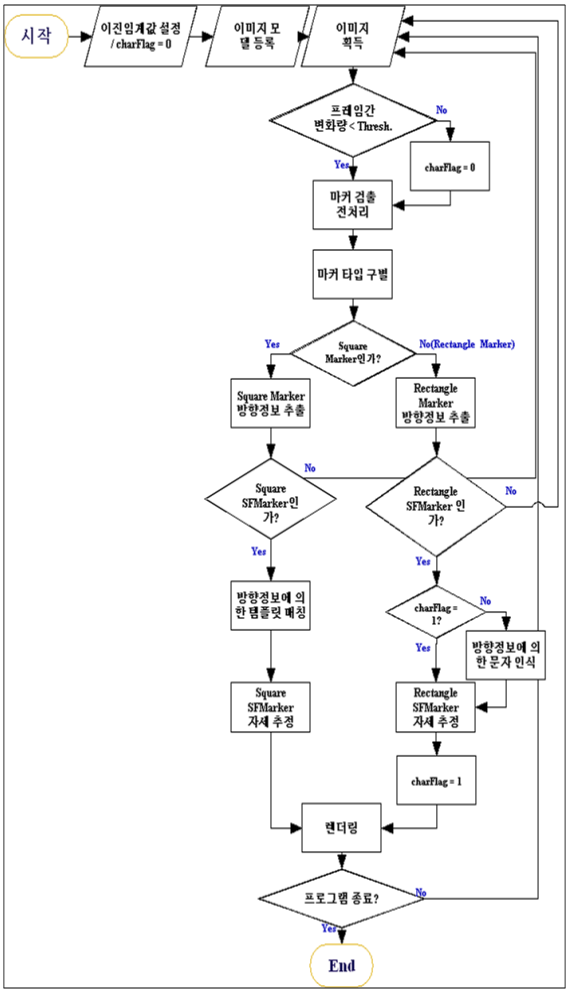


그림 2. 심플 프레임 마커 처리 순서도

표 1. 심플 프레임 마커 타입 구별의 처리시간과 정확도

수행 장치	처리시간	정확도
Notebook	약 4ms	〉95%
UMPC	약 6ms	〉95%

3.2 방향 삽입 유무에 따른 문자인식 처리시간

표 2는 심플 프레임 마커에 방향을 삽입하지 않았을 때와 삽입했을 때의 문자 인식의 처리시간에 대한 비교 측정 결과를 보여준다. 방향을 추출할 수 있는 경우 한 번의 문자인식 처리시간이 걸리지만 방향을 추출할 수 없는 경우는 사각형의 각 면을 상위 면으로 가정하고 문자인식을 수행함으

로써 4배의 처리시간이 걸리게 된다.

표 2. 방향 삽입 유무에 따른 문자인식 처리시간

수행 장치	방향을 삽입하지 않았을 때의 문자인식 처리시간	방향을 삽입했을 때의 문자인식 처리시간
Notebook	약 1.2s	약 0.3s
UMPC	약 1.6s	약 0.4s

3.3 이전 프레임 정보사용에 따른 문자 추적 속도 향상

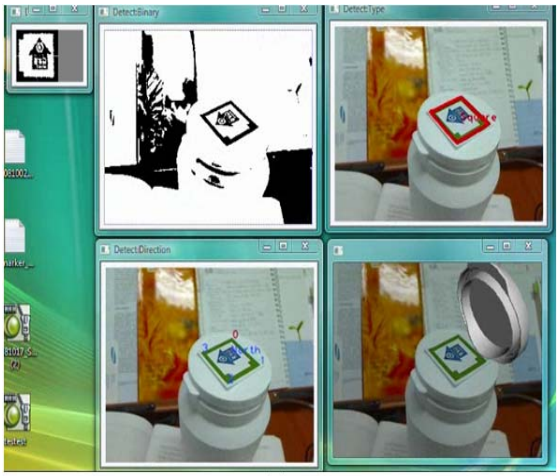
매 프레임 문자를 인식하여 추적을 수행할 때와 이전 프레임의 문자인식 결과를 사용하여 추적을 수행할 때의 속도를 비교 측정한 결과는 표 3과 같다. 이전 프레임의 문자인식 결과를 이용하기 위해서는 프레임 변화량을 측정해야 하는데 본 연구에서는 간단하게 마커가 사라졌다가 새로 나타나는 경우를 프레임 변화량이 크다고 판단하여 새로운 문자인식과 추적을 수행하였다. 반면에 프레임은 바뀌지만 마커가 계속 유지되어 보이는 경우 프레임 변화량이 작다고 판단하여 문자인식의 이전 결과 값을 이용하고 사각형 추적만 수행하였다. 이때, 병을 빠르게 움직여 프레임 변화와 동시에 앞, 뒷면에 붙은 마커가 교체되어 보이는 경우는 다른 문자임에도 불구하고 같은 문자로 인식되는 문제가 있었지만 이것은 극단적인 경우에 해당하였다.

표 3. 이전 프레임 정보사용에 따른 문자 추적 속도

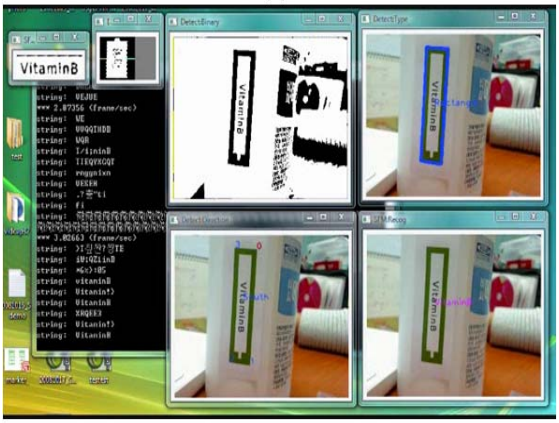
수행 장치	매 프레임 문자인식과 추적 시 속도	이전 프레임 문자인식 결과를 사용하여 추적 시 속도
Notebook	3~5fps	〉20fps
UMPC	1~3fps	〉17fps

4. 결론 및 추후연구

본 논문에서는 증강현실에서 이미지뿐 아니라 문자를 인식하기 위한 심플 프레임 마커를 제안하였다. 제안된 심플 프레임 마커는 마커의 가로, 세로 비율을 계산하여 이미지를 포함한 정사각형 타입과 문자를 포함하는 직사각형 타입을 구별하여 그에 따른 인식 알고리즘을 적용한다. 한편, 문자인식 알고리즘을 빠르고 정확하게 수행하기 위해서 마커 테두리에 삽입된 방향정보를 추출하여 문자 영역 시작점을 설정한다. 또한, 문자의 인식이 템플릿 매칭의 이미지 인식에 비해 속도가 느려 매 프레임 인식 시 추적이 어려워지기 때문에 이전 프레임의 문자 인식 결과 값을 이용하여 추적 속도를 빠르게 한다. 현재 구현된 심플 프레임 마커는 테두리의 크기가 0.5cm로 약 20cm이내의 거리에서 동작한다. 또한, 마커 타입의 비율은 1:1의 정사각형과 1:4의 직사각형을 사용하였다. 추후 연구로는 심플 프레임 마커의 테두리 두께와 마커 타입을 결정하는 비율에 대해 최적화하는 방법을 찾는 것이 필요하다.



(a)



(b)

그림 3. (a) Square SFMarker의 인식 및 추적 결과 (b) Rectangle SFMarker의 인식 및 추적 결과

참고문헌

[1] H. Kato, M. Billinghurst, and I. Poupyrev, "The magicbook – moving seamlessly between reality and virtuality", IEEE Computer Graphics and Applications, Vol. 21, No. 3, pp. 6~8, 2001.

[2] M. Fiala, "Artag, a fiducial marker system using digital techniques", In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 590~596, 2005.

[3] OpenCV, http://tech.groups.yahoo.com/group/OpenCV/

[4] osgART, http://www.artoolworks.com/community/osgart/

[5] tesseract-ocr, http://code.google.com/p/tesseract-ocr/