

패치 기반 텍스처 합성을 활용한 실시간 마커 은닉*

Realtime Marker Concealment using Patch-based Texture Synthesis

윤경담, Kyungdahm Yun*, 우운택, Woontack Woo**

*광주과학기술원 문화기술연구센터, **광주과학기술원 U-VR 연구실

요약 본 논문에서는 자연스러운 증강현실 환경을 위하여 패치 기반의 텍스처 합성을 통한 마커 은닉 방법을 제안한다. 증강현실에서 카메라의 자세를 구하기 위한 보편적인 방법은 음영 대비가 뚜렷한 정사각형의 마커를 사용하는 것이다. 이러한 인위적인 마커의 사용은 물체의 인식과 추적을 용이하게 하지만 증강된 장면의 실감성을 감소시켜 사용성 저하를 유발하기도 한다. 제안된 마커 은닉 방법은 실시간성을 보장하면서, 배경 텍스처의 전역적인 특성을 유지하고, 주변 환경의 변화에 유연하다.

핵심어: 증강현실, 마커 은닉, 텍스처 합성, 영상 인페인팅

1 서론

증강현실 환경에서 카메라 영상 속의 물체를 추적하기 위하여 명암비가 큰 정사각형 마커(marker)가 보편적으로 사용되고 있다. 이러한 마커의 인위적 사용은 물체의 인식과 추적을 용이하게 하지만 증강된 장면의 실감성을 감소시키기도 한다. 특히 가상객체가 마커의 크기보다 작거나 마커에서 이격된 위치에 증강되는 경우에는 영상 속의 마커를 가릴 수 없기 때문에 사용자의 몰입을 방해하는 요소가 될 수 있다.

인위적인 마커의 사용을 지양하기 위해 적외선에 반응하는 투명 마커를 사용하는 방법[1]과 마커 대신 영상의 특징점을 추적하는 방법이 제안되었다. 그러나 전자는 별도의 장비로 고정된 환경을 요구하고, 후자는 영상에 따라 안정적인 특징점 추출이 어려운 단점이 있다. 최근에는 현실공간의 마커를 물리적으로 제거하는 대신 가상공간의 마커를 은닉하는 방법이 시도되기도 하였다. 즉, 마커 영역을 주변 배경과 이질감이 적은 영상으로 합성, 대체하는 영상 인페인팅(inpainting)을 활용한다. 이러한 인페인팅[2]에는 비모수적(non-parametric)인 텍스처 합성(texture synthesis) 기법을 적용할 수 있으며 합성의 단위로 개별 픽셀[3, 4] 또는

텍스처 패치(patch)[5, 6]를 사용하는 방법 등이 가능하다. 이중에서 개별 픽셀 단위의 텍스처 합성을 이용한 마커의 은닉이 시도된 바 있다[7]. 하지만 픽셀 기반 합성의 한계로 텍스처의 전역적인 특성을 재현하지 못하고, 초기 합성한 텍스처를 계속 사용하기 때문에 주변 환경의 변화에 유연하지 않다는 문제가 있다.

본 논문에서는 패치 기반의 텍스처 합성을 활용한 마커 은닉 방법을 제안한다. 제안된 방법은 영상에서 마커 영역을 추출하는 전처리 과정과 마커의 대체 영상을 생성하기 위해 템플릿 매칭에 기반한 텍스처 합성 과정으로 구성되며, 이러한 과정들은 영상 프레임의 변화에 따라 선택적으로 최소한의 회수만큼 수행된다. 그 결과 기존의 방법과 유사한 수준의 실시간성을 보장하면서, 배경 텍스처의 전역적인 특성을 유지하고, 주변 환경의 변화에 유연한 마커 은닉을 수행할 수 있다. 또한 별도의 장비가 필요하지 않으며, 배경 영상의 특징점에 의존하지 않는다는 장점을 가진다.

논문의 구성은 다음과 같다. 2장에서는 마커 은닉 알고리즘의 전체적인 과정을 기술하고, 3장에서는 효율적인 알고리즘을 위한 최적화 방안에 대하여 논의한다. 그리고 4장에서 실험 결과에 대한 분석을 한 뒤, 5장에서 결론 및 향후 연구에 대하여 언급한다.

*본 연구는 광주과학기술원 실감콘텐츠연구센터(ICRC)의 지원에 의한 것임

2 텍스처 합성

본 논문에서 제안하는 방법의 전체적인 흐름도는 그림 1과 같이 표현할 수 있다.



그림 1: 전체적인 과정의 흐름

2.1 마커 인식

텍스처를 합성하기 위해서는 카메라를 통해 입력 받은 원본 영상 내에 존재하는 마커의 위치를 구해야 한다. 마커의 인식과 변환행렬 계산 과정에는 ARToolKitPlus[8]를 사용하였다. 이때 구해지는 마커의 크기는 패치를 기준으로 한 것이기 때문에 마커가 이격됨으로써 생기는 그림자나 마커의 테두리 너비 등을 고려하지 않으면 합성된 배경 텍스처가 마커 영역을 제대로 중첩시키지 못 하는 현상이 발생하기도 한다. 따라서 마커의 테두리 폭에 50%의 여유를 주었다.

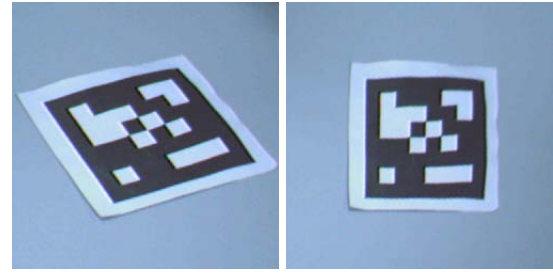
2.2 영상 정정(rectification)

영상 속의 마커는 카메라 공간상의 투영 왜곡(perspective distortion)으로 인해 정사각형에서 사다리꼴 형태로 변형되어 있는 상태이다. 따라서 정확한 결과를 위한 영상의 정정 과정이 수반되어야 한다.

우선 영상 속 마커의 4개 모서리점들이 이루고 있는 사다리꼴의 크기와 비슷한 정사각형의 모서리 좌표들을 구해야 한다. 사다리꼴 각 모서리의 이웃변들을 가로지르는 선분으로 직사각형을 만들고, 이 직사각형의 변 길이를 평균하여 정사각형을 만들어낸다. 이렇게 구해진 모서리 좌표들 간의 변환행렬을 계산하여 투영 변환(perspective warp)을 적용한다. 그림 2는 이러한 과정을 통해 정정된 영상을 보여준다.

2.3 패치 분할

정정된 영상 속에서 마커가 차지하고 있는 영역은 패치 기반 텍스처 합성 알고리즘의 적용을 위해 적절한

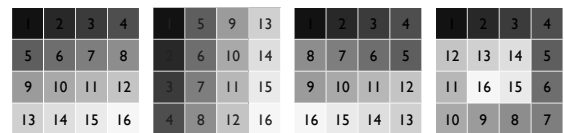


(a) 영상 정정 이전 (b) 영상 정정 이후

그림 2: 마커 영상의 정정 결과

크기의 패치로 분할된다. 이러한 패치의 크기는 텍스처의 전역적인 특성을 재현하는 요소로 작용한다. 패치의 크기가 크면 클 수록 영상에서 함께 복사되는 영역이 커지기 때문에 원본 배경의 전역적인 특성이 재현할 수 있으나, 패치 경계의 불일치 현상이 현저해지는 경향이 발생한다. 패치의 크기가 작으면 패치 경계는 자연스럽게 합성할 수 있으나 텍스처의 전역적인 특성은 표현하지 못 하게 된다. 패치의 크기가 1인 경우는 픽셀 단위의 텍스처 합성을 하는 것과 동일한 의미이다.

텍스처를 합성하는 순서 또한 결과에 영향을 주는 요소이다. 그림 3은 스캔라인(scanline) 기반의 횡대형(horizontal), 종대형(vertical), 교차형(zigzag) 합성 순서와 나선형(spiral) 합성 순서의 4가지 변형을 보여주고 있다. 본 논문에서는 이미지 인페인팅과 같이 제약 조건을 갖는 텍스처 합성에서 주로 사용되는 나선형의 합성 순서를 기본으로 하였다[4].



(a) 횡대형 (b) 종대형 (c) 교차형 (d) 나선형

그림 3: 분할된 패치들의 합성 순서

2.4 템플릿 매칭(template matching)

영상은 그림 4와 같이 알려진 영역과 알려지지 않은 영역으로 구분된다. 알려진 영역은 마커에 가려지지 않은 원본 영상 또는 텍스처 합성에 의해 대체된 패치들을 의미한다. 알려지지 않은 영역은 그 외의 부분으로, 텍스처를 합성해야 하는 빈 패치들의 모음이다. 패치들은 분할 단계에서 정의된 순서에 따라 그림 5와 같은 과정으로 합성된다.

알려지지 않은 영역의 패치는 패치 경계 부분의 영상과 알려진 영역의 영상에 대한 템플릿 매칭 결과 선택된 패치에 의해 채워진다. 채우고자 하는 패치의 좌

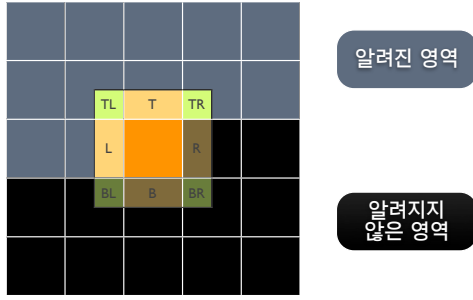


그림 4: 패치의 종류와 경계 구분

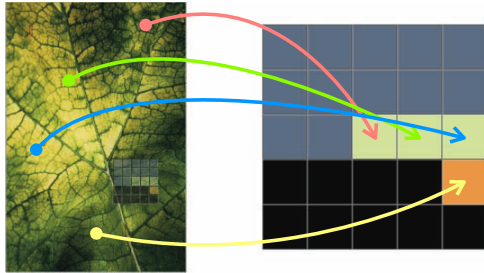


그림 5: 마커 영역의 합성

표가 x 일 때, 해당 패치를 위해 선택되는 패치의 좌표 $p(x)$ 는 수식 1과 같다.

$$p(x) = \operatorname{argmin}_{x' \in \mathbf{X}} \sum_{t \in \mathbf{T}} \frac{d(B_t(x), R(x'))}{|B_t(x)|} \quad (1)$$

\mathbf{X} 는 알려진 영역의 전체 좌표 집합, x' 은 \mathbf{X} 의 한 좌표, $R(x')$ 은 x' 이 형성하는 좌표 집합, \mathbf{T} 는 패치에 인접한 8방향 경계 $\{T, B, L, R, TL, TR, BL, BR\}$, $B_t(x)$ 는 x 가 속한 패치의 경계 부분이 형성하는 좌표 집합이다. $d(x, x')$ 는 패치를 구성하는 픽셀 값에 대한 SSD(Sum of Squared Differences)이며 수식 2와 같다.

$$d(x, x') = \frac{\sum_{i=1}^{|x|} (I(x_i) - I(x'_i))^2}{\sqrt{\sum_{x_i} I(x_i)^2 \sum_{x'_i} I(x'_i)^2}} \quad (2)$$

$I(x)$ 는 픽셀 값의 강도, x_i 와 x'_i 는 각각 좌표 집합 x 와 x' 의 원소이다.

그림 6은 패치 경계에 대한 개별적인 템플릿 매칭 결과와 이들의 합을 보여준다. 밝은 색상일 수록 유사한 매칭을 의미한다.

2.5 영상 페더링(feathering)

이전 단계에서 발견한 영상 패치를 실제로 복사하는 과정에서 원본 영상과의 일관성을 유지하기 위하여 각 패치들의 경계 부분을 혼합해주는 과정이 추가된다. 이를 위해 그림 7의 영상 페더링 기법을 도입하였다. 각

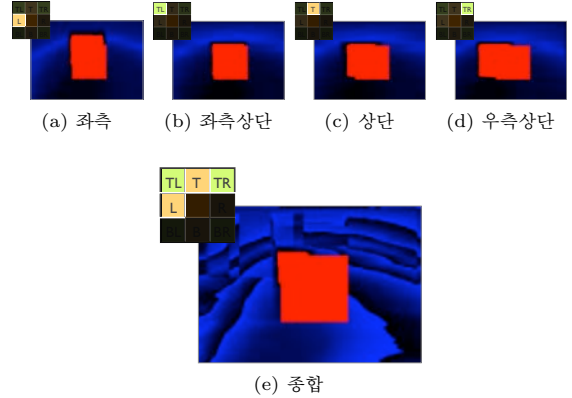


그림 6: 패치 경계에 대한 개별 템플릿 매칭



그림 7: 영상 페더링 기법

패치들의 경계 부분을 연장하여 중첩되는 픽셀들을 위치에 따라 다른 가중치를 사용해 병합함으로써 자연스러운 연결을 만들어낼 수 있다. 페더링된 영상 픽셀의 값 $I_f(x)$ 는 수식 3과 같다.

$$I_f(x) = wI_s(x) + (1-w)I_b(x) \quad (3)$$

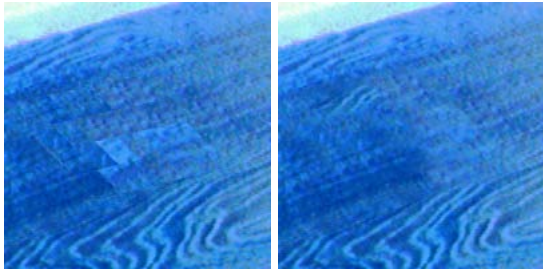
$I_s(x)$ 는 합성된 영상의 픽셀 값, $I_b(x)$ 는 배경 영상의 픽셀 값이며, w 는 합성 영상에 대한 가중치로써, 해당 픽셀이 경계 부분에서 자리하는 위치에 따라 변화한다. 만약 픽셀이 합성된 영상쪽의 경계에 위치하고 있다면 합성된 영상에 대한 가중치는 1이 되어 배경 영상의 픽셀은 고려되지 않게 되며, 배경 영상쪽의 경계에 위치한다면 가중치는 0이 되어 정반대의 결과를 나타낸다. 이들의 중간에 위치한 픽셀은 경계 부분의 전체 길이에 비례하여 값이 결정된다.

그림 8에서는 페더링의 적용 유무에 따른 결과를 비교하고 있다. 페더링이 적용된 영상은 패치 경계 부분이 부드러워졌다는 사실을 알 수 있다. 이러한 페더링 기법은 텍스처 합성을 위한 패치들이 모두 결정된 이후에 이루어지는 것이 아니라 각각의 템플릿 매칭 이후 개별적으로 적용되기 때문에 전체적으로 자연스러운 합성 결과를 유도한다.

3 최적화 기법

3.1 축소 템플릿 매칭

텍스처 합성을 위한 대부분의 연산은 템플릿 매칭에서 소요된다. 템플릿 매칭의 속도를 좌우하는 요소에는

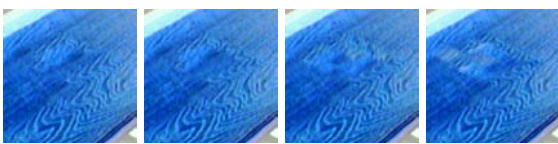


(a) 페더링 미적용 (b) 페더링 적용

그림 8: 페더링 적용에 따른 결과

입력 받은 영상의 크기, 분할되는 패치의 크기와 개수 등이 있다. 입력 영상의 크기는 템플릿 매칭을 해야 하는 영역의 범위를 정의하며, 패치의 크기는 경계 부분을 결정함으로써 템플릿의 크기에 영향을 미친다. 패치의 개수는 템플릿 매칭의 반복 회수를 결정한다.

템플릿 매칭 과정을 통해 선택되는 영상 패치는 인간 시각의 불완전성을 이용하여 주변 영상과 이질감 없이 합몰되는 것을 목적으로 하기 때문에 반드시 정확한 계산 결과에 의존해야 하는 것은 아니다. 따라서 입력 영상을 일정 비율로 축소하여 사용하여도 기존과 유사한 패치들을 찾아낼 수 있다. 이렇게 찾아진 패치의 좌표는 실제 패치가 복사될 때에는 원래의 비율로 조정되어 원본 영상의 패치를 참조하기 때문에 텍스처 합성의 결과 자체에 현격한 품질 저하가 발생하지는 않는다. 그림 9는 비율에 따른 축소 템플릿 매칭의 결과이다. 영상을 축소하여도 결과의 품질에는 큰 차이가 없다는 사실을 알 수 있다.



(a) 1 (b) 1/2 (c) 1/4 (d) 1/8

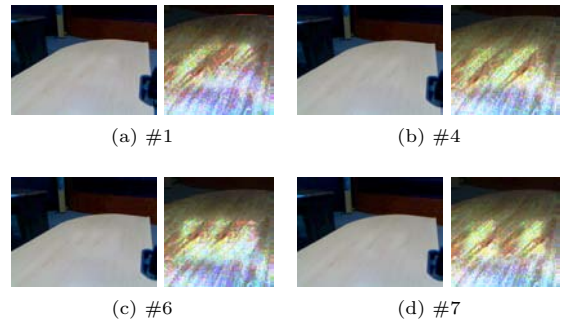
그림 9: 축소 템플릿 매칭 결과

텍스처 합성을 위해 선택되는 패치들은 주로 마커 주변의 영상에서 발견된다는 점에 착안하여 입력 영상을 일정한 크기로 잘라내는 방법도 적용하였다. 마커에서 멀리 떨어진 곳에 위치한 영상은 마커와 동일한 평면상에 위치하지 않을 가능성이 상대적으로 높기 때문에 불필요한 계산을 줄인다. 또한 관련이 없는 패치가 우연히 선택되어 합성을 방해하는 사태를 방지할 수 있다.

3.2 텍스처 합성 빈도

영상 프레임별 텍스처 합성 빈도에 따라 마커 은닉 결과가 달라진다. 한 장의 영상만을 합성하여 카메라의

시점에 맞게 변환시키는 방법은 연산이 적어 효율적이지만, 조명 등의 주변 환경이 변화하면 배경의 이질감을 유발할 수 있다[7]. 이를 극복하기 위해 매 프레임마다 합성을 시도하는 경우 개별 영상에서는 자연스러운 결과를 얻을 수 있다. 하지만 프레임들 간의 연관성이 없기 때문에 연속적인 영상의 관찰시에는 그림 10과 같이 합성 부위가 점멸하는 듯한 현상이 발견된다. 또한 인접 프레임들 간의 장면 변화가 크지 않은 경우 모든 프레임을 합성하는 것은 비효율적이다.



(a) #1

(b) #4

(c) #6

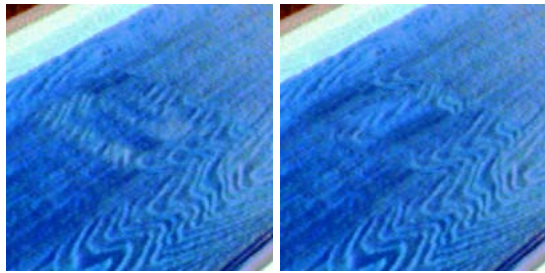
(d) #7

그림 10: 매 프레임 합성 결과: 좌측은 마커가 은닉된 영상. 우측은 은닉된 부분을 확대하여 색상 차이를 강조한 영상. 합성된 텍스처의 패턴이 서로 다르다(숫자는 프레임 번호).

따라서 새로운 합성이 필요할 때만 수행하는 선택적 방법을 취하였다. 합성하려는 패치의 개수가 이전 프레임과 같다면 템플릿 매칭 과정을 생략하고 이전에 패치와의 상대적인 좌표만을 재계산한다. 동일한 좌표를 사용하지만 현재 영상의 텍스처로 합성하기 때문에 주변 환경의 변화에 강건한 결과를 얻는다. 패치의 개수가 달라지는 경우에는 기존의 합성 정보를 적용할 수 없어 새로운 합성이 불가피하다.

3.3 색상 모델 변환

컬러 영상에서 템플릿 매칭을 하기 위해서는 기본적으로 각 채널별로 독립적인 매칭 과정을 수행한 후 그 결과를 정합해야 한다. 기본적으로 RGB 모델에서 해당 픽셀의 각 채널 평균값을 사용하는 방법이 있다. 하지만 RGB 모델의 색상값은 인간이 인지하는 강도와 상이하기 때문에 합성된 텍스처 패치들의 시각적 불일치 현상이 두드러질 수 있다. 따라서 인간의 시각적 인지에 적합한 HSV 모델로 변환한 영상을 사용하였다. 그림 11은 색상 모델에 따라 달라진 합성 결과를 보여준다. HSV 모델을 사용하였을 때 마커의 경계 부분이 더 자연스럽게 합성되었다는 사실을 알 수 있다.



(a) RGB 모델 (b) HSV 모델
그림 11: 색상 모델에 따른 합성 결과

4 실험 결과 및 분석

4.1 실시간성 실험

그림 12는 영상에서 인식되는 마커의 크기, 즉 합성되는 텍스처의 크기 변화에 따른 프레임 레이트를 보여준다. 640×480 픽셀의 15fps 입력 영상에서 24×24 크기의 패치를 기준으로 나선형 순서의 합성을 하였으며, 펜티엄 3.4G 시스템에서 Point Grey Research사의 Dragonfly 카메라를 사용하였다. 13번 프레임에서 마커를 인식하였고, 40번 프레임부터 마커 은닉 알고리즘을 적용하였다. 이에 따라 영상의 프레임 레이트가 평균 7fps 정도로 하락하였다. 최초로 인식한 마커의 크기는 78×78 픽셀이었으며 이를 합성하기 위해 약 0.2초가 걸려서 5fps까지 성능이 낮아진 것을 볼 수 있다. 텍스처의 크기가 증가함에 따라 이러한 합성 시간도 함께 증가하여 97번 프레임의 96×96 픽셀 텍스처 합성에는 약 0.3초가 걸려 3fps, 154번 프레임의 174×174 픽셀 텍스처 합성에는 약 1.3초가 걸려 0.7fps의 성능을 보여주었다. 이는 기존의 실시간 마커 은닉 연구에서 64×64 크기의 마커 은닉에 0.2초, 128×128 크기에 0.8초가 걸린 것과 유사한 성능이다[7]. 실제 환경에서 영상 속의 마커 크기가 급격하게 변하는 경우는 많지 않기 때문에 이러한 조건을 잘 고려한다면 충분히 실시간 적용 가능하다.

4.2 텍스처 특성 실험

그림 13은 다양한 배경에서의 마커 은닉 전후 결과를 보여준다. 640×480 픽셀의 입력 영상에서 16×16 크기의 패치를 기준으로 텍스처 외곽에서 내부 방향으로의 나선형 순서로 합성하였다. 주변 배경이 복잡한 경우에도 균일한 텍스처 패턴이 존재한다면 실제와 유사한 영상이 합성되어 마커를 은닉하였다. 즉, 배경 텍스처의 전역적인 특성이 유지되었다는 것을 알 수 있다.

4.3 주변 환경 변화 실험

그림 14는 기존 정보를 활용한 선택적 합성의 결과를 보여준다. 초기 환경에서 합성한 영상으로 마커를 은닉

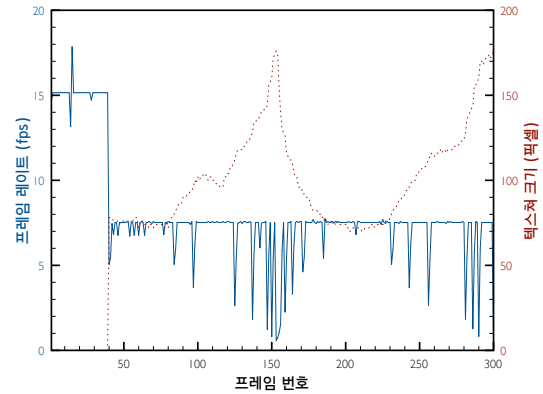


그림 12: 텍스처 크기에 따른 프레임 레이트 변화

하고 있으며, 조명 조건이 변한 이후에도 동일한 정보를 이용하고 있다. 조명의 강도가 변화하였지만 현재 영상의 패치를 이용해 합성하였기 때문에 변화된 조명 환경과 조화를 이루었다. 만약 초기 환경에서 합성한 영상을 그대로 사용한다면 텍스처 색상의 불일치로 인하여 마커 영역의 경계가 선명하게 부각되어 버린다.

4.4 가상객체 증강 실험

그림 15는 은닉된 마커 위에 적절한 가상 객체를 증강함으로써 사용자에게 새로운 경험을 제공하는 실험이다. 은닉되지 않은 마커가 부각됨으로써 사용성을 저하시키는 사례를 위하여 작은 크기의 가상객체들이 다수 등장하는 입자 시스템(particle system)을 선택하였다. 특히 마커를 통해 계산한 구조 정보를 활용하여 가상 입자들이 탁자와 충돌, 반사되는 장면을 연출함으로써 가상과 실체가 혼합되었다는 점을 강조하였다.

320×240 픽셀의 15fps 입력 영상에서 16×16 크기의 패치를 기준으로 나선형 순서의 합성을 하였으며, 펜티엄 1.7G 시스템에서 약 10fps의 성능을 보였다. 총 5000개의 입자가 매 프레임마다 10개씩 등장하여 200 프레임 동안 존재 후 소멸되었다. 운동 벡터에 임의성을 부여하고 중력이 가정되었으며 평면과의 충돌이 구현되었다.

5 결론 및 향후 연구

본 논문에서는 자연스러운 증강현실을 위하여 패치 기반의 텍스처 합성을 통한 마커 은닉 방법을 제안하였다. 실험 결과에서는 제안된 방법이 실시간성을 보장하면서, 배경 텍스처의 전역적인 특성을 유지하고, 주변 환경의 변화에 유연하다는 것을 확인하였다.

이러한 마커 은닉 방법은 증강현실 기반 햅틱 상호작용 시스템에 적용될 수 있다. 현재 개발되어 있는 시스템은 가상 객체와 햅틱 장치의 좌표계를 일치시키기 위하여 그림 16과 같이 다수의 마커가 부착된 평면을



그림 13: 다양한 배경 텍스처에서의 마커 은닉



그림 14: 주변 환경 변화에 유연한 마커 은닉

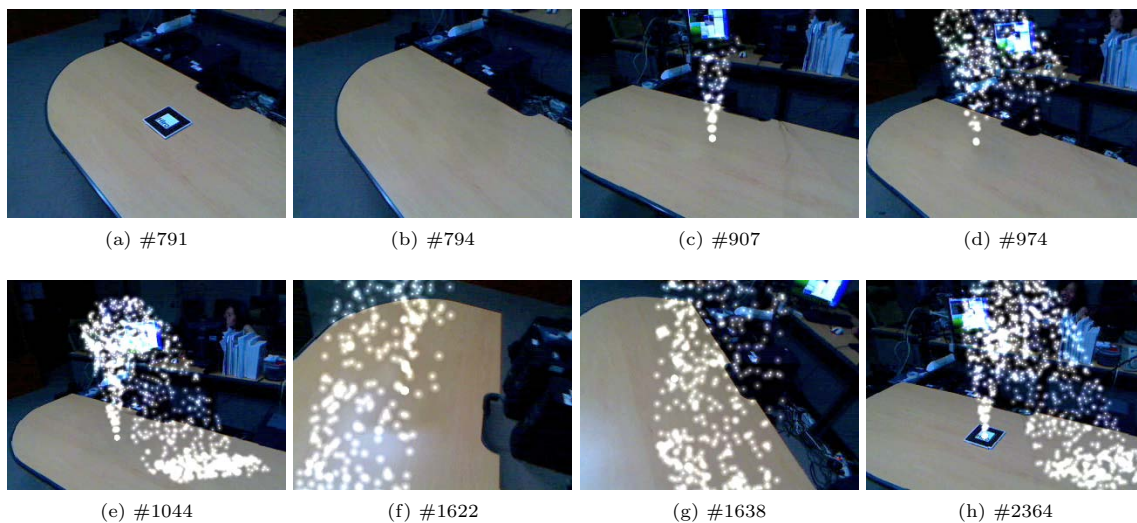


그림 15: 연속적인 영상에서의 마커 은닉과 가상 객체 증강(숫자는 프레임 번호)

사용하고 있다. 이러한 마커 평면을 은닉한다면 실감성 증대를 통한 사용성 향상을 기대할 수 있을 것이다.

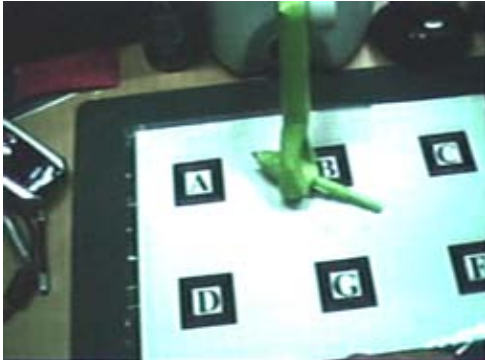


그림 16: 햅틱 장치의 좌표계 일치를 위한 마커

알고리즘을 개선하기 위해서는 입력 영상의 구조에 따른 세그멘테이션을 통해 템플릿 매칭의 부하를 줄이고 정확도를 높이는 방안을 생각해볼 수 있다. 또한 기존의 텍스처 합성 정보를 효과적으로 저장할 수 있는 자료 구조를 설계하여 합성시의 영상 구조나 조명 상태 등의 주변 정보를 통합하여 사용하는 방안을 고려해볼 수 있다. 텍스처들의 품질을 객관적으로 비교할 수 있다면 합성시 기준 자료로 활용되고, 향후 사용성 평가에도 도움이 될 것이다.

참고 문헌

- [1] H. Park and J. I. Park, "Invisible marker tracking for AR," *Proc. ISMAR*, 2004, pp. 272–273.
- [2] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," *Proc. SIGGRAPH*, 2000, pp. 417–424.
- [3] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," *Proc. ICCV*, 1999, p. 1033.
- [4] L. Y. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," *Proc. SIGGRAPH*, 2000, pp. 479–488.
- [5] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," *Proc. SIGGRAPH*, 2001, pp. 341–346.
- [6] L. Liang, C. Liu, Y. Q. Xu, B. Guo, and H. Y. Shum, "Real-time texture synthesis by patch-

based sampling," *ACM Trans. Graph.*, vol. 20, no. 3, pp. 127–150, 2001.

- [7] S. Siltanen, "Texture generation over the marker area," in *Proc. ISMAR*, 2006, pp. 253–254.
- [8] Handheld Augmented Reality, <http://studierstube.icg.tu-graz.ac.at/handheld.ar/artoolkitplus.php>