

# AR을 이용한 타투 시뮬레이션

한채운 김유진 박태정

덕성여자대학교

chyun414@naver.com, superkingyj@naver.com, tjpark@duksung.ac.kr

## Tattoo Simulation Using AR

Chaeyun Han Yujin Kim Taejung Park

Duksung Women's University

### 요약

본 논문에서는 타투 한 모습을 AR을 통해 미리 육안으로 확인할 수 있도록 하는 기술적인 방법을 제공함으로써 타투 시술을 실제로 시행하기 전 개인별로 선택을 지원하는 기술을 제안한다. 이러한 목표를 달성하기 위해 Unity Vuforia Engine을 이용하여 서버에서 미리 제공하는 타투 도안 혹은 사용자가 원하는 타투 도안을 AR로 신체에 적용할 수 있도록 지원한다. 제안하는 기술에서는 AR로 인한 시각적인 이질감을 줄이기 위해서 Region Capture를 이용하여 타겟 이미지를 주변 피부와 이질감을 감소시키고 그 위에 타투 AR을 사용자 신체 굴곡에 맞춰 적용하는 방법을 제안한다. 또한 AR로 확인한 타투를 사진이나 동영상으로 촬영하여 앱 내 갤러리에서 확인할 수 있고 앱 내 커뮤니티를 통해 타투에 관심있는 사람이라면 타투에 관한 자유로운 소통을 할 수 있도록 지원하는 온라인 기술을 구현하였다.

### 1. 서론

최근 타투는 나이와 성별에 관계없이 자신의 개성과 신념을 표현하는 하나의 문화로 자리 잡았다. 하지만 타투는 한 번 시술하면 되돌리기 어렵기 때문에 디자인이나 위치 선정 등 개인의 선택에 신중함을 요한다. 본 연구는 타투 시술 전 IT 기술을 통해 시술 후의 결과를 미리 시각적으로 확인할 수 있다면 여러 부작용들을 줄일 수 있고 개인의 선택에도 확신을 줄 수 있을 것이라는 아이디어에서 출발하였다.

이에 따라 본 연구에서는 서버에서 제공하는 타투 도안 혹은 사용자가 원하는 타투 도안을 신체에 AR로 적용하고 그것을 사진이나 동영상으로 촬영하여 확인하고 더 나아가 커뮤니티를 통해 타투에 대한 자유로운 소통을 할 수 있는 애플리케이션을 제시한다. 제안하는 방식은 타투 시술로 인한 부작용을 최소화할 수 있다는 점에서 타투 소비자, 타투이스트 등을 포함해서 현재 타투 산업에 종사하거나 관심이 있는 모든 사람들에게 도움이 될 것으로 생각한다.

### 2. 본론

AR을 구현하는 데에 있어서는 ARKit, ARCore, 그리고 그 둘을 통합한 Unity AR Foundation, Unity Vuforia 등의 SDK가 존재한다. 원하는 타겟 이미지를 인식하고 그 위에 AR 구현하기 위해 본 연구에서는 Unity Vuforia Engine을 사용하였다.

본 연구에서 타투를 적용하는 데에 있어서 가장 집중할 신체 부위는 손목, 팔뚝과 같은 원기둥(Cylinder) 형태의 신체 부위이다. 사용자 개인마다 다른 굵기와 굴곡을 가지고 있는 원기둥(Cylinder) 형태의 신

체 부위에 있어 타투 AR이 얼마나 자연스럽게 적용되느냐에 따라 연구의 완성도가 결정되기 때문이다. 따라서 신체에 타투 AR이 자연스럽게 둘러 지게 하기 위하여 원기둥(Cylinder) Object를 Plane 형태의 Target Image 위 최대한 가깝게 위치시켰다.

본 논문에서 제안하는 기술의 핵심적인 요소는 AR 카메라가 Target Image를 추적한 후 추적한 표면 위에 타투 AR을 오버레이 형태로 구성하는 부분이다. 사용자의 신체에 타투 AR이 위치하기 위해서는 Target Image가 반드시 사용자의 신체 위에 있어야 한다. 따라서 Target Image는 사용자의 신체에 실제로 부착할 수 있도록 스티커의 시각적인 정보를 출력에 그대로 반영할 수 있어야 한다. 그리고 신체에 부착하여 Target Image로 사용되는 스티커는 주변 피부와 이질감 없이 가려져야 그 위에 올라간 타투 AR이 더 자연스럽게 보인다라는 제한이 존재한다. 본 논문에서 제안하는 방법에서는 Region Capture[1]를 사용하여 이러한 문제를 해결하였다. Target Image 바로 아래의 신체에 Target Image와 크기가 같은 Plane을 두고 그 Plane의 Region을 카메라 실시간 Capture를 진행한다. 그리고 Target Image 위에 Plane을 하나 더 위치시켜 Region Capture로 따온 피부 표면을 입혀준다. 제안하는 방법은 조명과 그림자에 영향을 많이 받는다는 단점이 존재하지만 Target Image를 효과적으로 가릴 수 있다는 장점이 있다.

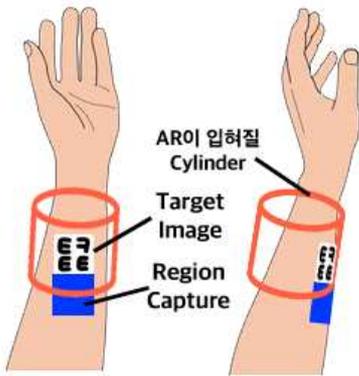


그림 1. Target Image, Cylinder Model, Region Capture의 위치

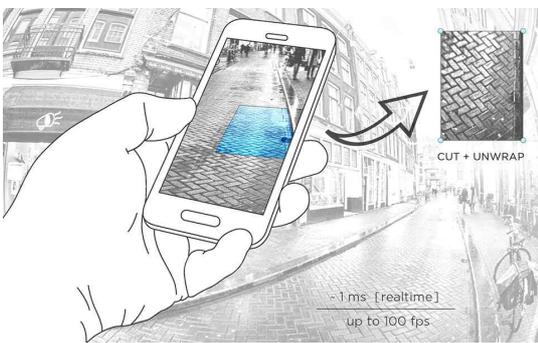


그림 2. Region Capture



그림 3. 신체에 Target Image를 부착한 모습

그림 4. Region Capture와 Vuforia를 통해 AR을 적용한 모습

본 논문에서 제안하는 기법을 실현하기 위해 Unity Vuforia와 Region Capture를 통해 AR 타투 시뮬레이션을 구현하였으며 구현한 애플리케이션은 <도안 선택>, <도안 업로드>, <갤러리>, <커뮤니티>의 네 가지 기능으로 나뉜다. 우선, <도안 선택>에서 사용자는 Django를 통해 구축한 서버를 통해 사전에 미리 업로드한 타투 도안들 중 하나를 선택할 수 있다. <도안 업로드>에서는 Unity Native Gallery Asset[2]을 사용하여 사용자의 디바이스에 직접적으로 접근하여 사용자가 원하는 타투 도안을 업로드할 수 있다. 도안 선택이나 업로드 후 AR로 확인한 타투의 모습을 사진이나 동영상으로 촬영하고 <갤러리>에서 확인할

수 있다. 마지막으로 <커뮤니티>는 js, html, css를 Front-end로 Django를 Back-end로 하여 회원가입, 로그인, 마이페이지, 게시물 및 댓글 작성, 좋아요, 구독 등의 서비스를 갖추고 있다.



그림 5. 메뉴 선택 화면 (왼쪽)

그림 6. <도안 선택> 메뉴를 통해 서버에서 가져온 타투 도안들 중 하나를 선택

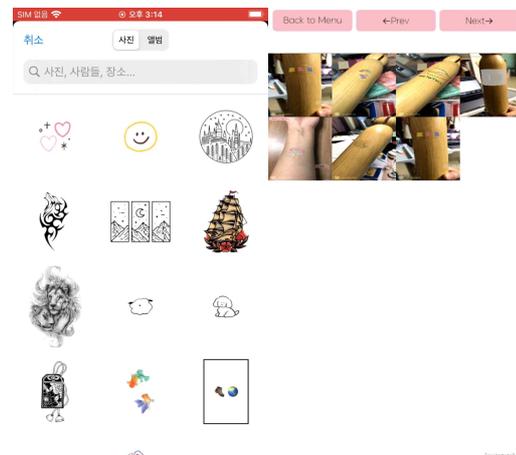


그림 7. <도안 업로드> 메뉴를 통한 디바이스 내 갤러리 접근

그림 8. <갤러리>를 통해 AR 화면에서 찍은 사진 및 동영상 확인

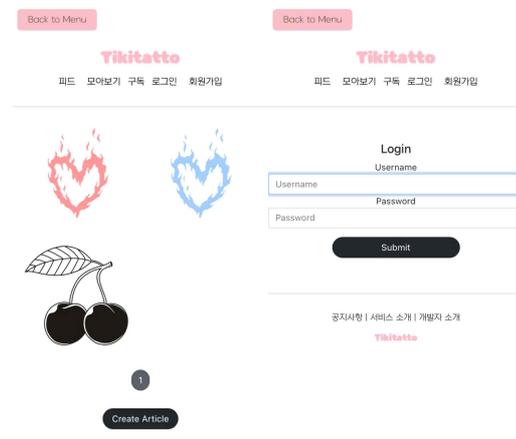


그림 9. <커뮤니티> 메인 피드 화면

그림 10. <커뮤니티> 로그인 화면

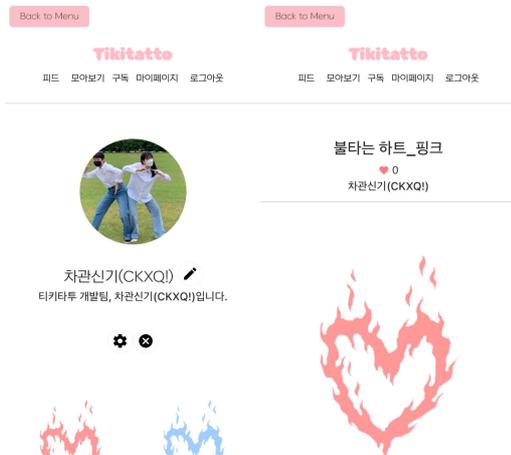


그림 11. <커뮤니티> 마이페이지 화면

그림 12. <커뮤니티> 게시글 상세 화면

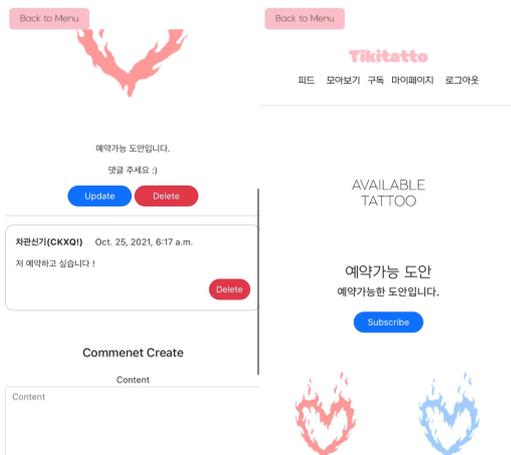


그림 13. <커뮤니티> 게시글에 댓글 작성 가능

그림 14. <커뮤니티>의 모아보기를 통해 관심있는 프로젝트 구독 가능

### 3. 결론

본 연구는 AR을 이용한 타투 시뮬레이션 애플리케이션을 제작하여 사용자가 미리 자신이 원하는 타투 도안을 신체에 AR로 적용하여 확인함으로써 실제 타투를 자신의 몸에 적용하기 전에 사전에 자신의 선택이 실제로 적용되었을 때의 결과를 시각적으로 미리 검토할 수 있도록 함으로써 신중하게 결정할 수 있도록 지원한다.

본 논문에서 제안하는 기술은 Unity Vuforia를 이용하여 구현되었으며, 논문에서 서술한 기능적인 장점에도 불구하고 타투 AR을 적용하였을 때 타투 AR이 타겟 이미지 위에서 부자연스러운 동작을 보인다는 점, Region Capture가 조명의 영향을 많이 받는다는 점, 사용자 개인의 신체 두께나 굴곡에 맞추지 않고 하나의 획일된 사이즈로 타투 AR을 적용한다는 기술적인 한계가 존재한다.

그러나 본 논문에서 제안하는 기술은 타투 산업에서의 부작용을 감소시킴으로써 개인의 만족을 극대화하기 위한 의미가 있다고 판단하며, 향후 앞서 논의한 제약 사항들의 극복을 위해 지속적인 연구를 수행

하고자 한다.

### 감사의 글

이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2021R1A4A5028907).

### 참고 문헌

- [1] maximrouf/RegionCapture, <https://github.com/maximrouf/RegionCapture> (Accessed Nov. 22, 2019)
- [2] yasirkula/UnityNativeGallery, <https://github.com/yasirkula/UnityNativeGallery> (Accessed Oct. 29, 2021)