# 모바일 증강현실을 위한 3차원 모델기반 카메라 추적

박정식 서병국 박종일 *

한양대학교

{nangsik, bkseo}@mr.hanyang.ac.kr, jipark@hanyang.ac.kr

# 3-D Model-Based Tracking for Mobile Augmented Reality

Park, Jungsik Seo, Byung-Kuk Park, Jong-Il

Hanyang University

## 요약

본 논문에서는 모바일 증강현실을 실현하기 위한 3차원 모델기반 카메라 추적 기술을 제안한다. 3차원 모델기반 추적 기술은 평면적이지 않은 객체에 적용 가능하며, 특히 텍스처가 없는 환경에서 유용하다. 제안하는 방식은 대상 객체의 3차원 모델 정보로부터 영상에서 추출한 에지와의 대응점을 찾고, 대응점의 거리를 최소화하는 카메라 움직임을 추정함으로써 이전 카메라 포즈(위치 및 방향)로부터 현재 포즈가 추적되는 방식이다. 안드로이드 플랫폼의 스마트폰 상에서 제안된 방식으로 카메라 포즈를 추적하고 3차원 가상 콘텐츠를 증강시켜 봄으로써 그 유용성을 확인한다.

## 1. 서론

최근 높은 연산 처리 속도를 갖는 하드웨어와 카메라가 탑재된 모바일 기기의 등장으로 모바일 증강현실을 위한 비전기반 카메라 추적 방법에 대한 관심이 높아지고 있다.

시각적 마커를 이용한 카메라 추적 방법은 그 간단함과 강건함으로 일찍부터 모바일 증강현실에 사용되어 왔다[1]. 하지만 인위적인 마커로 인해 발생되는 부자연스러운 환경은 사용자의 몰입감을 방해한다는 단점이 있다. 따라서 이러한 문제점을 해결하고 보다 자연스러운 일상 환경에서 몰입감 높은 증강현실을 제공하기 위해서 추적 환경에 존재하는 시각적 특이점, 예를 들어 코너나 에지 등을 이용하는 카메라 추적 방법에 대한 연구가 활발히 진행되어 왔다. 특히 모바일 기기에서는 이러한 카메라 추적 방법이 많은 연산량과 메모리 공간을 필요로 하기 때문에 모바일 기기의 제한적인 자원을 효율적으로 사용하여 실시간으로 카메라 포즈를 추적하기 위한 다양한 방법이 제안되어 왔다[2, 3]. 그 대표적인 예로, Wagner 등은 모바일 기기에서 빠르고 강건한 특징점기반 추적 방법을 제안하고 다양한 모바일 기기를 이용하여 이를 검증함으로써 모바일 플랫폼 상의 실시간 추적 기술에 대한 가능성을 보여주었다[4]. 또한, 이러한 추적 방법은 주로 평면으로 구성된 추적 환경의 특징점을 이용하는데 반해, 추적 환경의 3차원 정보를 이용하는 추적 방법도 제안되어왔다. 예를 들어, Klein 등은 3차원 특징점 지도를 생성하여 카메라 포즈를 추적하는 PTAM(Parallel Tracking and Mapping)을 제안하고, iPhone을 이용하여 구현, 검증하였다[5].

일반적으로 특징점기반 추적 방법은 정확하고 강건한 카메라 추적을 위해서 텍스처가 많은 추적 환경을 필요로 한다. 따라서 이러한 추적 방법은 다양한 추적 환경이 존재하는 일상 환경, 특히 텍스처가 적은 객체 또는 환경에 대응하기가 어렵다. 이에 3차원 모델기반 추적 방법은 추적 환경의 3차원 객체 정보를 이용하여 카메라 포즈를 추적하기 때문에, 평면으로 구성되어 있지 않거나 텍스처가 적은 환경에서 보다 효과적인 카메라 추적이 가능하다. 3차원 모델기반 추적 방법은 컴퓨터 비전 및 로봇 분야에서 오랫동안 연구되어 왔으며, 최근에는 모바일 플랫폼 상에서 이를 구현하려는 노력도 있어왔다[6].

본 논문에서는 모바일 플랫폼의 상에서 실시간 카메라 추적을 위한 3차원 모델기반 추적 방법을 제안하고, 안드로이드 플랫폼 기반의 스마트폰을 이용해 구현, 검증함으로써 그 유용함을 확인한다.

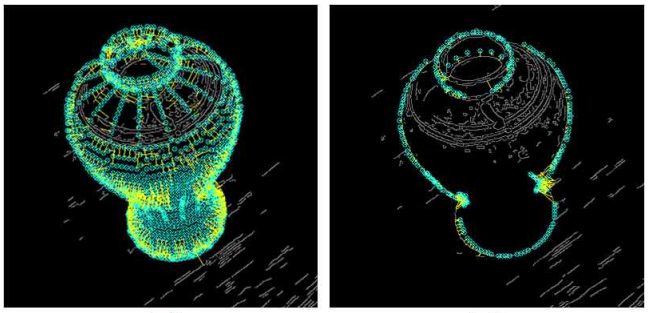
## 2. 방법

3차원 모델 정보를 이용하여 카메라 포즈를 추정하기 위해 먼저 이전 카메라 포즈로 투영된 3차원 모델을 영상에서 추출한 에지와 정합하여 대응점을 구한다. 그리고 구해진 대응점간의 거리를 최소화하도록 반복적으로 카메라 움직임을 계산함으로써 현재 카메라 포즈를 추정한다. 이러한 과정의 세부적인 내용은 이어지는 소단락에서 기술한다.

### 가. 3차원 모델의 정합

카메라 포즈 추정을 위해 사용되는 3차원 객체 모델은 와이어 프레임으로 모델링되고, 가시성 검사를 통해 이전 카메라 포즈에서 보이는 부분의 윤곽선에 해당하는 선만 추출하여 사용한다. 가시성 검사에서는 카메라 시점 벡터와 선의 법선 벡터의 내적의 부호로 가려지는

* 교신저자



(가) (나)

그림 1. (가) 가시성 검사 전, (나) 가시성 검사 후.

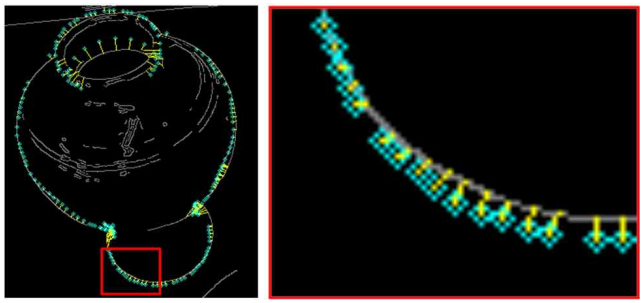


그림 2. 영상 평면에 투영된 3차원 모델의 선과 대응점(노란색은 선의 법선 방향을 나타낸다).

선을 판별하여 제거한 뒤, 두 면에 의해 공유되지 않는 선만을 외곽선으로 간주하여 추출한다. 추출된 선은 영상 평면으로 투영되어 일정 간격으로 샘플링되고 각 선의 법선 방향으로 가장 가까운 에지를 찾아 이를 대응점으로 정의한다. 여기서, 에지는 캐니에지 검출기(Canny edge detector)를 사용하여 영상으로부터 추출한다.

## 나. 카메라 포즈 추정

카메라 포즈 $\mathbf{E}$는 카메라 움직임 $\Delta\mathbf{E}$에 의해 이전 카메라 포즈로부터 갱신된다($\mathbf{E}_t = \mathbf{E}_{t-1}\Delta\mathbf{E}$). 카메라 움직임 $\Delta\mathbf{E}$는 동차 좌표계(homogeneous coordinate) 상의 3차원 강체 변환(rigid-body transformation)의 표현인 리 그룹(Lie Group) SE(3)로 표현된다[7].

$$\Delta\mathbf{E} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} = \sum_{i=0}^{5} e^{\alpha_i \mathbf{G}_i} \qquad (1)$$

계산의 효율성을 위해 이를 근사하면 다음과 같이 간단하게 표현된다.

$$\Delta\mathbf{E} \approx \mathbf{I} + \sum_{i=0}^{5} \alpha_i \mathbf{G}_i = \begin{bmatrix} 1 & -\alpha_5 & \alpha_4 & \alpha_0 \\ \alpha_5 & 1 & -\alpha_3 & \alpha_1 \\ -\alpha_4 & \alpha_3 & 1 & \alpha_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (2)$$

여기서 $\mathbf{G}_i$는 동차 좌표계 상의 x, y, z 방향에 대한 작은 평행 이동 및 회전 변환의 기저 행렬이고, $a_i$는 각 변환에 대한 변화량을 나타낸다. $a_i$는 대응점 간의 거리를 최소화 하도록 다음과 같이 계산된다.

$$\alpha = \arg\min_{\alpha_i} \left\{ \sum_{\Psi}^{N_e} \eta_e \left\| d^{\Psi} - \sum_{i=0}^{5} \alpha_i \left( \mathbf{n}_{\perp}^{\Psi} \cdot \mathbf{l}_i^{\Psi} \right) \right\|^2 \right\} \qquad (3)$$

여기서 $d$는 이전 카메라 포즈 $\mathbf{E}_{t-1}$로 투영시킨 모델의 선과 이에 대응되는 영상의 에지간의 거리, $\mathbf{n}_{\perp}$는 투영된 선의 법선 벡터, $N_e$는 대응점의 개수, $\eta_e$는 에지 정보의 신뢰도이다. $\mathbf{l}_i$는 투영행렬 $\mathbf{P}$로 투영된 선의 작은 카메라 움직임에 의한 변화량 벡터로 다음과 같이 계산된다.

$$\mathbf{l}_i = \left[ \left(\frac{u}{w}\right)' \quad \left(\frac{v}{w}\right)' \right]^T = \left[ \frac{u'}{w} - \frac{uw'}{w^2} \quad \frac{v'}{w} - \frac{vw'}{w^2} \right]^T \qquad (4)$$

$$\begin{bmatrix} u' & v' & w' \end{bmatrix}^T = \mathbf{P}\mathbf{G}_i \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T \qquad (5)$$

최소자승법으로 계산된 $a$로부터 카메라 움직임 $\Delta\mathbf{E}$이 계산되고 현재 프레임의 카메라 포즈 $\mathbf{E}_t$가 갱신된다. 카메라 포즈 $\mathbf{E}_t$의 갱신은 반복적으로 수행되는데, 재투영 오차가 일정 값 이하로 작아지거나, 미리 정해진 반복 횟수를 넘게 되면 카메라 포즈 $\mathbf{E}_t$의 갱신이 종료된다.

# 3. 실험 및 결과

## 가. 구현 환경

안드로이드 플랫폼 개발언어는 기본적으로 Java이지만 연산 속도가 C/C++에 비해 느리다. 예를 들어 Samsung Galaxy S 스마트폰에서 640×480 크기의 YUV420 형식의 영상을 RGB565 형식으로 변환하는 경우 Java 로 작성할 경우 약 50 ms가 소요되는 반면, C/C++ 언어로 작성할 경우는 약 16 ms가 소요된다. 따라서 카메라 영상 입력 및 UI(User Interface) 이외의 부분은 처리속도를 위해 C/C++ 언어 함수를 JNI(Java Native Interface)를 통해 호출하는 방식으로 구현하였다.

카메라 추적 처리를 위한 그레이스케일 영상은 YUV420 형식으로 입력되는 카메라 영상의 Y 채널만 추출하여 사용하였고, 카메라 프리뷰 렌더링을 위한 영상은 YUV420 형식에서 RGB565 형식으로 변환하였다. 그래픽 렌더링은 OpenGL ES 1.1을 이용하였고, 3차원 캐릭터 애니메이션 렌더링을 위해 Cal3D 라이브러리를 사용하였다.

표 1. 실험에 사용된 모바일 기기

| 사양 | Samsung Galaxy S | LG Optimus 2X | Samsung Galaxy S2 |
|---|---|---|---|
| CPU | ARM Cortex A8 single-core 1 GHz | ARM Cortex A9 dual-core 1 GHz | ARM Cortex A9 dual-core 1.2 GHz |
| GPU | ImgTec. PowerVR SGX 540 | NVidia ultra low-power GeForce | ARM Mali 400 |
| RAM | 512 MB | 512 MB | 1 GB |
| OS | Android OS 2.2 | Android OS 2.2 | Android OS 2.3 |

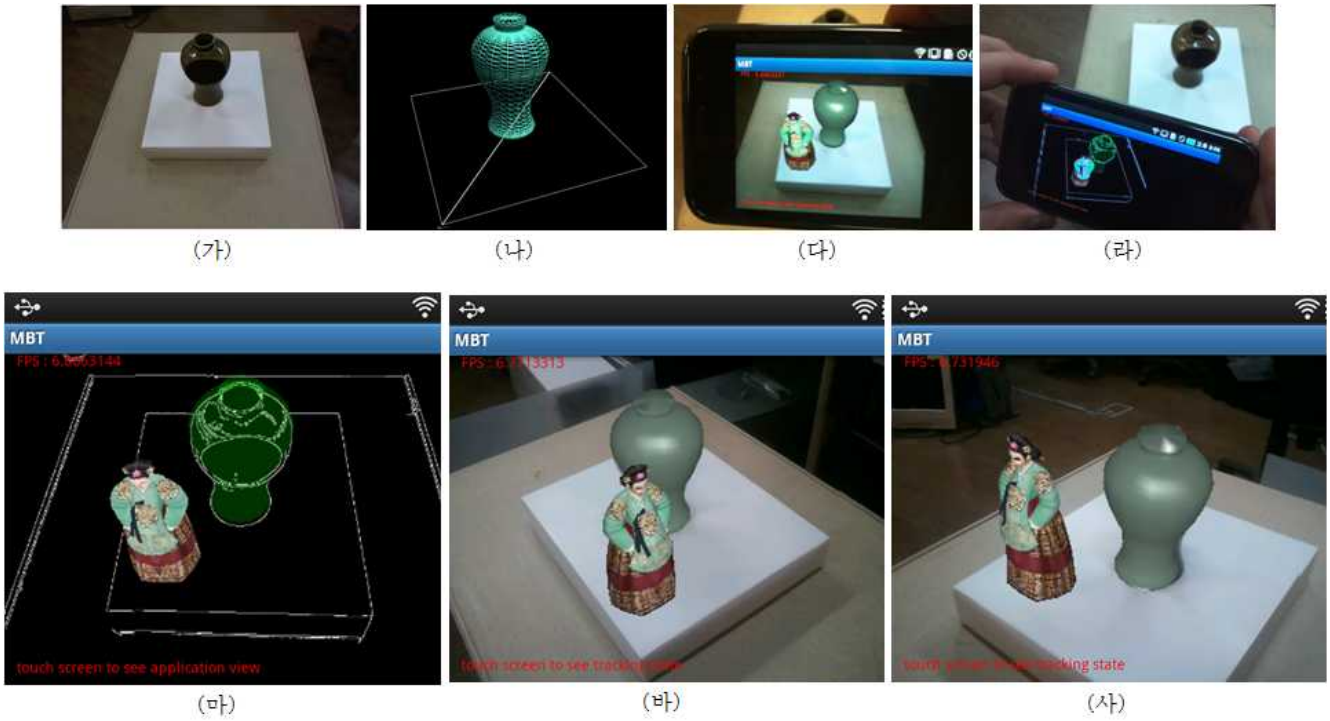



그림 3. 실험 환경 및 결과 영상. (가) 실험 환경, (나) 추적에 사용된 모델, (다, 라) 모바일 기기 (Galaxy S) 화면에서의 결과, (마) 추적에 사용된 추적 환경의 에지 정보, (바, 사) 제안된 방법으로 카메라 포즈를 추적하고 가상 캐릭터를 증강시킨 결과.

## 나. 실험 결과

그림 3-(가) 와 같은 특징점기반 추적 방법을 적용하기 어려운 환경에서 제안된 방식을 적용하여 카메라 포즈를 추적하고 가상 콘텐츠를 증강시켰다. 텍스처가 거의 없고 정반사가 강하게 나타나는 표면을 가진 물체를 추적에 사용하였음에도 그림 3-(바, 사) 와 같이 카메라 포즈를 추적하고 변화된 시점에 맞게 가상 캐릭터를 증강시키는 것을 확인할 수 있었다.

표 2 는 다양한 모바일 기기 상에서 500 프레임을 처리하는데 소요된 평균 시간을 나타낸다. 정육면체 모델은 8 개의 정점과 18 개의 선을, 도자기 모델은 673 개의 정점과 2000 개의 선을 가진 모델이 각각 사용되었다. 비교적 복잡한 도자기 모델을 640×480 영상에서 추적하는 경우에도 끊어짐 없이 원활한 추적이 가능하였다.

그림 4 는 표 2 에서 Galaxy S에서 640×480 영상과 도자기 모델을 사용한 경우의 처리 시간의 각 단계별 비율을 보여 준다. 영상 형식 변환 과정에서 약 16 ms 가 소요되었는데, 듀얼 코어 CPU에서는 렌더링 쓰레드에서 따로 수행될 수 있었기 때문에 전체 수행 시간에 큰 영향을 주지 않았지만, 싱글 코어 CPU에서는 전체 수행 시간에 비해 다소 큰 부담을 주었다. 전체 수행 시간에서 가장 큰 비중을 차지했던 부분은 영상 전역에 걸친 컨벌루션 연산과 잦은 메모리 참조를 필요로 하는 에지 검출 과정으로 약 39 ms가 소요되었다. 수행 시간이 모델의 복잡도에 크게 영향을 받는 과정은 가시성 검사와 에지 샘플링 과정뿐으로 전체 수행 시간에 비해 작은 비중을 차지하기 때문에 전체 수행 시간은 모델의 복잡도보다 영상 크기에 주로 영향을 받는 것을 확인할 수 있었다.

표 2. 평균 처리 소요시간

| 모바일 기기 | 3차원 객체 | 영상 해상도 | 처리 소요 시간 (ms) |
|---|---|---|---|
| Samsung Galaxy S | 정육면체 | 320×240 | 43.41 |
| | | 640×480 | 84.26 |
| | 도자기 | 320×240 | 60.23 |
| | | 640×480 | 102.47 |
| LG Optimus 2X | 정육면체 | 320×240 | 19.99 |
| | | 640×480 | 58.27 |
| | 도자기 | 320×240 | 25.94 |
| | | 640×480 | 66.75 |
| Samsung Galaxy S2 | 정육면체 | 320×240 | 15.36 |
| | | 640×480 | 42.06 |
| | 도자기 | 320×240 | 20.90 |
| | | 640×480 | 46.49 |

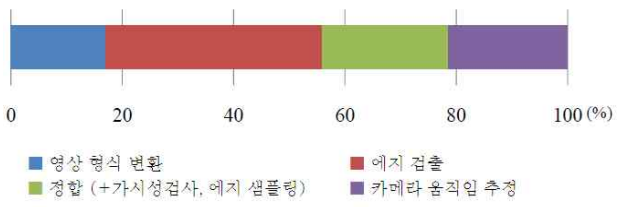



그림 4. 단계별 처리 소요시간 비율

(가) (나)

그림 5. (가) 실험 환경, (나) 영상에서 추출한 에지와 코너점.

그림 6. 모델 정보와 코너점을 함께 사용한 방법을 적용하여 카메라 포즈를 추적하고 가상 캐릭터를 증강시킨 결과.

## 다. 토의

제안된 방법은 텍스처가 없는 3차원 물체를 추적하는데 적합하고 모바일 기기에서도 빠르게 수행될 수 있었지만, 시각적 특이점으로 에지 정보만을 사용하는 것에는 몇 가지 한계가 있었다. 예를 들면, 에지 정보만으로는 대칭적인 물체의 대칭축에 대한 회전 변환을 알 수 없기 때문에 부정확한 카메라 포즈를 얻게 되었다. 이러한 문제는 여러 객체의 모델을 함께 사용하거나 모델 정보와 환경의 특징점을 함께 사용함으로써 극복될 수 있다[8]. 앞서 언급한 실험 결과에서는 도자기와 그 아래의 상자의 윗면을 함께 모델링함으로써 이러한 문제를 해결하였지만, 주변에 모델링이 용이한 물체가 없는 경우에는 주변 환경의 특징점을 모델 정보와 함께 이용하는 방법을 적용할 수 있다. 따라서 우리는 물체의 모델 정보와 물체 주변 환경의 특징점을 함께 사용하되, 모바일 기기의 성능을 고려하여 소수의 강한 특징점만을 추가로 사용하는 방법을 구현하고 실험하였다.

특징점 간의 대응점은 물체 주변 환경의 3차원 좌표를 알고 있는 강한 Harris 응답을 나타내는 코너점과 영상에서 추출한 코너점을 주변 패치을 이용하여 정합하였다. 3차원 좌표를 알고 있는 점을 사용하므로 연산량을 줄이기 위해 영상 전역에서 코너점을 추출하고 정합하는 대신, 이전 카메라 포즈로 투영된 코너점 주변의 좁은 영역에서 코너점을 추출하고 정합하였다. 여기서, 카메라 포즈 추적은 제안된 방법과 동일하게 수행하였다.

그림 5-(가)는 모델 정보와 주변의 강한 특징점(코너)을 함께 사용하는 방법을 적용하기 위한 실험 환경이다. 주변에는 텍스처가 거의 없고 모델링이 비교적 쉽지 않은 물체를 배치하였다. 그림 5-(나)는 영상에서 추출한 에지와 코너점을 나타낸다. 녹색 부분은 에지와 정합된 투영된 모델을 나타내고, 파란 원은 알고 있는 점의 3차원 정보를 바탕으로 구한 코너점의 탐색 범위이고 원의 중심의 붉은 점이 알고 있는 코너점을 나타낸다.

3차원 모델 정보와 주변 환경의 소수의 강한 코너점을 함께 사용한 결과 그림 6 과 같이 대칭적인 물체에 대해서도 정확한 카메라 포즈를 추정하고 변화된 시점에 맞게 가상 캐릭터를 증강시킬 수 있었다.

## 4. 결론

본 논문에서는 안드로이드 플랫폼의 모바일 기기를 이용하여 3차원 모델기반 추적 방법을 구현하고 실시간으로 카메라 포즈를 추적하여 가상 콘텐츠를 증강시켜 봄으로써, 다양한 3차원 환경에서 모바일 증강현실을 구현하기 위한 핵심적 역할을 수행할 수 있음을 확인하였다. 3차원 모델기반 추적 방법은 텍스처가 적은 환경에서 추적이 가능하다는 장점이 있지만 투영된 모델의 형태의 모호성으로 인하여 추적이 어려운 경우도 있다. 하지만 모바일 플랫폼 상에서도 여러 객체의 모델을 함께 사용하거나, 모델 정보와 특징점을 함께 사용하는 방법을 적용함으로써 이러한 문제점을 해결할 수 있음을 보였다.

현재 모바일 플랫폼 상에서 모델 정보와 특징점을 함께 사용하면서 보다 강건한 추적이 실시간에 가능한 카메라 추적 방법에 대한 연구를 진행하고 있다.

## 감사의 글

본 연구는 문화체육관광부 및 한국콘텐츠진흥원의 2009년도 문화콘텐츠산업기술지원사업의 연구결과로 수행되었음(3차년도).

## 참고문헌

[1] D. Wagner and D. Schmalstieg, “ARToolKitPlus for pose tracking on mobile devices,” in Computer Vision Winter Workshop, 2007, pp. 139-146.

[2] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, “Pose tracking from natural features on mobile phones,” in ISMAR, 2008, pp. 125-134.

[3] D. N. Ta, W. C. Chen, N. Gelfand, and K. Pulli, “SURFTrac: Efficient tracking and continuous object recognition using local feature descriptors,” in CVPR, 2009, pp. 2937-2944.

[4] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, “Real-time detection and tracking for augmented reality on mobile phones,” IEEE Transactions on Visualization and Computer Graphics, vol. 16, no. 3, pp. 355-368, 2010.

[5] G. Klein and D. Murray, “Parallel tracking and mapping on a camera phone,” in ISMAR, 2009, pp. 83-86.

[6] J. P. Lima, V. Teichrieb, J. Kelner, and R. W. Lindeman, “Standalone edge-based markerless tracking of fully 3-dimensional objects for handheld augmented reality,” in VRST, 2009.

[7] T. Drummond and R. Cipolla, “Application of lie algebras to visual servoing,” International Journal of Computer Vision, vol. 37, no. 1, pp. 21-41, 2000.

[8] B.-K. Seo, J.-I. Park, and H. Park. “Camera tracking using partially modeled 3-d objects with scene textures,” in ISVRI, 2011, pp. 293-298.