특집논문 (Special Paper)

방송공학회논문지 제18권 제3호, 2013년 5월 (JBE Vol. 18, No. 3, May 2013)

<http://dx.doi.org/10.5909/JBE.2013.18.3.362>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

스테레오 카메라 추적을 이용한 모바일 3차원 디스플레이 상의 실시간 증강현실

박 정 식^{a)}, 서 병 국^{a)}, 박 종 일^{b)†}

Real-Time Augmented Reality on 3-D Mobile Display using Stereo Camera Tracking

Jungsik Park^{a)}, Byung-Kuk Seo^{a)}, and Jong-Il Park^{b)†}

요 약

본 논문에서는 스테레오 카메라 추적을 이용한 모바일 3차원 디스플레이 상의 실시간 증강현실 프레임워크에 대하여 기술한다. 모델 기반 추적 방법에 기초하여 카메라 포즈는 스테레오 카메라 간의 기하 관계를 만족하도록 동시 추정된다. 동시 추정된 카메라 포즈를 통해 가상 콘텐츠는 교정된 스테레오 영상에 올바르게 증강된다. 스테레오 카메라 추적과 스테레오 영상 교정은 서로 다른 쓰레드에서 수행하고, 영상 형식 변환 및 스테레오 영상 교정은 GPU로 고속으로 처리함으로써 실시간으로 수행된다. 제안된 프레임워크는 스테레오 카메라와 3차원 디스플레이가 장착된 스마트폰에서 구현되었다.

Abstract

This paper presents a framework of real-time augmented reality on 3-D mobile display with stereo camera tracking. In the framework, camera poses are jointly estimated with the geometric relationship between stereoscopic images, which is based on model-based tracking. With the estimated camera poses, the virtual contents are correctly augmented on stereoscopic images through image rectification. For real-time performance, stereo camera tracking and image rectification are efficiently performed using multiple threads. Image rectification and color conversion are accelerated with a GPU processing. The proposed framework is tested and demonstrated on a commercial smartphone, which is equipped with a stereoscopic camera and a parallax barrier 3-D display.

Keyword : Stereo camera tracking, 3-D display, 3-D model-based tracking, mobile augmented reality

a) 한양대학교 전자컴퓨터통신공학과 (Hanyang University, Department of Electronics and Computer Engineering)

b) 한양대학교 컴퓨터공학부 (Hanyang University, Division of Computer Science and Engineering)

† Corresponding Author : 박종일(Jong-Il Park)

E-mail: jjipark@hanyang.ac.kr

Tel: +82-2-2220-0368

※ 본 연구는 미래창조과학부 및 정보통신산업진흥원의 IT융합 고급인력과정 지원사업의 연구결과로 수행되었음 (NIPA-2013-H0401-13-1008)

· Manuscript received April 2, 2013 Revised May 27, 2013 Accepted May 27, 2013

1. 서론

3차원 디스플레이는 사용자의 양쪽 눈에 각각 다른 시점의 영상을 보여줌으로써 사용자에게 입체감을 느끼게 해주는 디스플레이 장치이다. 이러한 3차원 디스플레이의 보급이 확산되면서 게임이나 영화 등의 3차원 콘텐츠를 즐길 수 있게 되었다. 또한 스테레오 카메라와 무안경식 3차원 디스플레이가 장착된 스마트폰이 등장함으로써 시간과 장소에 구애받지 않고 사진, 비디오와 같은 자신만의 3차원 콘텐츠를 제작하고 다른 사람들과 공유하는 것도 가능하게 되었다.

최근 실감 콘텐츠로서 많은 관심을 받고 있는 것 중에 모바일 증강현실이 있다. 증강현실은 현실 세계에 대한 카메라 영상에 가상 콘텐츠를 겹쳐 보여줌으로써 사용자의 몰입감을 증대시키거나 부가정보를 제공하는 기술이다. 실감나는 증강 현실을 위해서는 실제 세계와 가상 세계가 정확하게 정합되는 것이 중요하다. 따라서 일반적으로 증강현실 시스템은 센서나 카메라 영상을 이용하여 위치와 자세를 추정하게 된다. 최근에는 카메라와 여러 센서가 탑재된 스마트폰의 보급으로 인해 모바일 증강현실 서비스가 가능하게 되어 광고, 관광 등의 분야에서 활용되고 있다. 더욱이 3차원 디스플레이가 모바일 장치에 탑재되어, 이러한 기기를 이용한 더욱 실감나는 증강현실 응용이 가능하게 되었다.

모바일 3차원 디스플레이 상의 증강현실의 구현은 초기에는 HMD(Head Mounted Display)를 통해 이루어졌다. Feiner 등은 착용형 컴퓨터와 센서기반 증강현실을 이용한 캠퍼스 안내 시스템을 고안하였다^[1]. HMD를 이용하는 경우는 특수한 장비를 착용해야 입체 영상을 볼 수 있다는 불편함이 있었지만, 디스플레이 기술의 발전으로 특수한 장비 없이도 입체 영상을 볼 수 있게 되었다. Yuan 등은 스테레오 영상으로부터 3차원 기하구조를 복원하는 방법에 기반한 증강현실을 3차원 디스플레이가 장착된 스마트폰에 구현하였다^[2]. 하지만 이 방법은 연산량이 너무 많기 때문에 아직 실시간으로 동작하는데 어려움이 있다.

널리 알려진 모바일 플랫폼에서 실시간으로 동작하는 카메라 추적 방법으로는 마커기반 추적 방법^[3]과 자연 특징점

기반 추적 방법^[4]이 있다. 이러한 방법은 강건하고 빠른 추적이 가능하지만 대상이 평면적인 물체로 제한된다. 3차원 모델기반 추적 방법^[5] 또한 모바일 3차원 디스플레이 상의 증강현실의 구현에 유용하게 활용될 수 있다. 3차원 모델기반 추적 방법은 추적 대상 물체의 3차원 모델과 영상 에지를 매칭하여 추적하는 방법으로서 평면이 아닌 3차원 환경에 대해 추적이 가능하므로 3차원 디스플레이 상에서 증강콘텐츠의 입체감을 느끼는데 더욱 도움을 줄 수 있다. 게다가 3차원 모델기반 추적 방법은 모바일 플랫폼에서도 실시간으로 동작이 가능하다^[6].

하지만, 3차원 디스플레이 상의 증강현실의 구현을 위해서는 스테레오 카메라 영상에서 카메라 추적을 수행해야 하므로 더 많은 데이터의 처리와 연산량을 필요로 한다. 또한 스테레오 영상 교정과 같은 부가적인 처리도 필요하다. 따라서 이러한 처리들을 고속화할 필요가 있다. 컴퓨터 비전과 영상처리 분야에서는 많은 연산량이 필요한 알고리즘을 CUDA(Compute Unified Device Architecture)나 OpenCL(Open Computing Language)를 이용한 GPU의 병렬처리로 고속화하는 경우가 많다. 하지만, 모바일 플랫폼에서는 CUDA나 OpenCL을 이용할 수 없고, OpenGL ES(OpenGL for Embedded Systems) 2.0의 프로그래밍 가능한 셰이더를 통해 제한적이지만 GPGPU(General-Purpose Graphics Processing Unit)가 가능하다. 최근 모바일 플랫폼에서도 셰이더기반 GPU를 이용하여 영상처리 등의 알고리즘을 고속화하는 연구가 진행되고 있다. 예를 들어 Singhal 등은 모바일 GPGPU의 최적화 기법을 연구하고 다양한 컴퓨터 비전 알고리즘을 모바일 GPU 상에서 구현하였다^[7].

본 논문에서는 스테레오 카메라 추적을 이용한 모바일 3차원 디스플레이 상의 실시간 증강현실을 위한 프레임워크를 제안한다. 3차원 디스플레이 상의 증강현실을 위해서는 스테레오 영상으로부터 카메라 포즈가 실시간으로 정확하게 추정되어야 할 뿐만 아니라, 추정된 카메라 포즈가 두 카메라 간의 기하관계를 만족하도록 동시 최적화함으로써 증강된 콘텐츠의 입체감이 올바르게 나타나도록 할 필요가 있다. 그리고 3차원 디스플레이는 두 시점 영상이 평행하고 같은 평면상에 놓인 것을 가정하지만 스테레오 카메라 센

서가 실제로는 병행하지 않기 때문에 스테레오 영상 교정을 수행할 필요가 있다. PC 상에서는 스테레오 영상 교정이 빠르게 수행될 수 있지만 모바일 플랫폼에서는 잦은 메모리 접근과 실수 연산으로 인해 시간이 많이 소요되므로 다중 쓰레드와 GPU를 이용하여 실시간 처리가 가능하도록 하였다. 또한 제안된 프레임워크는 영상 전역에서 에지를 검출하는 대신, 투영된 3차원 모델의 선 샘플의 대응점 탐색과 동시에 국소 영역에서만 에지를 검출하는 3차원 모델 기반 추적방법^[6]을 적용하여 처리 속도를 향상시켰다.

II. 증강현실 프레임워크

그림 1은 제안된 증강현실 프레임워크를 나타낸다. 프레임워크는 두 개의 쓰레드로 구성되어 있다. 하나는 추적 쓰레드로서 스테레오 카메라 추적을 수행하고, 다른 하나는 렌더링 쓰레드로서 YUV 형식의 입력 영상을 RGB 형식으로 변환하여 스테레오 영상 교정을 수행한 뒤, 교정된 영상과

가상 콘텐츠를 렌더링한다. 렌더링 쓰레드에서 수행되는 영상 형식 변환과 스테레오 영상 교정 과정은 셰이더기반 GPU 병렬처리로 고속으로 처리된다. 쓰레드간의 병렬성을 위하여 추적은 교정되기 전의 카메라 간의 기하변환 관계를 이용하여 교정되지 않은 영상에 대해 수행하고, 추정된 카메라 포즈에 교정 변환 행렬을 곱한 뒤 렌더링 쓰레드로 전달하여 가상 콘텐츠가 교정된 영상 위에 올바르게 증강될 수 있도록 한다. 카메라 간의 기하변환 관계와 스테레오 교정 변환에 필요한 정보의 계산은 사전에 수행해 둔다.

1. 스테레오 카메라 추적

제안된 증강현실 프레임워크에서는 3차원 모델기반 추적 방법을 스테레오 추적으로 확장한 방법을 사용한다. 3차원 모델기반 카메라 추적 방법은 영상에서 검출된 에지와 영상평면에 투영된 3차원 모델을 정합하여 카메라 위치 및 자세를 추정하는 방법이다. 그림 2는 3차원 모델기반 카메라 추적 방법의 대응점 탐색 과정을 나타낸다.

그림 1. 프레임워크의 흐름도

Fig. 1. The overall workflow of the proposed framework

그림 2 . 3차원 모델기반 추적 방법의 대응점 탐색 과정
Fig. 2. Correspondence searching of the 3-D model-based tracking

먼저 이전 프레임의 포즈와 3차원 모델의 법선 벡터를 이용한 은면 제거법으로 후면의 선을 제거하고 두 면에 의해 공유되는 선을 제거하여 외곽선을 추출한다. 실제로는 이전 프레임의 포즈에서 본 3차원 모델의 은면과 현재 프레임의 은면은 다르지만 이전 프레임의 카메라 포즈로부터 현재 프레임의 카메라 포즈 간의 카메라 움직임이 작다고 가정함으로써 효과적으로 은면 제거가 가능하다. 추출된 선은 영상 평면으로 투영되어 일정 간격으로 샘플링되고, 각 선의 법선 방향으로 가장 가까운 영상 에지를 대응점으로 정의한다.

일반적으로 캐니 에지 검출기(Canny edge detector)를 이용하여 영상 전역에서 에지 성분을 검출할 수 있다^[8]. 하지만 3차원 모델기반 추적방법에서는 투영된 3차원 모델의 외곽선 부근의 에지만 필요하므로 영상 전역에서 에지를 검출하는 것은 불필요한 연산으로 인해 느리고 비효율적이

다. 따라서 투영된 3차원 모델의 선 샘플의 대응점 탐색범위에서만 에지를 검출하는 방법을 적용하여 속도를 개선하였다^[6].

그림 3은 개선된 에지 검출 방법을 나타낸다. 이전 카메라 포즈로 투영된 3차원 모델의 선 샘플의 위치 p 로부터 선 샘플의 법선 방향인 탐색 범위 선상으로 한 화소씩 이동하면서 소벨(Sobel) 필터를 적용하고, 이전 위치와 다음 위치의 필터 응답보다 큰 응답을 나타내는 위치를 에지 성분의 국소 극대점으로 간주하여 대응점으로 정의한다. 그림 3에서는 p 와 가장 가까운 국소 극대점인 p' 이 대응점으로 선택된다.

스테레오 카메라 추적의 경우 대응점 탐색은 좌우 영상에 대해 독립적으로 수행한다.

카메라 포즈 E 는 카메라 움직임 ΔE 에 의해 이전 프레임의 카메라 포즈 $t-1$ 으로부터 다음과 같이 갱신된다.

그림 3. 지역적 에지성분 검출을 통한 개선된 대응점 탐색 방법

Fig. 3. Improved correspondence searching using local edge component detection

$$\mathbf{E}_t = \mathbf{E}_{t-1} \cdot \Delta \mathbf{E} \quad (1)$$

카메라 움직임 $\Delta \mathbf{E}$ 는 이전 프레임으로부터 현재 프레임으로의 카메라 회전 변환 $\mathbf{R}$ 및 평행이동 변환 $\mathbf{t}$ 로 표현되는 데, 이는 x, y, z 축 방향에 대한 평행이동 및 회전 변환의 기저 행렬 $\mathbf{G}$ 와 각 변환 성분의 변화량 a 의 가중합으로 나타낼 수 있다.

$$\Delta \mathbf{E} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} = \sum_{i=0}^5 e^{\alpha_i \mathbf{G}_i} \approx \mathbf{I} + \sum_{i=0}^5 \alpha_i \mathbf{G}_i \quad (2)$$

이전 프레임의 카메라 포즈로 투영된 3차원 모델의 선

샘플과 대응하는 영상 에지 간의 2차원 영상 좌표계 상의 거리 d 는 투영된 위치 $\mathbf{x}$ 에서 변환 기저 $\mathbf{G}$ 에 의한 변화량과 같아야 한다. 이를 만족 시키는 a 는 다음과 같이 구할 수 있다.

$$\alpha^* = \arg \min_{\alpha_i} \left\{ \sum_j \left\| d^j - \sum_{i=0}^5 \alpha_i (\mathbf{n}_j \cdot \mathbf{l}_{i,j}) \right\|^2 \right\} \quad (3)$$

여기서 $\mathbf{n}$ 은 선 샘플의 법선 벡터, $\mathbf{l}$ 은 $\mathbf{x}$ 위치에서의 미소 변화량 벡터이다. 벡터 $\mathbf{l}$ 은 3차원 공간상의 점 $\mathbf{X}$ 의 편미분과 $\mathbf{x}$ 를 이용하여 계산할 수 있으며, $\mathbf{X}$ 의 편미분은 변환 기저 $\mathbf{G}$ 에 의해 다음과 같이 계산된다.

$$\begin{bmatrix} u' & v' & w' \end{bmatrix}^T = \mathbf{K} \mathbf{E}_{t-1} \mathbf{G}_t \mathbf{X} \quad (4)$$

$$\mathbf{x} = [u \quad v \quad w]^T \quad (5)$$

$$\mathbf{l}_i = \left[\left(\frac{u}{w} \right)' \quad \left(\frac{v}{w} \right)' \right]^T = \left[\frac{u'}{w} - \frac{uw'}{w^2} \quad \frac{v'}{w} - \frac{vw'}{w^2} \right]^T \quad (6)$$

K는 카메라의 내부 파라미터 행렬이다. 식 3에 따라 α 는 최소자승법으로 계산되고 식 2로부터 카메라 움직임 $\Delta \mathbf{E}$ 가, 식 1로부터 카메라 포즈 $\mathbf{E}$ 가 각각 계산된다. 이러한 과정은 재투영 오차가 작아질 때까지 반복적으로 수행된다.

스테레오 카메라의 경우 각 카메라의 포즈는 다음과 같이 계산될 수 있다.

$$\begin{aligned}\mathbf{E}_{L,t} &= \mathbf{E}_{L,t-1} \cdot \Delta \mathbf{E}_L \\ \mathbf{E}_{R,t} &= \mathbf{E}_{R,t-1} \cdot \Delta \mathbf{E}_R\end{aligned}\tag{7}$$

$$\begin{aligned}\Delta \mathbf{E}_L^* &= \arg \min_{\Delta \mathbf{E}_L} \left\{ \sum_i \left\| \mathbf{K}_L \mathbf{E}_{L,t-1} \Delta \mathbf{E}_L \mathbf{X}_i - \mathbf{x}_i \right\|^2 \right\} \\ \Delta \mathbf{E}_R^* &= \arg \min_{\Delta \mathbf{E}_R} \left\{ \sum_j \left\| \mathbf{K}_R \mathbf{E}_{R,t-1} \Delta \mathbf{E}_R \mathbf{X}_j - \mathbf{x}_j \right\|^2 \right\}\end{aligned}\quad (8)$$

여기서 $\mathbf{K}$ 는 카메라 내부 행렬, $\mathbf{X}$ 는 모델의 선 샘플의 3차원 좌표, $\mathbf{x}$ 는 $\mathbf{X}$ 의 대응점이고, 첨자 L, R 은 각각 왼쪽 카메라와 오른쪽 카메라를 나타낸다. 두 카메라의 포즈는 두 카메라의 움직임이 같다고 가정하면 왼쪽 카메라의 움직임 $\Delta \mathbf{E}_L^*$ 는 식 9와 같이 동시추정이 가능하다.

여기서 ΔE_L^* 를 계산하기 위해 ΔE_L 와 ΔE_R 를 모두 알아야

하지만, 스테레오 캘리브레이션을 통해 두 카메라간의 기하 관계를 알고 있다면 식 9는 식 10과 같이 ΔE_L 에 대한 식으로 표현할 수 있다.

오른쪽 카메라의 포즈 $\mathbf{E}_{R,t}$ 는 추정된 움직임을 통해 갱신된 왼쪽 카메라의 포즈 $\mathbf{E}_{L,t}$ 와 두 카메라 간의 기하변환 관계 $\mathbf{E}_{L \rightarrow R}$ 의 곱으로 계산할 수 있다.

2. 영상 형식 변환과 스테레오 영상 교정

모바일 플랫폼에서 카메라 영상의 렌더링은 OpenGL ES를 이용하여 사각형에 영상 텍스처를 입혀 그리는 방식으로 수행된다. 하지만 안드로이드 플랫폼의 카메라 영상의 기본 형식은 OpenGL ES API에서 텍스처 형식으로 지원하지 않는 YUV420 형식이므로 카메라 영상을 RGB 형식으로 변환할 필요가 있다. 이러한 변환은 화소별로 행렬 곱셈 연산을 통해 간단하게 계산이 가능하다.

스테레오 카메라의 센서는 이상적으로는 같은 평면상에
평행하게 배열되어야 한다. 하지만 제조 과정에서 평행하
지 않게 되고 렌즈 왜곡도 발생하기 때문에 3차원 디스플레
이에 렌더링된 스테레오 영상의 입체감이 올바르게 나타나
지 않을 수 있다. 스테레오 영상 교정과 렌즈 왜곡 제거를
통해 이러한 문제를 해결할 수 있다. 스테레오 영상 교정은
선형 변환이지만 렌즈 왜곡 제거는 비선형 변환이기 때문
에 미리 각 픽셀이 이동할 위치를 계산해서 룩업 테이블을
만들어두고, 이를 이용하여 영상 교정을 수행하는 방법이
효율적이다. 하지만, 모바일 플랫폼에서는 이러한 방법이
메모리 접근과 실수 연산을 많이 필요하기 때문에 시간이
많이 소요된다.

우리는 영상 형식 변환과 스테레오 영상 교정을 실시간으로 수행하도록 GPU 병렬처리를 이용하였다. OpenGL ES 2.0 API는 프로그래밍 가능한 셰이더를 지원하기 때문에 모바일 플랫폼에서도 제한적으로 GPGPU 기술이 사용

$$\Delta \mathbf{E}_L^* = \arg \min_{\Delta \mathbf{E}_L} \left\{ \sum_i \left\| \mathbf{K}_L \mathbf{E}_{L,t-1} \Delta \mathbf{E}_L \mathbf{X}_i - \mathbf{x}_i \right\|^2 + \sum_j \left\| \mathbf{K}_R \mathbf{E}_{R,t-1} \Delta \mathbf{E}_R \mathbf{X}_j - \mathbf{x}_j \right\|^2 \right\} \quad (9)$$

$$\Delta \mathbf{E}_L^* = \arg \min_{\Delta \mathbf{E}_L} \left\{ \sum_i \left\| \mathbf{K}_L \mathbf{E}_{L,i-1} \Delta \mathbf{E}_L \mathbf{X}_i - \mathbf{x}_i \right\|^2 + \sum_j \left\| \mathbf{K}_R \mathbf{E}_{L,j-1} \Delta \mathbf{E}_L \mathbf{E}_{L \rightarrow R} \mathbf{X}_j - \mathbf{x}_j \right\|^2 \right\} \quad (10)$$

가능하다. 영상 형식 변환과 스테레오 영상 교정은 모두 화소별로 독립적인 연산을 수행하므로 병렬 처리에 적합하다. 게다가, 교정된 영상을 디스플레이 장치로 바로 출력하기 때문에 GPU에서 CPU로의 데이터 전송으로 인한 오버헤드가 발생하지 않는다.

스테레오 영상 교정 방법으로는 기초행렬(Fundamental matrix)을 이용한 방법과 3차원 공간상의 회전 및 평행이동 변환을 이용한 Bouguet의 방법^[9]이 있다. 본 논문에서는 OpenCV 라이브러리에서 제공하는 Bouguet의 방법을 이용하여 3차원 변환행렬을 계산한 뒤 이를 이용하여 교정된 위치로 옮겨올 화소의 좌표를 룩업테이블로 만들어 사용하였다. 카메라의 초점 모드가 변하지 않는 한 룩업테이블의 값이 바뀔 필요가 없기 때문에 룩업테이블은 GPU로 최초 한 번만 전송되고 이에 따른 오버헤드는 없다고 볼 수 있다.

GPU에서는 룩업테이블을 참조하여 각 픽셀별로 양선형 보간법이 적용된 역방향 워핑을 수행함으로써 영상이 실시간으로 교정된다.

III. 실험결과

실험에 사용된 모바일 기기는 LG Optimus 3D로서

ARM Cortex A9 듀얼코어 1.0 GHz CPU, PowerVR SGX 540 GPU, 512 MB RAM, 그리고 스테레오 카메라와 패럴렉스 배리어 방식의 무안경식 3차원 디스플레이가 탑재되었다.

Java의 느린 속도를 고려하여 카메라 입력 및 사용자 인터페이스 이외의 처리는 C/C++로 구현하고 JNI(Java Native Interface)를 통해 호출하였다. 3차원 디스플레이의 구동에는 LG Real3D SDK를 이용하였다. 영상 형식 변환 및 스테레오 영상 교정은 GLSL(OpenGL Shader Language)로 작성되었다. 카메라 입력은 좌우 카메라의 영상이 side-by-side 방식으로 결합된 720×480 YUV420 영상으로 추적 처리에는 Y 채널만 추출하여 사용하였다. 이 때, 카메라 내부 파라미터 및 두 카메라의 기하학적 관계가 변하지 않도록 초점거리를 고정시켰다.

스테레오 카메라 캘리브레이션은 오프라인에서 수행하여 카메라 내부 파라미터, 왜곡 보정 파라미터, 두 카메라 간의 관계를 구하였고, 이를 이용하여 스테레오 영상 교정 변환 행렬과 영상 교정에 사용될 룩업테이블을 계산하였다. 그림 4의 오른쪽은 스테레오 카메라 캘리브레이션을 통해 추정된 왼쪽 카메라 좌표계 상의 오른쪽 카메라 및 체스보드 패턴의 위치를 도식화한 것이다. 두 카메라 간의 관계에서 x축 방향 평행이동 성분은 24.19 mm로, Optimus 3D의

그림 4. 실험에 사용된 Optimus 3D 스마트폰(좌)과 스테레오 카메라 캘리브레이션 결과(우)

Fig. 4. Optimus 3D smartphone(left) and its stereo camera calibration result(right)

스테레오 카메라의 실제 카메라간 거리 24 mm에 근접함을 볼 수 있다.

그림 5는 추적 대상으로 사용된 물체와 3차원 모델을 나

타낸다. 정육면체 모델은 8개의 정점과 18개의 선, 도자기 모델은 673개의 정점과 2000개의 선으로, 고양이 모델은 1252개의 정점과 3750개의 선으로 각각 구성되었다.

그림 5. 추적에 사용된 대상 물체(좌)와 3차원 모델(우)

Fig. 5. Target objects(left) and their 3-D CAD models(right) for tracking

그림 6. 추적 및 증강 결과(도자기). 상: side-by-side 영상, 하: 애너글리프 영상

Fig. 6. Tracking and augmentation results (pottery). Up: side-by-side images. Bottom: anaglyph images

그림 6는 도자기 모델에 대한 추적 및 증강 결과를 나타낸다. 3차원 영상은 패럴렉스 배리어 모드로 렌더링하는 대신 애너글리프(Anaglyph)로 렌더링하였다. 스테레오 카메라의 기하 관계를 이용하여 카메라 위치 및 자세를 동시 추정하고 영상을 교정한 뒤, 교정 후의 카메라 파라미터를 이용하여 각 카메라 위치 및 자세를 보정한 결과이다. 애너글리프 영상을 통해 증강된 가상 객체의 3차원 효과가 올바르게 표현되는 것을 확인할 수 있었다.

표 1은 500 프레임동안 측정한 평균 소요 시간을 나타낸다. 여기서 기존의 카메라 추적 방법은 영상 전역에서 에지를 검출하여 추적하는 방법이고, 개선된 카메라 추적 방법은 투영된 3차원 모델의 선 샘플의 대응점 탐색 범위에서만 소벨 필터를 적용하여 추적하는 방법이다. 카메라 추적은 모델에 따라 27 ~ 71 ms가 소요되어 간단한 모델의 경우엔 30 fps 이상으로 동작이 가능하였다. 그리고 추적 방법이 개선됨에 따라 추적 속도가 많게는 3배가량 향상되었다. 영상 형식 변환 및 스테레오 영상 교정에는 약 14 ms가 소요되는데, 듀얼코어 CPU를 가진 모바일 기기 상에서 두 스

레드가 병렬로 수행되므로 전체적인 속도는 상대적으로 느린 카메라 추적 속도에 좌우된다고 볼 수 있다. 본 논문에서는 영상 형식 변환 및 스테레오 영상 교정은 GPU를 활용하여 고속으로 처리되는데, 표 2에 나타나듯이 CPU에서 수행했을 때와 비교하여 약 8배가량 빠르게 처리된다.

표 2. 영상 형식 변환 및 스테레오 영상 교정의 CPU 구현과 GPU 구현의 성능 비교

Table 2. Performance comparison between CPU and GPU implementation of color conversion and stereo image rectification

	CPU 구현	GPU 구현
영상 형식 변환 및 스테레오 영상 교정	117 ms	14 ms

IV. 결 론

본 논문에서는 모바일 3차원 디스플레이 상의 증강현실 프레임워크를 제안하였다. 제안된 프레임워크에서는 3차원 모델 기반 추적 방법에 기초하여 두 카메라의 포즈를 동시 추정하는 스테레오 카메라 추적 과정과 스테레오 영상 교정 과정으로 구성되어 3차원 디스플레이 상에 스테레오 카메라 영상과 가상 콘텐츠가 올바르게 증강될 수 있도록 하였다. 각 과정은 다중 스레드를 이용하여 병렬로 처리되며, 특히 YUV 영상에서 RGB 영상으로의 변환과 스테레오 영상 교정은 GPU를 이용하여 고속처리 함으로써 모바일 플랫폼에서 실시간으로 동작할 수 있도록 하였다. 제안된 프

표 1. 평균 소요 시간

Table 1. Average processing time

	영상 형식 변환 및 스테레오 영상 교정	카메라 추적 (개선된 방법)	카메라 추적 (기존의 방법)
정육면체	14 ms	27 ms	85 ms
도자기	14 ms	61 ms	110 ms
고양이	14 ms	71 ms	102 ms

그림 7. 추적 및 증강 결과(고양이).

Fig. 7. Tracking and augmentation results (cat).

레이아웃은 스테레오 카메라와 3차원 디스플레이가 장착된 스마트폰 상에 구현되어 추적 및 증강이 올바르게 수행됨을 확인하였다.

참 고 문 헌

- [1] S. Feiner, B. MacIntyre, T. Hollerer, and A. Webster, "A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment," *Personal Technologies*, vol. 1, no. 4, pp. 208 - 217, 1997.
- [2] C. Yuan, M. Liao, X. Hu, and P. Mordohai, "Depth sensing and augmented reality technologies for mobile 3D platforms," *SID Symposium Digest of Technical Papers*, vol. 43, no. 1, pp. 233 - 236, 2012.
- [3] D. Wagner and D. Schmalstieg, "ARToolKitPlus for pose tracking on mobile devices," in *Proc. Computer Vision Winter Workshop*, 2007, pp. 139 - 146.
- [4] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Real-time detection and tracking for augmented reality on mobile phones," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 3, pp. 355 - 368, 2010.
- [5] T. Drummond and R. Cipolla, "Real-time visual tracking of complex structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 932 - 946, 2002.
- [6] B.-K. Seo, J. Park, H. Park, and J.-I. Park, "Real-time visual tracking of less-textured three-dimensional objects on mobile platforms," *Optical Engineering*, vol. 51, no. 12, pp. 127202, Dec. 2012.
- [7] N. Singhal, J. W. Yoo, H. Y. Choi, and I. K. Park, "Implementation and optimization of image processing algorithms on embedded GPU," *IEICE Trans. on Information and Systems*, vol. E95-D, no. 5, pp. 1475 - 1484, May 2012.
- [8] J. Park, B.-K. Seo, and J.-I. Park, "A framework of augmented reality on 3-D mobile display using stereo camera tracking," in *Proc. International Workshop on Advanced Image Technology (IWAIT'13)*, pp. 745 - 748, Jan. 2013.
- [9] G. Bradski and A. Kaehler, *Learning OpenCV*, O'Reilly, 2009.

저 자 소 개

박 정 식

- 2010년 : 한양대학교 전자전기컴퓨터공학부 학사
- 2012년 : 한양대학교 전자컴퓨터통신공학과 석사
- 2012년 ~ 현재 : 한양대학교 전자컴퓨터통신공학과 박사 과정
- 주관심분야 : 3차원 영상처리, 증강현실, GPGPU

서 병 국

- 2006년 : 한양대학교 전자전기컴퓨터공학부 학사
- 2008년 : 한양대학교 전자컴퓨터통신공학과 석사
- 2008년 ~ 현재 : 한양대학교 전자컴퓨터통신공학과 박사 과정
- 주관심분야 : 3차원 영상처리, 컴퓨터 비전, 증강현실, 인간컴퓨터 상호작용

박 종 일

- 1987년 : 서울대학교 전자공학과 학사
- 1989년 : 서울대학교 전자공학과 석사
- 1995년 : 서울대학교 전자공학과 박사
- 1992년 ~ 1994년 : 일본 NHK 방송기술연구소 객원연구원
- 1995년 ~ 1996년 : 한국방송개발원 선임연구원
- 1996년 ~ 1999년 : 일본 ATR 지능영상통신연구소 연구원
- 1999년 ~ 현재 : 한양대학교 공과대학 컴퓨터공학부 교수
- 주관심분야 : 가상현실, 컴퓨터그래픽스/비전, 3차원 영상처리, 인간컴퓨터 상호작용