
디지로그북에서의 비전 기반 실시간 페이지 인식 및 마커리스 추적 방법

A Real-time Vision-based Page Recognition and Markerless Tracking in DigilogBook

김기영, Kiyoun Kim*, 우운택, Woontack Woo**

요약 마커리스 추적 기술은 명시적 마커의 부착없이 실시간으로 카메라의 자세를 획득하게 해, 증강 현실 애플리케이션에서의 활용성이 높아지고 있다. 본 논문에서는 마커리스 증강 현실 책인 디지로그북에 적합한 새로운 비전기반 페이지 인식 및 추적 방법을 제안한다. 제안된 방법은 페이지의 직교 영상만을 요구하며, 긴 학습 과정이 필요없고 실시간으로 동작한다. 실시간 페이지 인식은 페이지의 비교 우선 순위 평가 함수와 SIFT (Scale Invariant Feature Transform) 설명자를 활용하여 두 단계로 수행된다. 또한, 멀티 코어 프로그래밍을 이용하여 프레임간 특징점 추적과 페이지 인식을 분리해, 25fps ~ 30fps 의 카메라 추적 실시간성을 보장한다. 제안된 알고리즘은 향후 다수의 객체를 추적하는 증강 현실 애플리케이션에 확장될 수 있다.

Abstract Many AR (Augmented Reality) applications have been interested in a marker-less tracking since the tracking methods give camera poses without attaching explicit markers. In this paper, we propose a new marker-less page recognition and tracking algorithm for an AR book application such as DigilogBook. The proposed method only requires orthogonal images of pages, which need not to be trained for a long time, and the algorithm works in real-time. The page recognition is done in two steps by using SIFT (Scale Invariant Feature Transform) descriptors and the comparison evaluation function. And also, the method provides real-time tracking with 25fps ~ 30fps by separating the page recognition and the frame-to-frame matching into two multi-cores. The proposed algorithm will be extended to various AR applications that require multiple objects tracking.

핵심어: *Augmented Reality, DigilogBook, Markerless Tracking, Page Recognition, SIFT*

본 연구는 문화체육관광부 한국문화콘텐츠진흥원의 문화 콘텐츠 기술 연구소 육성 사업의 연구결과로 수행되었음

*주저자 : 광주과학기술원 정보통신공학과 박사 학생 e-mail: kkim@gist.ac.kr

**교신저자 : 광주과학기술원 정보통신공학과 교수 e-mail: wwoo@gist.ac.kr

1. 서론

증강 현실 (Augmented Reality)에서 여러 응용들 중 가상 콘텐츠를 실제 책에 결합시킨 매직북 [1]과 디지로그북 [7]등이 하나의 킬러 애플리케이션 (Killer Application)으로 여겨지고 있다. 책 속 이야기와 그에 상응하는 디지털 콘텐츠를 실제 책 페이지에 오버레이함으로써, 책이라는 물리적 한계를 극복해 독자에게 새롭고 확장된 경험을 주고 있는 것이다.

이러한 증강 현실 책 애플리케이션의 확장성을 위해서는 마커리스 카메라 추적 (Marker-less Camera Tracking) 기술이 요구된다. 책 페이지의 카메라 자연 영상만을 이용해 콘텐츠를 증강하기 때문에, 기존의 눈에 거슬리는 마커를 출판된 책에 붙일 필요가 없어 출판된 책 자체를 이용할 수 있다. 그러나, 마커 대신 마커리스 추적 기술을 실제 책에서 활용하기 위해서는 실시간 고유 페이지 추출과 강건한 카메라 자세 추적 문제가 해결되어야 한다.

본 논문에서는 마커리스 증강 현실 책인 디지로그북에 적합한 새로운 비전기반 페이지 인식 및 추적 방법을 제안한다. 실시간 페이지 인식은 추출된 자연 특징점의 SIFT (Scale Invariant Feature Transform) 설명자 [4], 페이지의 히스토그램 분포 정보, 그리고 페이지 순서 요소를 이용한다. 페이지가 인식되면, 페이지에서 인식된 특징점과 추가 특징점을 추출하여 강건하게 호모그래피 (Homography)를 계산하고, 이로부터 카메라 자세를 추정한다.

2. 관련 연구 및 문제 정의

2.1 관련 연구

마커리스 증강 현실을 구현하기 위한 방법은 크게 두 종류로 나눌 수 있다. 첫번째 방법은 인식에 의한 추적 방법 (Tracking by Detection)이다. 이 방법은 매 프레임마다 저장된 특징점을 인식하는 과정을 수행하고, 인식된 특징점을 이용해 카메라 자세를 구한다. 그러나 인식하는 과정에 긴 시간이 소요되어 실시간성을 보장하지 못한다는 단점이 있었다 [4]. 최근에는 특징점을 사전에 학습하여 인식 시간을 단축한 알고리즘들이 많이 연구되었다 [2][3].

두번째 접근 방법은 SLAM (Simultaneous Localization and Mapping) 응용이다. 이 방법은 특징점들에 대한 정보가 없는 임의의 공간에서 특징점들을 추출하고 3 차원 복원을 동시에 수행하여, 카메라의 상대적인 자세를 계산하는 방식이다. 증강 현실의 범위를 넓은 환경 자체로 확장할 수 있다는 큰 장점이 있었지만, 실시간으로 복원된 3 차원 점들을 놓칠 경우 카메라의 자세를 다시 회복하기 어려운 단점과 좋은 결과를 위한 광각 렌즈의 필요성 문제가 있었다. 최근에는 카메라의 위치 회복을 위한 빠른 Relocalisation 알고리즘들과 SLAM의 안정성을 향상시킨 알고리즘들이 개발되었다 [8][9].

책을 이용한 증강 현실 응용에는 첫번째 방식의 추적 기법이 대부분 사용되어오고 있다. C. Scherrer 등은 M.

Ozuysal 등의 알고리즘을 사용하여 실제 책에 적용한 응용을 보였다 [10]. H. Yang 등은 마커와 [10] 방식을 하이브리드 방식으로 사용하여 다수 책 페이지 인식을 지원하고 있다 [12]. N. Taketa 등은 다수의 멀티코어를 활용하고, 페이지의 컬러 분포를 활용한 책 응용을 개발하였다 [11].

제안하는 알고리즘은 상대적으로 느린 SIFT 특징점 인식 방법을 멀티 코어 프로그래밍 방식을 적용하여, 상대적으로 빠른 Optical flow 방식과 혼용해 특징점을 인식 및 추적한다. 제안된 알고리즘은 기존의 마커리스 추적 방법들 [2, 3, 10]에 비해 긴 학습 과정을 요구하지 않는다. 또한, 카메라 프레임의 저하없이 실시간으로 추적하고자하는 하나의 페이지를 인식하고, 추적한다. 실시간 페이지 인식을 위한 강건한 단계적 접근 방법을 제시한다. 이는 마커리스 증강 현실 책 저작 과정에서 실시간으로 저작할 수 있는 필수 요소라고 할 수 있다.

2.2 문제 정의

마커리스 디지로그북은 텍스처가 풍부한 그림과 글로 이루어진 일반 책을 사용한다. 일반 책의 페이지는 평면이고, 사용자는 한 번에 한 페이지씩 본다고 가정한다. 그리고 사전에 인식 및 추적하고자하는 페이지를 직각 방향에서 캘리브레이션 (Calibrated)된 카메라로 영상을 획득하고, 특징점을 추출했다고 하자. 실시간 페이지 인식은 입력으로 들어오는 카메라 영상에서 추출된 특징점과 총 S 개의 페이지로부터 저장된 동일 특징점들을 비교해, 유사성이 큰 영상을 찾는 과정이다. 이를 표현하면 다음과 같다.

$$k^* = \underset{0 \leq k < S}{\operatorname{argmax}} \sum_{i=1}^{N_k} \operatorname{Sim}(x_{k,i}, M_{k,i}) \quad (1)$$

위 식에서, N_k 는 k 번째 페이지에서 추출된 총 특징점 개수이며, M_k 는 그 특징점을 의미한다. 유사성 측정 함수는 $\operatorname{Sim}(\cdot)$ 으로 표기한다.

실시간 카메라 자세 추정은 식 (1)에서 구해진 페이지 인덱스 (k^*)와 대응하는 특징점을 기반으로 최적의 회전 (R)과 이동 행렬(t)을 구하는 과정이다.

$$P^* = \underset{(P)}{\operatorname{argmin}} \sum_{i=1}^{N_{k^*}} \operatorname{Dist}(x_{k^*,i}, P_{k^*} M_{k^*,i}) \quad (2)$$

위 식에서, P 는 (R t)를 포함하는 투영행렬을 의미하고, $\operatorname{Dist}(\cdot)$ 는 투영된 점과 관찰된 점간의 유클리드 거리 측정 함수이다.

3. 페이지 인식 및 추적 방법

제안된 알고리즘은 25fps~30fps의 증강 현실 애플리케이션 개발을 지원하기 위해, 멀티 코어 (Multi-core) 기반의 프로그래밍 기법을 활용한다. 즉, 페이지 인식과 페이지 추적이 각각의 CPU 코어에 할당되어 동작하게 된다. 또한, 페이지 인식에서 강건한 특징점 인식 및 매칭을 위해 피라미드 영상과 SIFT 설명자를 활용한다. 전체 알고리즘 흐름도는 그림 1과 같다. 그림에서 P 는

사전에 획득한 직교 페이지 영상에서 추출한 SIFT 특징점과 설명자 등의 정보를 의미하며, I_i 는 i 시간에서의 카메라 입력 영상을 의미한다. 성공적인 카메라 추적 결과는 N_{in} (인라이어 개수)와 V_{rp} (재투영 오차)의 값에 의해 판별된다. 상대적으로 프레임간 매칭은 인식 과정에 비해 빠르다는 사실과 인식 과정이 반드시 동기화가 되지 않아도 추적이 실패하지 않는다는 사실을 이용한다.

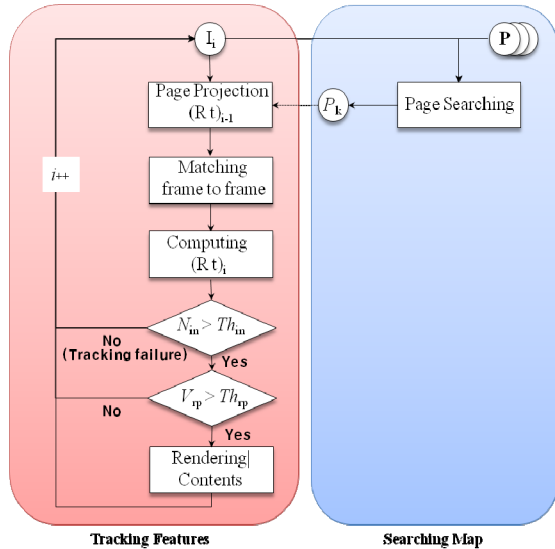


그림 1. 제안된 멀티코어 알고리즘 전체 흐름도 개요

3.1 페이지 인식

빠른 페이지 인식을 위해 두 단계 하이브리드 방법을 이용한다. 첫번째는 대략적으로 가능성이 높은 후보 페이지를 선정하는 단계이다. 영상의 히스토그램 정보와 페이지 히스토리 정보를 제한인자로 사용한다. 다음 식값을 프레임당 계산함으로써, 페이지의 비교할 순서를 결정하게 된다. k 번째 페이지의 비교 순위 값은 다음 식과 같이 결정된다.

$$\text{Score}(k) = \omega_1 E_{\text{DIST}}(k) + \omega_2 E_{\text{SSD}}(k), (\omega_1 + \omega_2 = 1) \quad (3)$$

$E_{\text{DIST}}(\cdot)$ 는 전에 안정적으로 인식된 페이지와 k 번째 페이지의 인덱스 차이값이며, 페이지 변화가 없을 경우 0 을 갖는다. $E_{\text{SSD}}(\cdot)$ 는 k 번째 영상과 입력 영상의 SSD 값이다. 가중치 w_1 과 w_2 는 실험에 의해 정해진다. $\text{Score}(\cdot)$ 값이 0 에 가까울수록 높은 비교 우선 순위를 갖는다.

페이지의 우선순위가 정해지면, 두번째 단계로 높은 순서대로 조밀한 인식 과정을 수행한다. 이 과정에서, 사전에 추출한 각 페이지의 SIFT 설명자와 현재 입력 영상의 SIFT 설명자간의 매칭을 수행한다. 특정 개수만큼의 인식된 특징점이 검출되지 않을 경우, 두 번째 우선 순위 페이지 영상을 검사한다. 이러한 과정을 통해, 현재 보이는 페이지의 적절한 상태를 실시간으로 파악한다.

3.2 추적 기술 방법

페이지 인식 과정을 통해, 현재 카메라의 초기 자세를 구한다. 평면과 평면의 관계는 호모그래피로 설명되어지며, 초기 직교 방향에서 획득한 페이지 영상으로 인해 카메라 내부 파라미터와 호모그래피만으로 카메라 자세를 구할 수 있다 [6]. 즉, 식 (2)는 관찰된 특징점과 저장된 페이지 특징점 간의 호모그래피를 구하는 과정과 같다.

초기 자세가 구해지면, 현재 페이지의 특징점을 지속적으로 입력 영상에서 추적하는 과정을 수행한다. 지속적인 추적과 동시에 구해지는 호모그래피로 카메라의 자세도 갱신한다. 현재 영상에서의 특징점의 위치는 Optical Flow 와 특징점 주변의 패치 정보, 그리고 추정된 현재 특징점의 위치를 이용하여 구한다.

4. 구현 및 실험 결과

본 실험에서는 제안된 알고리즘의 행렬 연산을 위해 OpenCV [14]를 사용했으며, 3 차원 콘텐츠 관리 및 증강 현실 렌더링을 위해 OpenSceneGraph [13]를 이용하였다. 구현된 추적 알고리즘은 OpenSceneGraph 의 플러그인으로 동작하도록 설계되었다. CPU 로는 Core 2 duo 2.66GHz 와 SIFT 는 GPU 구현을 활용하였다 [15].

실험을 위해 총 30 페이지의 직교 영상을 사전에 획득하였고 (그림 2), 페이지 직교 영상에서 SIFT [4]와 [2] 에서 사용된 특징점을 추출하여 저장하였다. 그림 2 는 카메라 입력 영상에 대한 페이지 인식 결과를 나타낸다.



그림 2. 제안된 페이지 인식 방법 결과: 30 여페이지로 이루어진 디지털북에서 실시간으로 페이지를 인식함 인식 결과는 Map # = '페이지'로 표기.

페이지의 인식 속도는 사전에 획득된 페이지 직교 영상의 개수와 그에 따라 증가하는 특징점의 개수에 비례하게 된다. 그림 3 은 특징점 개수 (페이지 개수)에 따른 페이지 인식 알고리즘의 속도를 동일한 연속 영상을 이용해, 측정한 결과이다. 페이지 탐색 시간에는 SIFT 특징점 추출, 탐색, 호모그래피 계산 시간이 포함되었다. 식 (3)의 $E_{\text{SSD}}(\cdot)$ 를 위해, 40×30 크기의 Blur 영상을 사용하였고, 우선 순위가 부여된 후 최대 3 개의 페이지까지 두번째 단계를 수행하였다. 그래프에서 SIFT (Heavy)는 SIFT 의 특징점 추출에 관여하는 두 인자 Octave 와 Level 값을 3 으로 설정한 것으로, 특징점의 개수를 많이 얻는다. SIFT (Light)는 상대적으로 특징점의 개수가 많지 않게 조정하였다. 페이지 인식은 배경 쓰레드 (Background Thread)에서 수행됨으로, 실질적으로 주 쓰레드 (Main Thread)의 카메라 자세 추정 과정의 수행 시간에 큰 영향이 없었다. 특히, 사전에 추출된 SIFT 특징점의 개수의 증가에 따라 탐색을 위한 설명자 간의 비교 회수가 증가하므로, 그림 3 에서의 (a), (b)에서와 같은 수행 속도 차이를 보였다. 그러나 제안된 방법은 모든 페이지에 대한 탐색을 수행하는 방식 보다는 우선 순위에 근거하였기

때문에, 특징점의 개수 증가에 상대적으로 수행 시간 상승폭이 크지 않음을 알 수 있다.

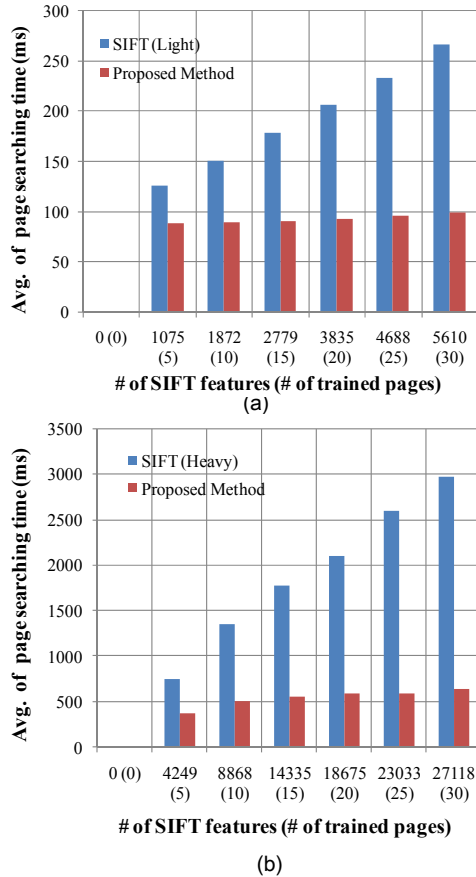


그림 3. 30 여페이지에 대한 동일 연속 영상에 대해 수행한 제안된 페이지 인식 방법과 SIFT 방법 수행 시간 비교 결과: (a) SIFT (Light) - Octave(1), Level (3) 제한 (b) SIFT (Heavy) - Octave(3), Level (3)

그림 4 는 페이지 인식후, 추적 알고리즘을 적용해 실시간으로 카메라 자세를 추정하고, 그 결과를 춤추는 아바타 증강으로 나타낸 것이다. 실험에서 패치 크기는 16×16 을 사용했으며, 25fps~30fps 의 결과를 보였다.

5. 결론

본 논문에서는 증강 현실 책 응용에서 필요한 실시간 페이지 인식 알고리즘과 마커리스 추적 기술을 제안하였다. 인식과 추적을 두 개의 멀티 코어로 분산하여, 30 여개의 페이지가 실시간으로 분별됨을 보였다. 특히, 페이지 수가 증가해도 제안된 평가 함수를 이용해 비교 회수와 수행 시간을 줄였다. 페이지 인식 후, 카메라 자세를 실시간으로 계산해 실험에서 책 위의 가상 아바타 증강을 보였다. 본 기술은 기존의 출판된 책에 마커를 붙이지 않고, 콘텐츠를 저작하여 사용할 수 있는 기반기술로 효용 가치가 크다고 할 수 있다. 향후 연구에서는 코드 최적화를 통한 성능 향상과 인식 가능한 페이지의 개수 한계에 대한 실험 및 분석이 추가적으로 진행될 것이다. 또한, 모바일 증강 현실

응용에서 사용될 수 있도록 다양한 테스트가 이루어 질 예정이다.

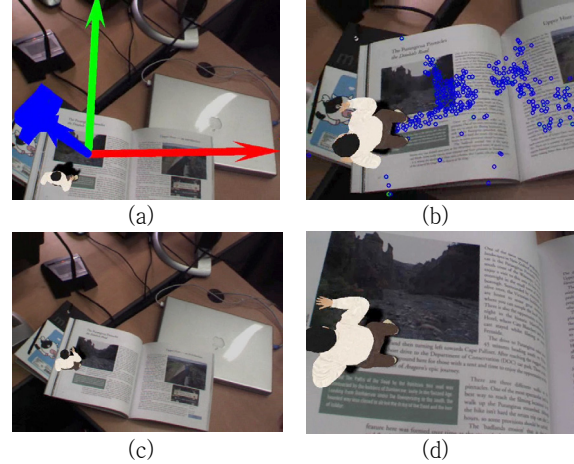


그림 4. 페이지 인식 및 카메라 추적: 책 위에 춤추는 아바타 증강 결과 (a) 페이지 인식시 증강 현실에서의 가상 좌표계 (b) 페이지에서 안정적으로 추적되는 일부 특징점들 (파란색 점들) (c) 카메라와 페이지간의 거리가 멀어질 경우 (d) 카메라가 페이지에 근접했을 경우

참고문헌

- [1] M. Billinghurst, H. Kato and I. Poupyrev, "The MagicBook - moving seamlessly between reality and virtuality", Computer Graphics and Applications, IEEE, Vol.21, pp.6-8, 2001.
- [2] V. Lepetit, P. Laguerre and P. Fua, "Randomized trees for real-time keypoint recognition", Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, Vol.2, pp.775-781, 2005.
- [3] M. Ozuysal, P. Fua and V. Lepetit, "Fast Keypoint Recognition in Ten Lines of Code", Computer Vision and Pattern Recognition, IEEE Conference on, pp.1-8, 2007.
- [4] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, Vol.60, pp.91-110, 2004.
- [5] M. Billinghurst, H. Kato and I. Poupyrev, "The MagicBook - moving seamlessly between reality and virtuality", Computer Graphics and Applications, IEEE, Vol.21, pp.6-8, 2001.
- [6] R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision", 2003.
- [7] 이영호, 하태진, 이형목, 김기영, 우운택, "디지로그 북 - 아날로그 책과 디지털 콘텐츠의 융합", 정보통신분야학회 합동학술대회, 14 권, pp.186-189, 2007.
- [8] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces", Mixed and Augmented Reality, 2007. ISMAR2007. 6th IEEE and ACM International Symposium on, pp. 225-234, Oct, 2007.
- [9] R. Castle, D. Gawley, G. Klein, and D.Murray, "Video-rate recognition and localization for wearable cameras", British Machine Vision Conf, Jan, 2007.
- [10] C. Scherrer, J. Pilet, P. Fua, and V. Lepetit, "The Haunted Book", Mixed and Augmented Reality, 2008. ISMAR2008. 7th IEEE and ACM International Symposium on, Oct, 2008.
- [11] N. Taketa, K. Hayashi, H. Kato and S. Noshida, "Virtual Pop-Up Book Based on Augmented Reality", LECTURE NOTES IN COMPUTER SCIENCE, Vol. 4558, pp.475-484, 2007.
- [12] H. Yang, K. Cho, J. Soh, J. Jung, and J. Lee, "Hybrid Visual Tracking for Augmented Books", International Conference on Entertainment Computing, 2008.
- [13] OpenSceneGraph, <http://www.openscenegraph.com>
- [14] OpenCV, <http://sourceforge.net/projects/opencvlibrary/>
- [15] SIFTGPU, <http://www.cs.unc.edu/~ccwu/siftgpu>