

GPU 를 활용한 특징 검출 및 추적의 가속화

이만희, 박인규

인하대학교 정보통신공학과

{maninara@inha.ac.kr, pik@inha.ac.kr}

요 약

본 논문에서는 GPGPU 를 활용한 효과적인 특징 검출 및 추적 방법을 제안한다. 본 논문에서 제안하는 추적 방법은 프레임간의 특징 정합 대신 특징 배열간의 정합을 수행함으로써 특정 프레임에서 연결이 끊어진 특징에 대해서도 연속적으로 특징을 추적할 수 있고 또한 매 프레임에서 Fundamental 행렬과 Homography 행렬을 추정함으로써 해당 카메라의 이동에 더욱 적합한 카메라 상관 관계를 추정할 수 있다.

1. 서론

입력 영상 또는 동영상으로부터 특징을 검출하고 영상 또는 프레임 간에 이를 정합하는 작업은 3 차원 물체를 복원하기 위한 핵심 기술로서 파노라마 영상 합성 및 3 차원 모델 생성 등 다양한 분야에서 그 수요가 증가하고 있다[1]. 특히 고속 영상 처리를 위하여 GPGPU 를 활용한 특징 검출 기술도 꾸준히 발전되어 왔다[2]. 그러나 기존 특징 검출 및 정합의 경우 인접한 두 영상 또는 프레임 간의 정합을 위하여 개발되어 왔기 때문에 연속된 동영상에서의 특징 추적에는 어려움이 있다. 따라서 본 논문에서는 입력 동영상에서 특징 검출 및 추적에 대하여 GPGPU 기술을 활용한 효과적인 가속화 방법을 제안한다. 본 논문에서는 특징 검출을 위하여 SURF[3] 기법을 적용하였으며 효과적인 특징 정합 및 추적을 위하여 GPGPU 상에서의 최적화를 수행하였다.

2. 특징 검출 및 추적

본 절에서는 특징 검출 방법 및 GPGPU 를 활용한 가속화 방법에 대하여 설명한다.

2.1 SURF 를 이용한 특징 검출

SURF 는 영상에서 타원 형태의 특징을 검출하고 크기와 회전에 강인한 기술자를 이용하여 정합을 수행할 수 있는 특징 검출 방법중의 하나로 널리 사용되고 있다. 특히 기존의 SIFT[4] 방법과 비교할 때 적분 영상을 이용하여 Hessian 행렬의 연산을 가속화 하고 행렬 크기의 변화를 이용하여 영상의 축소 연산을 피할 수 있는 장점이 있다. 본 논문에서는 위와 같은 특징 검출 기술을 적용하여 특징 검출 및 추적을 수행하였다.

2.2 특징 정합 및 추적

기존의 특징 정합의 경우 인접한 두 프레임간의 특징 정합을 통하여 특징 사이의 연관관계를 검출하였다. 그러나 카메라의 흔들림이나 물체의 이동에 의하여 영상이 흐려졌을 경우 특징 정합이 실패할 수 있고 후에 다시 해당 특징이 발견되더라도 기존에 추적된 내용과 연결되지 못하고 새로운 특징으로 검출되게 된다. 이와 같은 특징 추적의 불연속성을 방지하기 위하여 본 논문에서는 검출된 특징들을 배열에 저장한 뒤 배열간의 정합을 수행함으로써 위와 같은 상황에 의하여 특징 추적이 실패하더라도 해당 특징을 연속적으로 추적할 수 있도록 하였다. 또한 매 프레임마다 Fundamental 행렬과 Homography 행렬을 추정하고 어느 행렬이 더 적합한지 판별함으로써 보다 정확한 카메라 상관관계를 추정하였다.

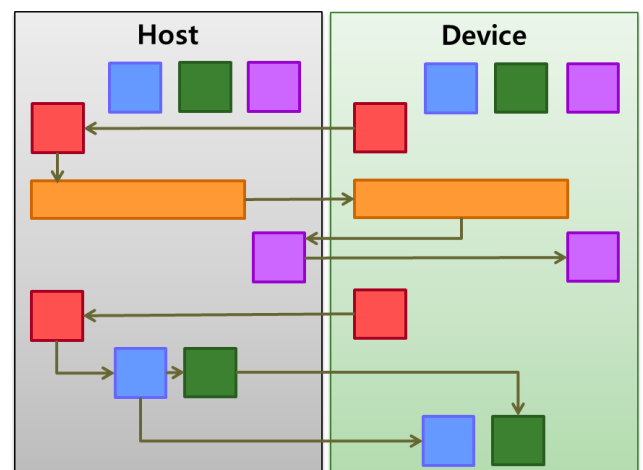


그림 1. 제안하는 시스템의 GPGPU 메모리 구조.

2.3 GPGPU 를 활용한 고속화

본 논문에서는 영상에 대한 특징 검출 및 추적을 가속화 하기 위하여 NVIDIA 의 CUDA[5] 기술을 활용하여 GPGPU 를 구현하였다. 그림 1 은 본 논문에서 제안하는 GPGPU 시스템의 메모리 구조를 보여주고 있다. 그림 1 에서 보는 바와 같이 우선 입력 프레임에 대해 SURF 를 이용하여 특징 검출을 수행한다. 검출된 특징은 기존의 정합된 특징 및 특징 후보와 정합을 수행하고 해당 정보는 CPU 메모리로 전달된다. 그 후 정합된 정보를 이용하여 정합 쌍을 생성하고 해당 정합 쌍을 이용하여 Fundamental 행렬과 Homography 행렬을 추정한다. 추정된 두 행렬을 이용하여 참 값과 거짓 값을 분류하고 참 값은 정합된 특징 배열에, 거짓 값은 특징 후보 배열에 추가한 뒤 위 과정을 모든 프레임에 대하여 반복한다.

이 때 GPU 를 활용한 최적화를 위하여 우선 적분 영상 생성의 경우 환산 기법과 분할 정복 알고리즘을 이용하여 최적화 하였다. 그리고 생성되는 스레드가 유희상태에 들어가지 않도록 스레드의 색인을 배열함으로써 GPU 의 활용을 최대화 하였다. 또한 기본적으로 루프 언롤링 기법을 적용하여 연산 속도를 향상시켰으며 블록 내부에 생성되는 스레드의 개수를 조절함으로써 블록에서의 동기화를 제거하였다.

3. 실험 결과 및 분석

본 논문에서는 3.4GHz 의 Intel Core i7 CPU 와 8GB 의 램을 장착하고 1.5GB 의 메모리를 갖고 있는 GTX 580 GPU 에서 실험을 진행하였다. 표 1 은 1280x720 의 해상도를 갖는 동영상에 대하여 특징을 검출하고 정합 및 추적을 수행한 수행시간을 보여주고 있다. 표 1 에서 보는 바와 같이 매 프레임당 평균 1,000 개의 특징을 추출하는데 CPU 에 비하여 18 배의 속도 향상을 얻을 수 있었고 GPU 의 경우 매 프레임당 약 120ms 의 수행시간을 나타냄으로써 본 논문에서 제안하는 방법의 실시간성을 확인할 수 있었다. 그림 2 는 720x576 의 medusa 동영상에 대하여 특징 검출 및 추적을 수행한 결과를 보여주고 있다.

표 1. 1280x720 동영상에 대하여 평균 1,000 개의 특징을 추출한 실험 결과.

	CPU (ms/f)	GPU (ms/f)
특징 검출	395	36.11
정합	239	42.07
카메라 추정	1492	36.37
거짓 제거	14	3.77
총 수행시간	2144	119.67

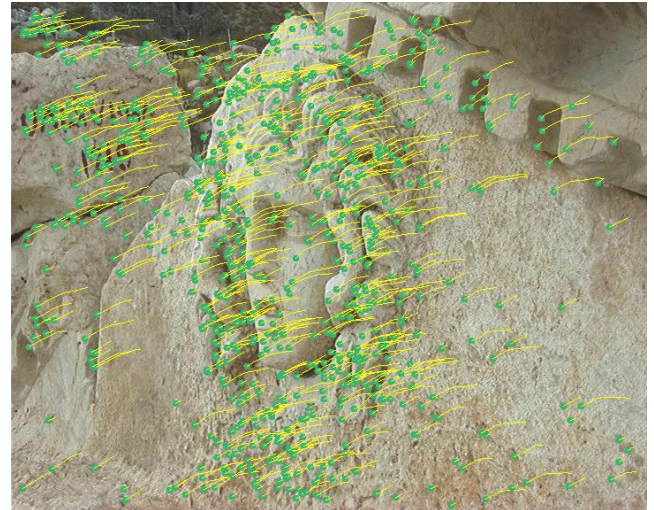


그림 2. 실험 결과 영상.

4. 결론

본 논문에서는 특징 검출 및 추적에 대하여 GPGPU 기술을 활용한 효과적인 가속화 방법을 제안하였다. 특히 프레임간의 특징 정합 대신 찾아진 특징들을 배열에 저장한 뒤 배열간의 정합을 수행함으로써 특정 프레임에서 특징 추적이 실패하더라도 뒤에 해당 특징이 발견되었을 경우 연속적으로 특징을 추적할 수 있도록 하였다. 또한 매 프레임마다 Fundamental 행렬과 Homography 행렬을 추정함으로써 해당 카메라 이동에 더 적합한 움직임을 추정할 수 있었다.

감사의 글

본 연구는 한국전자통신연구원의 지원으로 수행되었음.

참고문헌

- [1] M. Pollefeys, L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch, "Visual modeling with a hand-held camera," *International Journal of Computer Vision*, vol. 59, no. 3, pp. 207-232, September 2004.
- [2] CUDA SURF – A real-time implementation for SURF, <http://www.d2.mpi-inf.mpg.de/surf>
- [3] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346-359, June 2008.
- [4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, November 2004.
- [5] NVIDIA CUDA, <http://developer.nvidia.com/category/zone/cuda-zone>