

NVIDIA Jetson TX1 기반 실시간 Stixel 추출 기법

*이규철 **유지상

광운대학교

*gyucheol0116@gmail.com

NVIDIA Jetson TX1 based Real-Time Stixel Extraction Method

*Lee, Gyu-Cheol **Yoo, Jisang

KwangWoon University

요약

자율 주행 자동차에서 스테레오 카메라를 이용하여 실시간으로 깊이 정보를 추출하는 것은 매우 중요한 문제 중 하나이다. 널리 사용되는 방법 중에 하나인 Semi-Global Matching (SGM)은 영상에서 여러 방향에 대한 비용 함수를 이용하여 평탄한 변이 지도를 획득하는 알고리즘이다. 알고리즘의 특성 상 병렬화가 용이하기 때문에 실시간으로 구동해야 되는 어플리케이션에 자주 사용되는 알고리즘이다. 하지만 픽셀 단위로 표현되는 dense한 특성은 영상 내의 관심 객체를 추출하고 추적하기에는 부적합하다. 따라서 제안하는 기법에서는 픽셀과 객체 레벨 사이의 표현인 stixel을 이용하여 관심 객체들을 추출하고 NVIDIA에서 출시한 Jetson TX1을 이용하여 실시간으로 구동한다.

1. 서론

스테레오 비전은 가까운 미래에 자동차에서 촬영한 영상을 이해하는 데 필수적인 역할을 할 것으로 예상된다. 최근에는 정확한 물체 경계와 매끄러운 표면을 제공하는 dense한 스테레오 매칭 알고리즘인 Semi-Global Matching (SGM)이 제안되었다[1]. Middlebury 데이터 베이스[2]에 따르면 가장 강력한 10 가지 스테레오 알고리즘 중 3 가지가 현재 SGM을 변형한 것이다. 하지만 계산상의 부담, 특히 필요한 메모리 대역폭으로 인해 기존의 SGM 알고리즘은 CPU에서 구현하기 힘들다. 2015년에 NVIDIA에서 출시한 Jetson TX1[3]은 모바일 임베디드 시스템에서 고성능의 연산이 가능한 장치이다. TX1에 내장되어 있는 GPU를 이용하여 병렬 처리가 가능하기 때문에 SGM의 실시간 구현이 가능하다.

스테레오 카메라를 이용한 영상 분석의 핵심은 영상 내에서 촬영된 관심 객체를 추출하고 추적하는 것이다. 지난 수십 년간의 연구는 모바일 플랫폼에서 자동차 및 보행자를 탐지하는 데 중점을 두었다. 하지만 SGM 같이 픽셀 단위로 표현되는 dense한 변이 지도는 객체를 표현하기에는 부적합하다. 따라서 픽셀과 객체 레벨 사이의 갭을 연결하는 중간 레벨 표현인 stixel[4]을 사용한다. Stixel은 stick과 pixel의 합성어로 도로 면의 경계와 장애물의 높이를 추정하여 이를 이어주는 다수의 막대들로 전방 상황을 표현하는 방법이다. Stixel을 이용하면 dense한 변이 지도와 비교하여 정보 축약 측면에서 매우 효율적이다. 또한, stixel 형태로 검출한 객체 후보들을 인식단으로 넘기기만 하면 되기 때문에 모노 카메라의 객체 추출 기법에서 사용하는 슬라이딩 윈도우 방식보다 수행 속도가 훨씬 빠르다.

제안하는 기법에서는 모바일 임베디드 시스템에서 실시간으로

stixel을 추출하는 기법을 제안한다. 모바일 임베디드 시스템은 Jetson TX1을 이용하며 SGM과 stixel 알고리즘을 TX1에 포팅하여 실시간으로 stixel을 추출한다.

본 논문의 2 절에서는 제안하는 기법 설명 및 실험 결과를 분석하고 3 절에서는 결론을 맺는다.

2. 본론

제안하는 기법에서 SGM 첫 번째 단계인 matching cost를 계산하기 위해 Census Transform (CT)[5]를 적용한다. CT는 윈도우의 가운데 픽셀과 주변 픽셀들을 비교하여 가운데 픽셀이 큰 경우에는 1, 그렇지 않은 경우에는 0으로 맵핑하여 비트 스트림 형태로 변환한다. 일반적으로 사용하는 Block Matching (BM) 기법은 단순 윈도우의 밝기 값을 비교하기 때문에 텍스처의 품질에 영향을 많이 받는 반면에 CT는 객체의 구조적인 특성에 robust한 특성을 가지고 있다. 또한, 비트 스트림 형태로 변환한 후 Hamming Distance를 이용하여 거리를 측정하기 때문에 BM에서 SAD 방식으로 거리를 측정하는 방식에 비해 훨씬 수행속도가 빠르다는 이점을 가지고 있다.

각 화소에 대한 cost를 구한 이후에 각 방향 별로 cost를 종합하여 최적의 변이를 찾는 cost aggregation 과정[1]을 진행한다. 방향의 수는 최소 2 개에서 최대 16 개의 방향으로 계산하며 하드웨어 설계 시 메모리와 계산량에 영향을 미치기 때문에 적절한 수를 사용해야 한다. 본 논문에서는 2개와 4개의 방향으로 실험을 진행하였다. 방향 별로 각 변이에 대한 cost를 모두 더한 후 최소값에 해당하는 변이를 최종 변이로 결정한다.

제안하는 기법에서는 TX1에서 실시간으로 구동하기 위해 병렬 프

로그래밍인 CUDA[6]를 적용하였다. 그림 1은 SGM을 적용하여 변이를 추출한 영상을 보여준다.

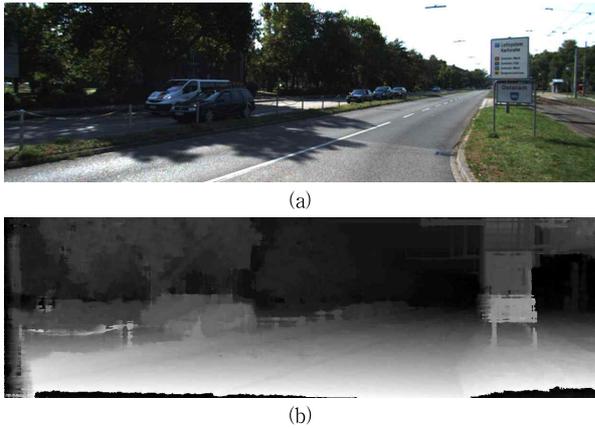


그림 1. SGM 결과
(a) 컬러 영상 (b) 변이 지도
Fig. 1. Result of SGM
(a) Color image (b) Disparity Map

변이를 이용하여 stixel을 추출하기 위해서 먼저 polar occupancy grid를 생성한다. Polar occupancy grid는 가로 축은 영상의 x좌표, 세로 축은 변이를 나타내는 영상으로써 객체와 도로면이 맞는 경계인 free space를 검출하기 위해 사용된다.

Free space의 검출은 각 열에 대해 개별적으로 threshold를 이용하여 검출한다. 장애물의 높이는 전경과 배경의 변이 사이의 최적의 segmentation을 찾아냄으로써 얻어진다. 먼저 membership voting 방법[4]을 이용하여 cost 영상을 계산한 다음 cost를 최소화하는 객체의 상단 경계를 찾는다.

Free space와 모든 열의 높이가 계산되면 계산된 높이와 사전에 지정한 폭을 이용하여 하나의 stixel을 생성한다. 그림 2는 폭이 5인 stixel을 추출한 모습을 보여준다.

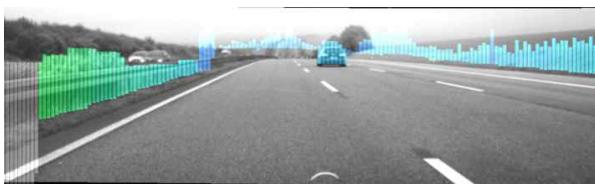


그림 2. Stixel 알고리즘 결과
Fig. 2. Result of stixel algorithm

표 1은 TX1에서 제안하는 기법을 측정된 시간을 보여준다. 640x480 해상도에서 cost aggregation을 4방향으로 하는 경우 14.8 FPS로 실시간에 못미치는 측정 결과를 보여주었다. 하지만 같은 조건에서 2방향으로 하는 경우 28.5 FPS로 실시간에 근접한 속도를 보여주는 것을 확인할 수 있다. 해상도가 320x240인 경우에는 2방향, 4방향 모두에서 실시간 이상의 FPS를 보여주었다.

표 1. TX1에서 속도 측정 결과

Table1. Speed measurement result in TX1

해상도	2-방향	4-방향
640 x 480	28.5	14.8
320 x 240	72	35.4

3. 결론

본 논문에서는 NVIDIA Jetson TX1에 SGM과 stixel 알고리즘을 실시간으로 구현하였다. SGM에서 소요되는 시간을 줄이기 위해 CT를 이용하여 matching cost를 구하였으며 CUDA를 이용하여 병렬 처리를 수행하였다. Stixel을 추출하기 위해 polar occupancy grid 방법과 free space 검출 방법을 이용하여 도로면을 제거한 후 membership voting 방법을 통해 객체의 높이를 추정하였다. 실험 결과 TX1에서 stixel 검출이 실시간으로 동작됨을 확인하였다.

Acknowledgement

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 대학ICT연구센터육성 지원사업의 연구결과로 수행되었음 (IITP-2017-2016-0-00288)

Reference

- [1] H. Hirschmuller, "Accurate and Efficient Stereo Processing by Semi-Global Batching and Mutual Information," *Computer Vision and Pattern Recognition (CVPR) 2005*, Vol. 2. 2005.
- [2] D. Scharstein and R. Szeliski, "Middlebury Stereo Vision Page," Online at <http://www.middlebury.edu/stereo2> (2002).
- [3] NVIDIA, "NVIDIA JETSON TX1," Online at <http://kr.nvidia.com/object/embedded-systems-dev-kits-modules-kr.html>.
- [4] H. Badino, U. Franke and D. Pfeiffer, "Efficient Representation of Traffic Scenes by Means of Dynamic Stixels," *Joint Pattern Recognition Symposium*, Berlin Heidelberg, 2009.
- [5] F. Stein, "Efficient Computation of Optical Flow using the Census Transform," *Joint Pattern Recognition Symposium*, Berlin Heidelberg, 2004.
- [6] Nvidia, C. U. D. A. "Compute Unified Device Architecture Programming Guide." 2007.