

자율주행을 위한 GPU 기반 OpenCV 가속화

한우림, 이강
한동대학교 전산전자공학부
e-mail : rimwoohan@gmail.com

Accelerating OpenCV based on GPU for Autonomous Driving

Woorim Han, Kang Yi
School of Computer Science and Electrical Engineering,
Handong Global University

1. 연구 필요성 및 문제점

자율주행 차량은 지속적으로 주변 상황에 관한 대량의 정보를 받고 이를 분석하여 판단과 제어에 이용해야 한다. 따라서 방대한 데이터를 빠른 시간 안에 처리하는 능력을 갖추어야 하므로 차량의 컴퓨팅 파워가 매우 중요하다.

그러므로 필요한 모든 연산을 데이터를 순차적으로 처리할 수 있는 CPU에 맡기기 보다는 연산량이 많이 필요한 부분의 경우 병렬 연산이 가능한 GPU의 활용이 적합하다. 본 논문에서는, GPU를 활용하여 컴퓨팅 속도를 향상시키기 위해 CUDA를 사용하여 OpenCV를 가속시키는 방법을 제안한다.

2. 연구내용과 방법

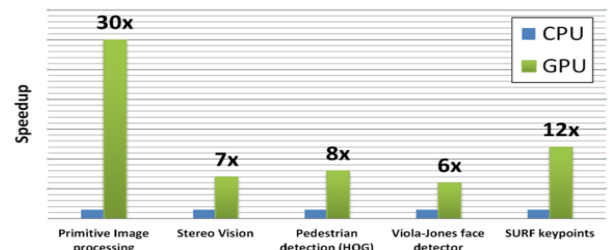
자율주행 차량에서 OpenCV를 사용해서 자율주행에 필요한 주변 환경에 대한 정보를 인식할 수 있다. 실시간으로 방대한 정보를 받아오고 필요한 연산을 수행하기 위하여 GPU를 이용한 OpenCV의 가속화가 필요하다. 본 논문은 Nvidia GPU가 탑재되어 있는 Jetson TX2 Board 환경에서 자율주행 시스템을 개발하고자 한다. 따라서 GPU를 활용하여 가속화를 하고자 할 때 CUDA를 사용하는 것이 적합하다.

	CPU	GPU(CUDA)
HEADER FILE	<opencv2/opencv.hpp>	<opencv2/gpu/gpu.hpp>
FUNCTION/NAMESPACE EX) CANNY DETECTION	Canny()	Gpu::Canny()

(표 1). OpenCV CPU vs. OpenCV CUDA

CUDA의 사용을 위해서 CUDA의 설치 및 OpenCV 기반 영상인식 소스코드를 변경해야 한다. (표 1)에서처럼 기본적으로 넣는 OpenCV 헤더파일에서 GPU 헤더파일을 추가하고 namespace Gpu를 기존 OpenCV 함수 앞에 추가해야 한다. Canny Detection을 GPU를 활용하여 연산을 할 경우, CPU보다 약 12.7배 빠르게 연산이 수행된다[1]. 또한 Tesla C2050와 Core i5-760 2.8Ghz로 다양한 연산을 비

교할 경우 (그림 1)와 같은 속도 차이가 보인다. 따라서 실제 주행에서는 한번에 많은 데이터들을 받아와 한꺼번에 연산이 이루어져야 하기 때문에 더욱 더 큰 속도차이가 있을 것으로 예상된다.



(그림 1). OpenCV performance Tesla C2050 vs Core i5-760

3. 결론 및 향후 연구

OpenCV에서 실시간으로 자율주행에 필요한 recognition을 모두 수행하려면 CPU의 컴퓨팅 능력으로는 부족하다. 이러한 문제를 해결하기 위해 CUDA를 이용하여 GPU의 활용도를 높인다면 더욱 원활한 연산이 가능할 것으로 예상된다. OpenCV로 차선 인식, 차량 인식 등을 할 때, 연산집약적인 부분을 GPU로 넘기고 남은 부분만 CPU에서 처리하여 빠른 컴퓨팅효과를 볼 수 있도록 하는 것이 목표이다. 따라서 최고의 성능향상을 위하여 어떠한 부분을 GPU로 넘기는 것이 적합한지 연구할 것이다.

감사의 글

이 논문의 연구는 과학 기술 정보 통신부와 정보 통신 기술진흥 센터(IITP)의 소프트웨어중심대학 지원사업 (2017-0-00130)의 지원을 받아 수행하였음

참고문헌

- [1]. NVIDIA. OpenCV on a GPU. Online at <http://on-demand.gputechconf.com/gtc/2013/webinar/opencv-gtc-express-shalini-gupta.pdf>
- [2]. OpenCV. CUDA. Online at <https://opencv.org/platforms/cuda.html>