Modularpipeline 보고서

전공: 컴퓨터공학과 학년: 4학년 학번: 20202106 이름: OSHIMA ASUKA

1. 자율주행 모듈공정에 대한 전체적인 흐름 설명 및 각 구현모듈에 대한 설명

Low-level Perception, Scene Parsing,Path Planning, Vehicle Control의 4개의 pipe line로 구성되어 있다. 차의 이미지를 잘라 gray scale로 변환시켜 그것부터 edge검출을 수행하고 중요한 edge를 강조한다. 차선의 포인트를 찾고 저장하고 현재 위치에서 최적 경로를 계획한다. 최적 경로를 기반으로 차량을 제어하여 커브에 가까워지면 감속, 직선에서는 가속 등의 적절하게 조작한다.

1. 본인이 구현한 모듈공정 기반 자율주행 에이전트의 행동 분석
2. HW7 lean detection

처음에 실행을 하였을 때 선이 하나만 나타나며 기대와 다른 결과가 나왔다. 여기서 def lane\_detection()함수에서 오른쪽 차선인지 왼쪽 차선인지 점을 판단을 하지 못하고 있지 않을 까 생각해 중심 위치부터 오른 쪽에 있는 것은 오른 차선으로 왼쪽에 있는 것은 왼쪽 차선으로 구분을 했다. 또 여러가지 수정을 했지만 원하는 결과는 얻지 못했다. Debug를 위한 코드를 추가하여 찾아보도록 했다. def find\_maxima\_gradient\_rowwise(self, gradient\_sum):을 실행한 결과를 이미지로 출력하여 기대하는 결과와 같은지를 확인했고 그림1와 같은 결과를 얻었고 원하는 결과를 얻을 수 있었다.

グラフィカル ユーザー インターフェイス

自動的に生成された説明

그림 1

동일하게 lane\_boundary의 초기위치가 어디인지를 확인하려고 이미지로 출력했고 그림2을 얻었다. 여기서 시작위치는 차 선상이어야 하는데 크게 다른 위치를 가리키고 있는 것을 발견했다. 이는 중간 과정에서 x,y를 반대로 처리해버리기 때문이었다. 또 cut\_size는 65로 수정했다. 이러므로 직선상은

グラフィカル ユーザー インターフェイス

中程度の精度で自動的に生成された説明

그림

원하는 차선을 그릴 수 있게 되었다. (그림3) 하지만 커브 부분에서는 그림4와 같은 결과가 되어 원하는 결과를 얻지 못했고 이는 해결하지 못했다.

グラフ

中程度の精度で自動的に生成された説明

그림 3

グラフ

自動的に生成された説明

그림

1. Hw8

Way point를 예측하고 속도를 조절하는 알고리즘을 구현했다. Way point는 차량이 따라야 하는 최적 목표위치를 표시한다. 이 점을 기반으로 적절한 속도를 계산한다. 실행 결과는 그림5이고 그림처럼 커브에서 적절한 점을 표시하고 있을 때도 있었지만 속도가 떨어지지 않고 커브를 못한 경우도 있었다. 이는 def target\_speed\_prediction함수에서 적절하게 감속하는 알고리즘이 부족하는 것 같지만 문제를 이해하고 수정하지 못했다.

グラフィカル ユーザー インターフェイス

自動的に生成された説明グラフィカル ユーザー インターフェイス, アプリケーション

自動的に生成された説明

그림