



컴퓨터 비전

과제 #7

담당교수: 김낙현 교수님

제출일자: 21.06.09

학 과: 전자공학과

학 년: 4학년

이 름: 배준성

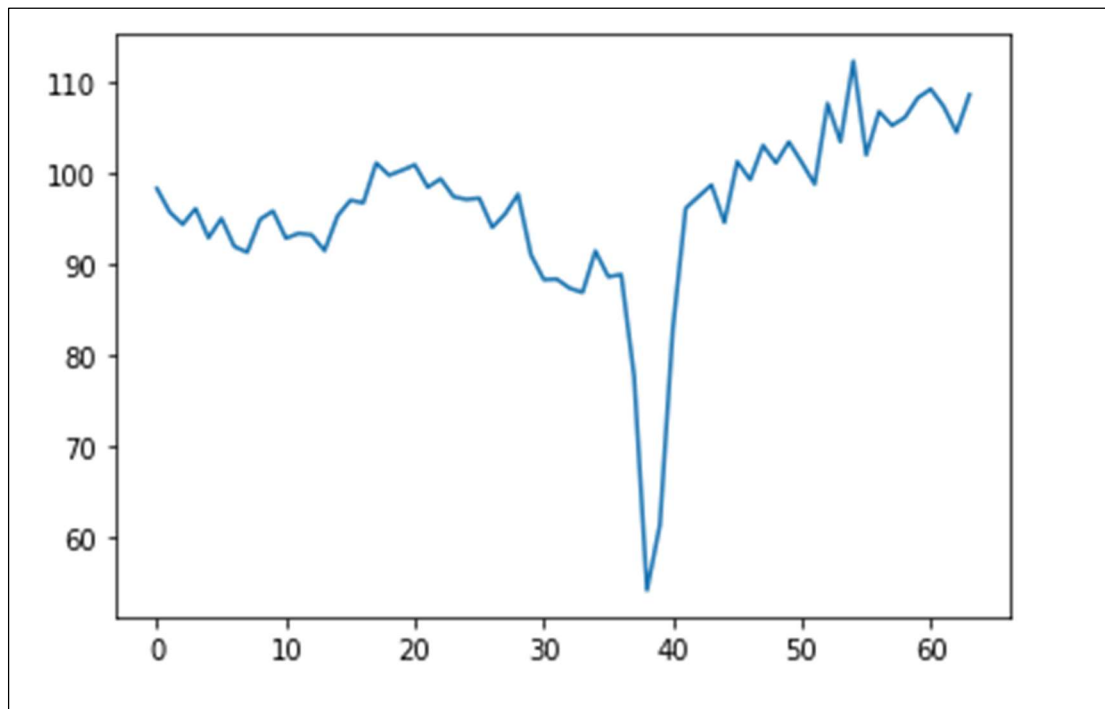
학 번: 201501487



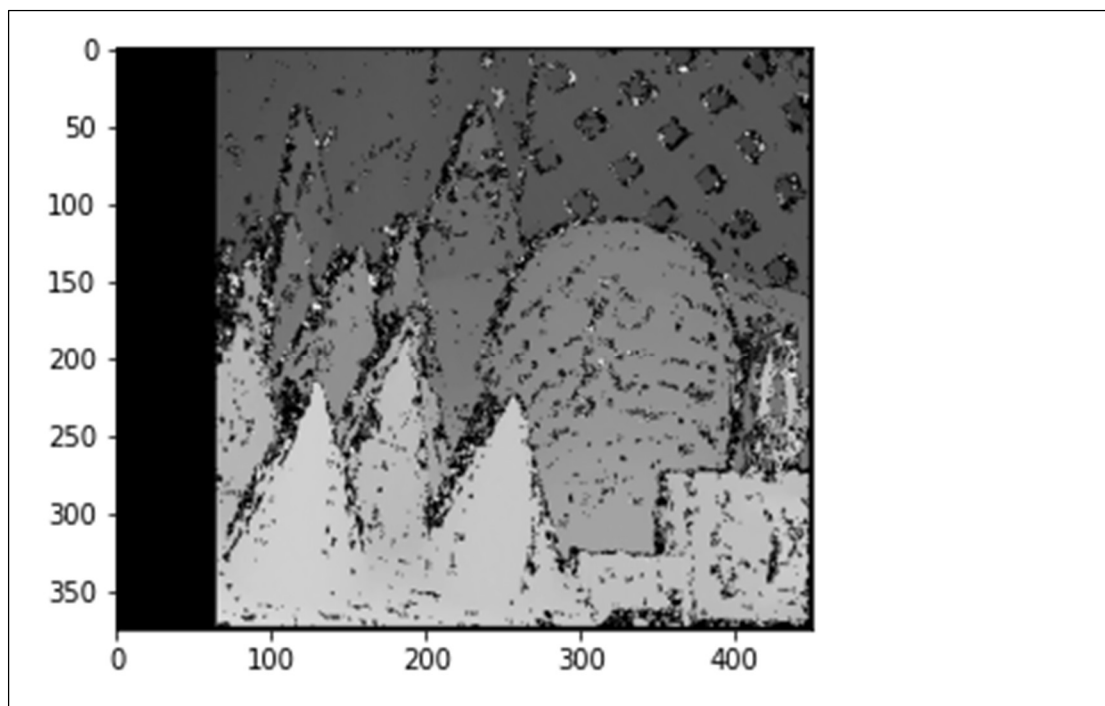
한국외국어대학교
HANKUK UNIVERSITY OF FOREIGN STUDIES

1. 실습 1~2 의 결과를 제시

실습 1 결과



실습 2 결과

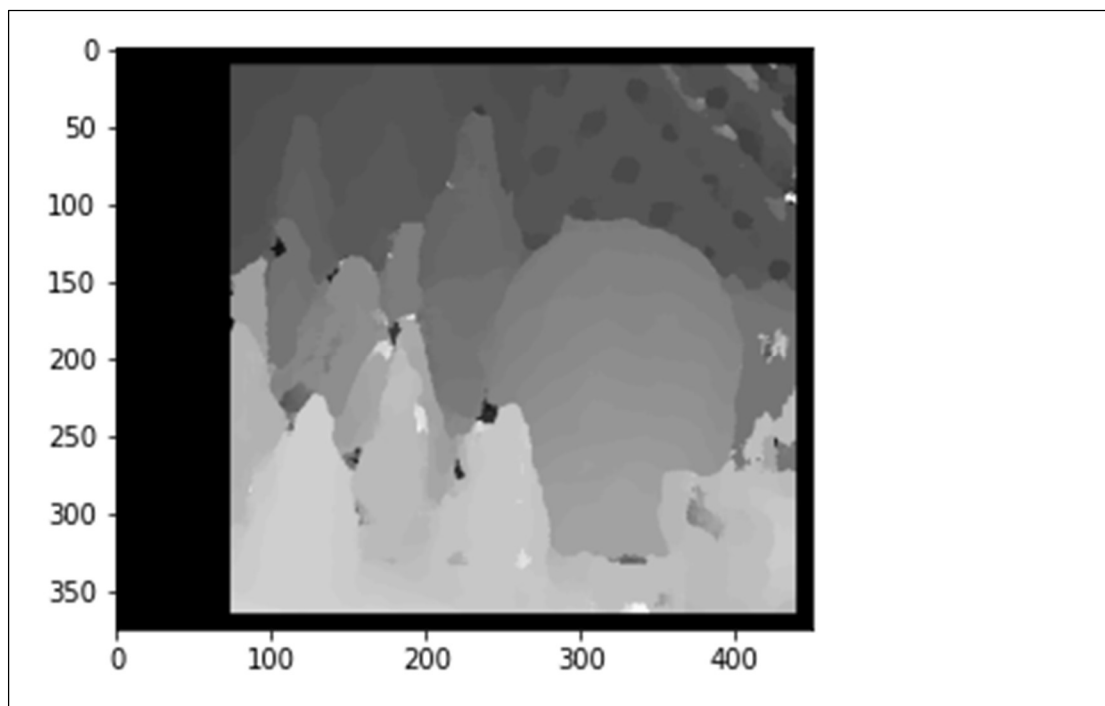


2. computeSSD 함수를 이용하여 disparityBM 을 대체하는 함수를 작성하고자 함.
depthMap 을 계산하는 함수(다음 장)를 완성하고 이 함수를 이용하여 실습 2 를 다시 수행한 결과를 제시함

완성한 코드

```
def disparityMSD(imgL, imgR, dispRange):  
    row, col = imgL.shape  
    disparityMap = np.zeros((row, col))  
  
    for y in range(11, row-10):  
        for x in range(74, col-10):  
            N = 7  
  
            ref = imgL[y-N:y+N+1, x-N:x+N+1]  
  
            MSD = np.zeros(dispRange)  
            for k in range(dispRange):  
                target = imgR[y-N:y+N+1, x-k-N:x-k+N+1]  
                Diff = ref - target  
                MSD[k] = np.mean(Diff*Diff)  
  
            disparityMap[y,x] = np.argmin(MSD)  
    return disparityMap
```

수행 결과



3. 결과를 실습 2의 결과와 비교하여 어떤 차이가 있는지 설명

DisparityBM 함수를 이용하여 결과를 출력하는 경우 SAD(Sum of Absolute Difference) 알고리즘을 이용하여 disparity를 구한다. 하지만 실습 2에서 구현한 함수는 SSD알고리즘을 이용하여 disparity를 구한다는 차이가 있다.

SAD알고리즘은 픽셀 값의 차이에 절댓값을 취한 후 합산하여 정합비용을 구한다.

반면 SSD알고리즘은 픽셀 값의 차이에 제곱을 한 후 합산하여 정합비용을 구한다.

이처럼 정합비용을 구하는 알고리즘이 다르기 때문에 최소가 되는 점이 달라지고 결국 disparity가 달라지게 된다. 이 때문에 실습 결과에 차이가 생기게 된다.

또한 수업시간에 다루었듯이 DisparityBM 함수의 경우 양쪽의 이미지를 각각 기준으로 disparity를 구하여 결과가 일치하는 점만 남기고 반면 실습 2의 함수의 경우 왼쪽 이미지 값에서 오른쪽 이미지의 값을 빼서 disparity를 구한다. 따라서 한 번만에 disparity를 구하기 때문에 DisparityBM함수에 비해서 지저분한 결과가 출력된다.