

KANGWON NATIONAL UNIVERSITY

컴퓨터비전 실습

실습6 | Moravec

실습과제 이루리 내 제출

CVMIPALAB @ KNU

실습 6-1 | Moravec

문제

주어진 코드를 활용하여 “bucks.jpg” 파일을 흑백으로 읽은 뒤,
Moravec Algorithm을 구현하세요.

요구 결과

`Moravec::FindConfidenceMap` 함수 안의 `confidence_map`에
마지막 페이지의 이미지와 같이 Moravec Algorithm를 적용하여 Edge를 검출하고 저장합니다.
결과 이미지는 “bucks_moravec.bmp” 파일로 저장합니다.
저장된 영상과 구현한 “.cpp” 총 2개의 파일을 압축하여 이ური 시스템에 제출합니다.

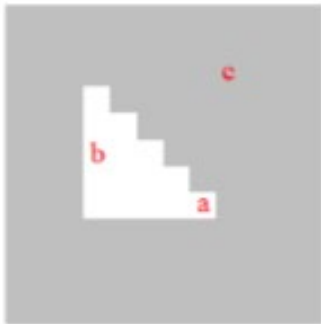
실습 6-1 | Moravec

Moravec Algorithm

$$S(v, u) = \sum_y \sum_x w(y, x) (f(y + v, x + u) - f(y, x))^2 \quad (4.1)$$

	0	1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0	0	0
4	0	0	0	1	1	0	0	0	0	0	0	0
5	0	0	0	1	1	1	0	0	0	0	0	0
6	0	0	0	1	1	1	1	0	0	0	0	0
7	0	0	0	1	1	1	1	1	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0

(a) 합성 영상



		u		
		-1	0	1
v	-1	3	4	4
	0	2	0	2
	1	4	3	2
a				
		u		
		-1	0	1
v	-1	3	1	6
	0	3	0	4
	1	3	0	3
b				
		u		
		-1	0	1
v	-1	0	0	0
	0	0	0	0
	1	0	0	0
c				

(b) 세 지점에서 $S(v, u)$ 맵

그림 4-3 $S(v, u)$ 맵

$$C = \min(S(0, 1), S(0, -1), S(1, 0), S(-1, 0))$$

모라벡 알고리즘 Confidence

실습 6-1 | Moravec

Moravec Algorithm

B에 Confidence를 계산하여 할당합니다.

$$S(v, u) = \sum_y \sum_x w(y, x) (f(y + v, x + u) - f(y, x))^2 \quad (4.1)$$

	0	1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0	0	0
4	0	0	0	1	1	0	0	0	0	0	0	0
5	0	0	0	1	1	1	0	0	0	0	0	0
6	0	0	0	1	1	1	1	0	0	0	0	0
7	0	0	0	1	1	1	1	1	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0



0	1	1
0	1	1
0	1	1

$B=f(y,x)$

	0	1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0	0	0
4	0	0	0	1	1	0	0	0	0	0	0	0
5	0	0	0	1	1	1	0	0	0	0	0	0
6	0	0	0	1	1	1	1	0	0	0	0	0
7	0	0	0	1	1	1	1	1	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0



0	1	0
0	1	1
0	1	1

$f(y-1,x)$

실습 6-1 | Moravec

Moravec Algorithm

$$C = \min(S(0, 1), S(0, -1), S(1, 0), S(-1, 0))$$

Diagram illustrating the Moravec Algorithm for a 3x3 grid. The current state is a 3x3 grid with all cells containing 1. The algorithm calculates the squared difference between the current state and four possible neighbor states, resulting in a cost value for each direction.

Top Left Calculation:

$$\left(\begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \right)^2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow 1$$

The second matrix is labeled $f(y-1, x)$.

Top Right Calculation:

$$\left(\begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \right)^2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \rightarrow 3$$

The second matrix is labeled $f(y, x-1)$.

Bottom Left Calculation:

$$\left(\begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \right)^2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow 0$$

The second matrix is labeled $f(y+1, x)$.

Bottom Right Calculation:

$$\left(\begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right)^2 = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \rightarrow 4$$

The second matrix is labeled $f(y, x+1)$.

실습 6-1 | Moravec

결과화면

