**電腦視覺**

**Homework 6**

**Contents:**

1. Description
2. Methodology
3. Code fragment
4. Image

Description

This is the report for the sixth homework for Computer Vision 2019. It is completed in full.

The environment used is Windows, with the code being written in Python 3 with the cv2 module.

The contents of this report are as follows:

1. Page 1 (this page): Description and Methodology
2. Page 2: Code fragment
3. Page 3: Resulting image. Note that the image spans the *entire* page.

Methodology

The method described in the slides is followed.

The resulting Python program, **hw6.py**, should print out the 64 x 64 matrix of Yokoi values. If the resulting picture is too unclear, redirecting the output to a file may be considered.

Note that in the code, if a coordinate is out of bounds it is given the value of -1. This has no effect on the final image.

Code Fragment

The fragment shows the calculation of the Yokoi value (with Q, R and S being macros):

1. **def** boo(a): **return** 1 **if** a **else** 0
3. **def** f(a, b, c, d):
4. **return** 5 **if** a == b == c == d == R **else** (boo(a == Q) + boo(b == Q) + boo(c == Q) + boo(d == Q))
6. **def** H(x, a, b, c):
7. **if** x != a: **return** S
8. **if** x == a == b == c: **return** R
9. **if** x == a **and** (x != b **or** x != c): **return** Q
10. **assert** 0
12. **def** yokoi(image, i, j):
13. rmax, cmax = image.shape[0], image.shape[1]
14. **def** c(r, c):
15. **return** image[r, c] **if** 0 <= r < rmax **and** 0 <= c < cmax **else** -1
16. a1 = H(c(i,j), c(i,j+1), c(i-1,j+1), c(i-1,j))
17. a2 = H(c(i,j), c(i-1,j), c(i-1,j-1), c(i,j-1))
18. a3 = H(c(i,j), c(i,j-1), c(i+1,j-1), c(i+1,j))
19. a4 = H(c(i,j), c(i+1,j), c(i+1,j+1), c(i,j+1))
20. f\_ = f(a1, a2, a3, a4)
21. **return** f\_

Image

