National Institute of Technology Karnataka
Computer Science and Engineering Department

Computer Vision (CO462) Assignment 1
Canny Edge Detection
Harris Corner Detection

Done by

Vignesh K (14CO252)

Chosen Image

Assignment_1/Input_Images/building_original.jpg

Canny Edge Detection

For this algorithm a larger version of the image is used because we need the algorithm to detect even closely spaced edges as distinct ones. If we scale down the image, the lower resolution renders these closely spaced edges as almost a uniform intensity space. The effect of using a big image can be seen in the output, where the closely spaced lines running across the window frames have also been detected as edges. The algorithm has been tested by varying the size of the **Gaussian blur (kernel_size)** and the **threshold ranges (min_val, max_val)**.

- **Varying Gaussian blur**
  For a fixed value of the **threshold range (min_val = 10, max_val = 20),** the following results were obtained.

  3 x 3 : Assignment_1/Output_Images/Canny/Gaussian_Blur/canny_output_gauss_3x3.png
  7 x 7 : Assignment_1/Output_Images/Canny/Gaussian_Blur/canny_output_gauss_7x7.png
  11 x 11 : Assignment_1/Output_Images/Canny/Gaussian_Blur/canny_output_gauss_11x11.png

  As visible from the images, increasing the kernel_size of the Gaussian blur tends to reduce the number of detected edges. An explanation for this would be that, with more blurring, the changes in the intensity values at the edge points of the image becomes more gradual or smooth. This means that the gradient magnitude computed at these points will also reduce. Consequently, some edge points which were previously detected by smaller Gaussian blurs, will now fall below the min_val of threshold and hence not be detected. Another possibility could be that these points could get disconnected from strong edge points due to intermediate edge points falling below the min_val of threshold. Hence the Hysteresis Thresholding step eliminates more edge points for a big Gaussian kernel than for a small Gaussian kernel.

- **Varying threshold ranges**
  For a fixed **kernel_size of (3 x 3) for the Gaussian blur,** the following results were obtained.

  min_val = 10, max_val = 20 :
  Assignment_1/Output_Images/Canny/Threshold/canny_output_threshold_10_20.png
  min_val = 50, max_val = 80 :
  Assignment_1/Output_Images/Canny/Threshold/canny_output_threshold_50_80.png
  min_val = 100, max_val = 150 :
  Assignment_1/Output_Images/Canny/Threshold/canny_output_threshold_100_150.png

  By varying the threshold ranges, we observe that as the magnitude of min_val and max_val increases, the number of detected edges decreases. This is naturally expected from the algorithm because higher thresholds mean that more number of edge points will fall below the range of thresholds and hence more get eliminated in the Hysteresis Thresholding step. Also, increasing the range of thresholds has the same effect because doing so accommodates more edge points in the range of thresholds and there is a greater chance that they together form a component disconnected from the strong edge points.

- **Best Result**
  The best result is obtained with a Gaussian blur size of 3 x 3 and a threshold range of min_val = 10 and max_val = 20.

  Best Result : Assignment_1/Output_Images/Canny/Best_Result/canny_output_best.png

Harris Corner Detection

For this algorithm, a scaled down version of the image has been used. The image has been reduced to 52% of its actual size before calling the algorithm. The purpose of this scale down operation is to avoid corner points from being elongated in an enlarged image. If it is elongated and the size of the chosen neighborhood is unable to subsume the point with its neighborhood, then we will end up getting a region where the variation in the intensity is low. This will render the point as a non-corner point though it actually is. In order to ensure that the variation in intensity around the point of interest is captured by the chosen neighborhood size two methods were tried out as explained below.

- **Choosing a relatively large neighborhood** : Neighborhood sizes of 11 x 11 and 15 x 15 were experimented with. But some spurious corners were detected in this approach.

- **Scaling down the image** : The image was scaled down by different factors (40%, 50%, 60% and 70%) to reduce the elongation effect of a large image. After fine-tuning a value of 52% was used. This approach outperformed the previous approach and hence was chosen.

The noise in the scaled down image was reduced by a Gaussian blur of 5 x 5.
The algorithm has been tested by varying the size of the **Harris neighborhood (n_hood), k (Harris detector free parameter)** and the **threshold (as a percentage of the maximum R-score received among all neighborhoods checked by the algorithm)**. Alos, two weight functions were experimented with, namely, the **Averaging weights (all 1s with normalization)** and **Gaussian weights.**

- **Varying the Harris neighborhood**
  For a fixed value of **k = 0.04, threshold = 0.01 x max(R-score)**, and using **Gaussian weights**, the following results were obtained.

  3 x 3 : Assignment_1/Output_Images/Harris/Neighborhood/harris_output_3x3.png
  5 x 5 : Assignment_1/Output_Images/Harris/Neighborhood/harris_output_5x5.png
  7 x 7 : Assignment_1/Output_Images/Harris/Neighborhood/harris_output_7x7.png
  9 x 9 : Assignment_1/Output_Images/Harris/Neighborhood/harris_output_9x9.png
  11 x 11 : Assignment_1/Output_Images/Harris/Neighborhood/harris_output_11x11.png

SL.No	Neighborhood Size	Number of corners detected
1.	3 x 3	10995
2.	5 x 5	14604
3.	7 x 7	15174
4.	9 x 9	15153
5.	11 x 11	15126

As the neighborhood size increases, the number of detected corners increases. This is because more points observe a high variation in intensity inside their increasing neighborhoods. The number of detected corners stagnates beyond a certain neighborhood size because this size essentially represents the largest neighborhood required to detect almost every legitimate corner point in the image. Beyond this stagnation, there is a sudden hike in the number of detected corner points which is on account of spurious corner points.

- **Varying the thresholds**
  For a fixed value of **k = 0.04, neighborhood = 3 x 3**, and using **Gaussian weights**, the following results were obtained.

0.01 x max(R-score) : Assignment_1/Output_Images/Harris/Threshold/harris_output_thresh_0.01.png
0.02 x max(R-score) : Assignment_1/Output_Images/Harris/Threshold/harris_output_thresh_0.02.png
0.03 x max(R-score) : Assignment_1/Output_Images/Harris/Threshold/harris_output_thresh_0.03.png
0.04 x max(R-score) : Assignment_1/Output_Images/Harris/Threshold/harris_output_thresh_0.04.png
0.05 x max(R-score) : Assignment_1/Output_Images/Harris/Threshold/harris_output_thresh_0.05.png
0.06 x max(R-score) : Assignment_1/Output_Images/Harris/Threshold/harris_output_thresh_0.06.png

SL.No	Fraction of max(R-score)	Number of corners detected
1.	0.01	10995
2.	0.02	5628
3.	0.03	3534
4.	0.04	2352
5.	0.05	1668
6.	0.06	1260

As expected, the number of detected corner points decreases with increasing threshold percentages because fewer eligible corner points cross the threshold barrier with each increasing value.

- **Varying the Harris free parameter (k)**
  For a fixed value of **threshold = 0.01 x max(R-score), neighborhood = 3 x 3**, and using **Gaussian weights**, the following results were obtained.

k = 0.04 : Assignment_1/Output_Images/Harris/Free_Parameter_k/harris_output_k_0.04.png
k = 0.06 : Assignment_1/Output_Images/Harris/Free_Parameter_k/harris_output_k_0.06.png
k = 0.08 : Assignment_1/Output_Images/Harris/Free_Parameter_k/harris_output_k_0.08.png
k = 0.10 : Assignment_1/Output_Images/Harris/Free_Parameter_k/harris_output_k_0.10.png
k = 0.16 : Assignment_1/Output_Images/Harris/Free_Parameter_k/harris_output_k_0.16.png
k = 0.22 : Assignment_1/Output_Images/Harris/Free_Parameter_k/harris_output_k_0.22.png

SL.No	Harris free parameter (k)	Number of corners detected
1.	0.04	10995
2.	0.06	9408
3.	0.08	7716
4.	0.10	6558
5.	0.16	3393
6.	0.22	1368

Increasing the k value causes more eligible corner points to be eliminated.

- **Varying the weights**
  For a fixed value of **threshold = 0.01 x max(R-score), neighborhood = 5 x 5**, and **k = 0.04**, the following results were obtained.

  Gaussian : Assignment_1/Output_Images/Harris/Weights/harris_output_gauss.png
  Averaging (all 1s) : Assignment_1/Output_Images/Harris/Weights/harris_output_avg.png

SL.No	Weights	Number of corners detected
1.	Gaussian	14604
2.	Averaging (all 1s)	23478

As the results suggest, the use of Gaussian weights over Averaging weights reduces the detection of spurious corner points substantially. This is due to the fact that if any corner point falls within the neighborhood of a non-corner point, then it is not directly given a weight of 1 but a weight which varies inversely with the distance from the central pixel. This reduces the contribution of a corner point to the inclusion of a non-corner point.

- **Best Result**
  The best result is obtained with **k = 0.04, threshold = 0.01 x max(R-score), neighborhood = 3 x 3,** and using **Gaussian weights** on an image scaled to **52%** of it actual size

  Best Result : Assignment_1/Output_Images/Harris/Best_Result/harris_output_best.png