CV Assignment 2

Part II

Name 1: Abdelrahman Salem Mohamed
ID 1: 6309
Name 2: Reem Abdelhalim
ID 2: 6114

Car Tracking

Implementation

1. An initial rectangle, defined by the coordinates (49,109) and (157,162), is established with the first point representing the top left corner and the second point representing the bottom right corner.

2. The initial template is retained for further computations as it is important not to rely on the previous template when computing the current frame, as it may be incorrect.

3. For each frame:

    a. A parameter list of length 6 is initialised with zeros in order to compute the affine warp matrix using the formula provided in the PDF.

    b. Image gradients in the x and y directions are obtained using a Sobel filter.

    c. The process loops until the parameter matrix converges and performs the following actions:

        i. The points of the new rectangle required for error calculation are determined by multiplying the affine warp matrix by the old points (which are retained for this purpose). The bottom right point is calculated using a fixed distance from the top left point, as the camera is fixed and thus there is no scaling.

        ii. The boundaries of the rectangle are taken into account to ensure that the rectangle does not extend beyond the frame.

        iii. The frame is warped using the parameter matrix and the region of interest is extracted from the warped frame according to the established rectangle.

        iv. The template and the warped image are unrolled in order to compute the error.

        v. The loop iterates over the region of interest to calculate the 2x6 Jacobian, 1x2 Ix and Iy for each point, and the steepest descent matrix (mx6, where m is the number of rows in the rectangle multiplied by the number of columns in the rectangle).

        vi. The Hessian matrix is calculated using the obtained steepest descent matrix and the inverse of the Hessian is calculated using the pseudo-inverse method.

        vii. vii. The delta-P matrix is determined and the list of parameters is updated in order to obtain the correct 2x3 matrix of the affine warp, which is used when the loop returns to the beginning.

viii. The norm of the delta-P matrix is used to determine when the loop will converge.

d. Once the correct parameters for the specific frame have been determined, the corresponding points of the rectangle are calculated and stored in a list.

e. The list of old points is updated with the correct points at the end of the process.

4. After all frames have been processed, the frames and their respective rectangles are converted into a video by drawing the rectangle over each frame and converting both into RGB format.

Result

Notebook

https://colab.research.google.com/drive/1R-GhOJDn5QdjXF9XwTXedJF7VJg5wn0r?usp=sharing#scrollTo=q3jxR083fDgq

Helicopter Tracking

Implementation

1. The initial rectangle is defined as the coordinates (435,90) for the top left corner and (560,130) for the bottom right corner.

2. The initial template is retained for subsequent computations as it is not dependent on the previous template.

3. For each frame:
    a. A list of length 6 parameters is initialised with zeroes in order to compute the affine warp matrix (2x3) using a specified formula.

    b. Image gradients in the x and y directions are obtained using a Sobel filter.

    c. The process of looping until the parameter matrix converges is performed, during which:

        i. The points of the new rectangle needed for error calculation are obtained by multiplying the affine warp matrix with the old points.

        ii. boundaries of the rectangle are taken into consideration to prevent it from going out of frame.

        iii. the frame is warped with the parameter matrix and the region of interest is obtained.

        iv. The template and warped image are unrolled and the error is computed.

        v. over the template size (not the region of interest), the 2x6 Jacobian, 1x2 Ix, and Iy are obtained for each point, and the steepest descent matrix mx6 is calculated.

        vi. The Hessian is then calculated, its inverse is obtained through pseudo-inverse.

        vii. delta-P is calculated, and the parameters are updated.

        viii. The norm of the delta-P matrix is used to determine when the loop will converge.

    d. The corresponding points of the rectangle are calculated and stored in a list after the correct parameters have been determined.

    e. The list of old points is updated with the correct points at the end.

4. After finishing all of the frames, We convert the frames alongside their rectangles into a video by drawing the rectangle over each frame and converting them both into RGB.

Result

Notebook

https://colab.research.google.com/drive/1BV4bX7KzzGqaPbN7-vYBmx7wPLMlgVJG?usp=sharing