CMSC 491/691: Computer Vision **Spring 2024**

Lecture 1: 2024-02-12 Image Features

Lecturer: Tejas Gokhale *Scribe: Jabril Hall*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

1.1 Recap of Last Lecture

In the last lecture, we explored more about Fourier Transforms. We explored 2 types of Discrete Fourier transforms (DFT) , the Magnitude DFT and the Phase DFT. The Magnitude DFT gives us information about the intensity of the Image. While the Phase DFT gives us information about the edges in the image. We also studied a new way to combine images using a Laplacian Pyramid: ***Combination method with Laplacian Pyramid formula:***

$$l_k = l_k^A * m_k + l_k^B * (1 - m_k) \tag{1.1}$$

1.2 Image Features

1.2.1 Definitions

Definition 1.1 ***Feature****: Points on images that are unique, important, and unusual. You can use them to map one image to another.*

Definition 1.2 ***Invariant Local Features****: Features that can be detected regardless of* ***Geometric*** *or* ***Photometric*** *changes.*

Definition 1.3 ***Harris Corner Detector****: An algorithm that sweeps a window across an image and measures changes in intensity to find the corners of the image. It saves those corners as features.*

1.2.2 Features

- Features Allow us to map images together, count objects, and stitch images.
- We ideally want Invariant Local Features.

1.2.3 Harris Corner Detector

- This algorithm gives pixels on an image a rating. Flat regions get the lowest, lines get an average rating, and corners get the highest.
- The scale it rates the pixels on is the regions error value given by this function:

$$E(u,v) = \sum_{(x,y)\in W} [I(x+u, y+v)] - I(x,y)^2 \tag{1.2}$$

The Error value function is hard to calculate at a large scale so a work around was made. Using T-series expansion you can derive this equation:

$$E(u,v) = Au^2 + Cuv + Buv + Dv^2 \tag{1.3}$$

Which can become:

$$E(u,v) = [uv][\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}][\begin{smallmatrix} u \\ v \end{smallmatrix}] \tag{1.4}$$

By finding the eigan vectors of the ABCD matrix we have a faster way to solve for the error. If one eigan vector is significantly bigger than the other, the pixels are an edge. If they are both small, it is a flat surface. Finally if they are both large, it is most likely a corner.

1.2.4 Issue With Harris Corner Detector

The main issue with the Harris Corner Detector is that, while the features it detects stay the same regardless of rotation or brightness, they do not hold when the image is scaled. To solve this we get a Laplacian pyramid for the image and run the algorithm and run the Corner Detector on each level of the pyramid. Then, to get the final set of features, we get the max each window earned during the iteration.