CSE 252A: Corner Detection

Lecturer: David Kriegman

Scribed by Owen Jow on November 01, 2018

1 Corner Detection

A corner is a location in an image around which there are 2+ dominant gradient directions. This is recognizable because if you look at the corner through a window and then slightly shift the window in any direction, its content will change significantly – as opposed to the case of a flat region, where there'll be no change, or an edge region, where there'll be no change for a shift in the edge direction.

Since they are distinctive, corners make good points of interest. If we want to independently identify points in different images and then match them, corners would be good points to identify.

Identification More concretely, a corner is a point at the center of a window in which there are two strong but different gradient directions. In the window, we want to have a distribution of large gradients in multiple directions. We can characterize this in terms of two numbers by fitting a zero-centered ellipse around the (dx, dy) gradient plot and looking at the major and minor axis lengths.

We can fit an ellipse to the gradients around (x, y) by computing the below matrix of second moments:

$$C(x,y) = \begin{bmatrix} \sum_{x,y \in W} I_x^2 & \sum_{x,y \in W} I_x I_y \\ \sum_{x,y \in W} I_x I_y & \sum_{x,y \in W} I_y^2 \end{bmatrix}$$

where I_x, I_y are image derivatives w.r.t. x and y respectively and we sum over a small window W. C arises as part of a quadratic approx. to the surface representing SSD window intensity change.

As a 2×2 symmetric matrix, C can be factored via eigendecomposition as

$$C = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

where R is an orthogonal (rotation) matrix. This can be viewed as an ellipse with

- axis lengths $\sqrt{\lambda_1}$ and $\sqrt{\lambda_2}$ [1]
- orientation determined by R (specifically, eigenvectors are axis directions)

λ_i describes the amount of increase in the corresponding eigenvector direction. The $\lambda_{\max}$ eigenvector is the shift direction with the largest increase in window intensity; likewise, the $\lambda_{\min}$ eigenvector is the shift direction with the smallest increase in window intensity.

For corner detection we don't have to look at the directions, just the eigenvalues.

[1] Alternatively $1/\sqrt{\lambda_1}$ and $1/\sqrt{\lambda_2}$; however, this will create an ellipse which is thin in the direction of faster change and wide in the direction of slower change.

If the eigenvalues are similar and both large, we have a corner because there are strong image changes in different directions. If one eigenvalue is much greater than the other, we have an edge because there are strong image changes in *one* main direction. If both eigenvalues are small, it's considered a "flat" region: intensity isn't going to change much no matter which direction we shift the window.

Accordingly, we can summarize C as a single corner response value which is defined in terms of the eigenvalues, and which encodes the idea that we want λ_1 and λ_2 to be large and similar.

For example,

$$\lambda_1\lambda_2 - \alpha(\lambda_1 + \lambda_2)^2 = \det(C) - \alpha \cdot \mathrm{tr}(C)^2$$

for α some constant (often in the range $[0.04, 0.06]$). **(Harris)**

Another option is to use the smaller eigenvalue $\lambda_{\min}$ as a response value. **(Tomasi)**

Finally, we'll take a corner to be a point which is a local maximum in its neighborhood in terms of corner response value, under the condition that the response value is greater than a threshold τ.

1.1 The Full Process: *Shi-Tomasi Corner Detection*

The full process comes together as

1. Filter image with a Gaussian to remove noise.
2. Compute the gradient at every point.
3. Compute C at each point (based on a window around each point).
4. Compute the eigenvalues λ_1 and λ_2 of each C.
   (a) If $\lambda_{\min} > \tau$, and the point has the largest $\lambda_{\min}$ in its region, call it a corner.

with parameters σ (for the Gaussian filter), window size, and τ.