detecting corners

rapid changes of image intensity in two directions within a small region.

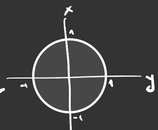
- Detect discontinuities → derivatives
  → convolve the image with derivative filters

Visualizing Quadratics

equation of a circle ⇒ $1 = x^2 + y^2$
equation of a bowl ⇒ $f(x,y) = x^2 + y^2$

if you slice a bowl at $f(x,y) = 1$
→ $f(x,y) = x^2 + y^2$

$f(x,y) = [x\ y] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$



- if you increase coefficient of x $\left\{ [x\ y] \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \right.$
  → decrease within y!
  smash the bowl in x dimension
- if you increase coefficient of y $\left\{ [x\ y] \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \right.$
  → decrease within x!
  you smash the bowl in y dimension

Singular Value Decomposition

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \underbrace{\overbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}^{\text{eigenvectors}}}_{\text{axis of the ellipse slice}} \underbrace{\overbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}^{\text{eigenvalues along diagonal}}}_{\text{inverse sqr of length of the quadratic along the axis}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^T$$

Error function for Harris Corners

$E(u,v) = [u\ v]\, M \begin{bmatrix} u \\ v \end{bmatrix}$

$M = \sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$

Detecting Edges

- we calculate and plot image gradients
  → distribution reveals edge orientation and magnitude.
- And then fit ellipse to the distribution
  → centered at the origin.
  → based on λ_1 (semi-major axis) and λ_2 (semi-major axis) we're going to classify the region

Finding Corners

1 - Compute image gradients over small region
→ $I_x = \frac{\partial I}{\partial x}$, $I_y = \frac{\partial I}{\partial y}$

2 - Substract mean from each image gradient
→ to make the distribution centered

3 - Compute the covariance matrix

4 - Compute eigenvectors and eigenvalues.

- $\underbrace{M e}_{\text{eigenvector}} = \underbrace{\lambda}_{\text{eigen value}} e$

- $(M - \lambda I)e = 0$

compute $\overbrace{\det(M - \lambda I)}^{\text{polynomial}}$
find roots of polynomial $\underbrace{\det(M - \lambda I) = 0}_{\text{eigenvalues}}$
for each eigenvalue solve $\underbrace{(M - \lambda I)e = 0}_{\text{eigenvectors}}$

5 - Use threshold on eigenvalues to detect corners.

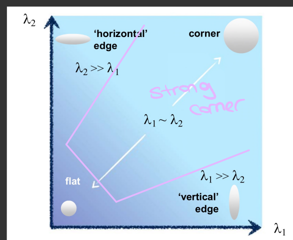
Bilinear Approximation

$E(u,v) = [u, v]\, M \begin{bmatrix} u \\ v \end{bmatrix}$

→ $M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$

⇒ The surface of $E(u,v)$ is locally approximated by a quadratic form

Harris Corner Detector



$R = \underbrace{\lambda_1 \lambda_2}_{\det(M)} - k\underbrace{(\lambda_1 + \lambda_2)^2}_{\text{trace}(M)}$

$= \det(M) - k\, \text{trace}^2(M)$

$\det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = ad - bc$ $\quad$ $\text{trace}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = a + d$

1 - compute x and y derivatives of image
$I_x = G_\sigma^x * I$ $\quad$ $I_y = G_\sigma^y * I$

2 - Compute products of derivatives at every pixel.
$I_{x^2} = I_x . I_x$ $\quad$ $I_{y^2} = I_y . I_y$ $\quad$ $I_{xy} = I_x . I_y$

3 - Compute the sums of the products of derivatives at each pixel
$S_{x^2} = G_{\sigma'} * I_{x^2}$ $\quad$ $S_{y^2} = G_{\sigma'} * I_{y^2}$ $\quad$ $S_{xy} = G_{\sigma'} * I_{xy}$

4 - Define the matrix at each pixel
$M(x,y) = \begin{bmatrix} S_{x^2}(x,y) & S_{xy}(x,y) \\ S_{xy}(x,y) & S_{y^2}(x,y) \end{bmatrix}$

5 - Compute the response of the detector at each pixel
$R = \det M - k(\text{trace}\, M)^2$

6 - Threshold on value of R; compute non-max suppression.

- Harris corner response → invariant
  - image rotation
  - intensity changes
  → not invariant
  - scale

Scale selection

```
For each level of the Gaussian pyramid
    compute feature response (e.g. Harris, Laplacian)

For each level of the Gaussian pyramid
    if local maximum and cross-scale
    save scale and location of feature   (x, y, s)
```