# Convolution and Filters

$$\text{Gaussian Kernel} \implies \frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}$$

$$\text{Sobel Kernel} \implies G_x = \begin{bmatrix}1 & 0 & -1\\ 2 & 0 & -2\\ 1 & 0 & -1\end{bmatrix}, G_y = \begin{bmatrix}1 & 2 & 1\\ 0 & 0 & 0\\ -1 & -2 & -1\end{bmatrix}$$

$$Sharpen \implies \begin{bmatrix}0 & -1 & 0\\ -1 & 5 & -1\\ 0 & 5 & 0\end{bmatrix}$$

- Zero padding → Pad edges of matrix with all 0's
- Reflection padding → Reflect end of matrix
- Repeated Cyclic → $[|y \quad z|a \quad b \quad ... \quad y \quad z]$
- Repeated Edge
- Box filters cause artifacts → Gaussian filters better for images
- Correlation is not associative $\implies$ cannot combine filters

## Projective Geometry

- Pinhole Model
  - Projection from 3D onto 2D plane
  - Vanishing point at infinity $\implies$ line is in plane parallel to camera plane
    $$y = \frac{fY}{Z}, x = \frac{fX}{Z}$$
  - Orthographic Projection keeps parallel lines

Given 4 points on an image find the vanishing lines

$$A = (2,3), B = (5,6), C = (10,15), D = (11,17)$$

Convert to 3D coordinates using $z = f$

$$A = (2,3,1), B = (5,6,1), C = (10,15,1), D = (11,17,1)$$

Find $V_1$ by finding intersection of $\overline{AB}$ and $\overline{BC}$ $\overline{AB} = \vec{a} \times \vec{b}$

$$\begin{vmatrix}a_x & a_y & a_z\\ b_x & b_y & b_z\\ i & j & k\end{vmatrix} = i \cdot \underbrace{\begin{vmatrix}a_y & a_z\\ b_y & b_z\end{vmatrix}}_{\text{x}} - j \cdot \underbrace{\begin{vmatrix}a_x & a_z\\ b_x & b_z\end{vmatrix}}_{\text{y}} + k \cdot \underbrace{\begin{vmatrix}a_x & a_y\\ b_x & b_y\end{vmatrix}}_{\text{z}}$$

Alternate Cross Product calculation

$$c_x = a_y \cdot b_z - a_z \cdot b_y, c_y = a_z \cdot b_x - a_x \cdot b_z, c_z = a_x \cdot b_y - a_y \cdot b_x$$

Repeat cross product of $\vec{c}$ and $\vec{d}$ to get $\overline{CD}$

$$\overline{AB} = (-3,3,-3), \overline{CD} = (-,-,-)$$

Cross Product of two lines → Vanishing Point

## Geometric Transformations

$$\text{Projective/Homography (8DoF/4pts)} \implies \begin{bmatrix}h_{11} & h_{12} & h_{13}\\ h_{21} & h_{22} & h_{23}\\ h_{31} & h_{32} & h_{33}\end{bmatrix}$$

Invariants - Colinearity, Cross Ratios

$$\text{Affine (6DoF/3pts)} \implies \begin{bmatrix}a_{11} & a_{12} & t_x\\ a_{21} & a_{22} & t_y\\ 0 & 0 & 1\end{bmatrix}$$

Invariants - Parallelism, Area Ratios, Length Ratios

$$\text{Similarity (4DoF/2pts)} \implies \begin{bmatrix}sr_{11} & sr_{12} & t_x\\ sr_{21} & sr_{22} & t_y\\ 0 & 0 & 1\end{bmatrix}$$

Invariants - Angles, Length Ratios

$$\text{Euclidean/Translation (3DoF/2pts)} \implies \begin{bmatrix}r_{11} & r_{12} & t_x\\ r_{21} & r_{22} & t_y\\ 0 & 0 & 1\end{bmatrix}$$

Invariants - Angles, Length Ratios

- Solve for homography given 4 points

$$\begin{bmatrix}\prime x_1\\ \prime y_1\end{bmatrix} = \begin{bmatrix}m_1 & m_2\\ m_3 & m_4\end{bmatrix} + \begin{bmatrix}t_1\\ t_2\end{bmatrix}$$

$$\begin{bmatrix}x_1 & y_1 & 0 & 0 & 1 & 0\\ 0 & 0 & x_1 & y_1 & 0 & 0\\ ... & ... & ... & ... & ... & ...\end{bmatrix}\begin{bmatrix}m_1\\ m_2\\ m_3\\ m_4\\ t_1\\ t_2\end{bmatrix} = \begin{bmatrix}\prime x_1\\ \prime y_1\\ \prime x_2\\ \prime y_2\\ .\\ .\end{bmatrix}$$

## RANSAC

RANSAC process:

1. Select n feature pairs at random, n = points required for transform
2. Compute affine transform $T$ - n points $\implies$ solve system for $T$
3. Computer inliers → distance ¡ thresh
4. Keep largest set of inliers
5. Recompute $T$ using all inliers → find $T$ using Least Squares

RANSAC iterations: For each iteration, probability of outlier is

$$p = 1 - (1 - p_{outliers})^n, n = \text{points required}$$

After $i$ iterations, probability of at least 1 iteration having zero outliers:

$$1 - p^i = c, c = \text{required confidence}$$

Solve above for $i$ or find number of iterations $N$ with required probability $p$ and outlier ratio $e$, and points required $n$

$$N = \frac{log(1-p)}{log(1-(1-e)^n)}$$

# Image Features

Harris Corner Detector

1. Compute image derivatives $I_x, I_y$ by convolving I with derivatives of Gaussian filter
2. Form Harris Matrix
   $\implies H(x,y) = \sum_{u,v} g(u,v)\begin{bmatrix}I_x^2 & I_xI_y\\ I_xI_y & I_y^2\end{bmatrix}$
3. Select points such that $K(x,y) = \frac{det(H)}{trace(H)}$ is above a threshold
4. Perform non-maximal suppression

Invariant means that the value of the descriptor shouldn't change when the image undergoes a change in appearance (brightness, rotation, etc.) This is useful for matching features between different images of the same scene which may look slightly different. SIFT Process:

1. Compute Corners
2. Adaptive Non Maximal Suppression
3. Compute feature descriptor at each point
   (a) Take 40 px window centered on point
   (b) Downsample to 8x8
   (c) Blur
   (d) Convert to 64x1 array
4. Compute SSD for each pair of points → keep matches where ratio between first best and second best is low
5. RANSAC to compute Homography

ANMS → Ensure points are equally distributed across image. Pick regional maxima.

## Color Spaces

$L^*a^*b^*$ → lightness, green - red, blue - yellow. Change in value = change in importance
HSL → hue, saturation, lightness
Why use different spaces? May be able to identify patterns more invariantly in some spaces vs. others

## Optical Flow

Vector field of pixel motion between images
Pencil - All lines through a single point
Ego Motion - How the system itself is moving
Rotation $\implies$ Flow becomes tangential to concentric circles around axis of rotation
Rotation around horiz/vertical axis with velocity in that direction $\implies$ tangential to concentric hyperbolas that limit to opposite
Aperture Problem $\implies$ Only information of a line moving through a hole is motion perpendicular to line
$I(x,y,t) = I(x,y,t) + \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t}\frac{dt}{dt}$
Low texture regions (*i.e. sky*)

- Gradients have small magnitudes
- Small $\lambda_1$, small $\lambda_2$

High texture regions (*i.e. ground*)

- Gradients are different, large magnitudes
- Large $\lambda_1$, large $\lambda_2$

Equations: Rotational Vector = $\begin{bmatrix}\alpha\\ \beta\\ \gamma\end{bmatrix}$ Translational vector = $\begin{bmatrix}u\\ v\\ w\end{bmatrix}$

$$u(x,y) = \frac{-U + xW}{Z}, v(x,y) = \frac{-V + yW}{Z}$$

Flow will be along lines that pass through the camera center Flow values increase as you move further away from the origin since they are linear with $x$ and $y$
Rotational flow equations:

$$\dot{x} = \alpha xy - \beta(1+x^2) + \gamma y,\ \dot{y} = \alpha(1+y^2) - \beta xy + \gamma x$$

Sum of translational and rotational optical flow

$$u = \frac{W}{Z}(x - \frac{U}{W}) + \alpha xy - \beta(1+x^2) + \gamma y$$

$$v = \frac{W}{Z}(y - \frac{V}{W}) + \dot{y} = \alpha(1+y^2) - \beta xy + \gamma x$$

Find flow from vector field:
Only have W and $\gamma$ → Moving straight and rotating around axis of translation
Derotate → Check if flow makes a pencil

## GMM

Multiple Gaussians to cover more varying color distributions
Steps: 1. Initialize $\pi, \mu, \Sigma$ to random. 2. Alternate between expectation (eval model,assign points to clusters) and Maximization step (eval best parameters to fit points) 3. Estimate posterior with calculated $\pi, \mu, \Sigma$