Taha Khudher(takh2100) week 5 PM

Tracking and Optical Flow

Code

5.1 Tracking Algorithms

1.

Algorithm	Applications	Pros	Cons
Kalman Filter [1]	Best for objects whose motion can be approximated as linear with gaussian noise	Computationally efficient making it capable to work in real-time applications. Provides smooth predictions	Assumes linear dynamics and Gaussian noise. Not robust against unexpected changers, occlusions or non linear motion.
Particle Filter [2]	Suitable when motion is non-linear or non-Gaussian and when there are significant uncertainties	Can model complex, multimodal, distributions, More robust under occlusions and unexpected motion changes	Computationally more expensive. May suffer from sample degeneracy if not enough particles are used.
Mean-Shift [3]	Effective for tracking objects based on their color or intensity distribution, especially when the objects appearance is distinct relative to the background.	Simple and fast Camshift adapts to scale and rotation changes.	Sensitive to background clutter and similar colors. May drift when the target appearance change or during fast motion.
Template Matching [4]	Applicable when the objects appearance is largely constant and the objects does not undergo significant scale, rotation or illumination changes	Easy to implement Directly compares image regions using similarity measures.	Sensitive to changes in scale rotation and lightning. Computational cost increases with search area size.

Optical FLow [5]	Best for tracking small displacements between consecutive frames in textured areas.	Accurate for small motions Can provide dense or sparse motion information.	Suffers from the aperture problem meaning that only the motion in the gradient direction is measurable Sensitive to noise and large displacements, relies on brightness constancy
Feature-Based Tracking [6]	Useful when the scene contains distinctive features, often used for motion estimation, pose tracking or 3 reconstruction	Robust to moderate changes in viewpoint and scale. Works well in feature-rich environments	Requires reliable detection and matching often used with outlier rejection like ransac. Can be computationally intensive if many features are tracked.

2.

Tracking Objects:

Tracking objects using color-based methods such as mean-shift or camshaft involves modeling the targets color or intensity distribution as a whole. These methods are effective when objects exhibit a distinct consistent appearance and are simple and fast to implement. However they are vulnerable to illumination changes, shadows and similar backgrounds and may lose track if the objects color varies.

Feature tracking:

Feature-based tracking identifies distinctive keypoints such as corners or patches using techniques like optical flow or descriptor matching. This approach is typically more robust to partial occlusions scale and viewpoint changes . Though it depends on the presence of sufficient features and can be sensitive to noise often requiring additional steps such as outlier detection.

5.2 Optical flow (theory)

5.2 Lucas-kandde

Brightness Constancy equation:

$I_x u + I_y v = -I_t$ mutiple pixels creates a system of equations:

$Ax = b$ Where:

A = A matrix containing the spatial gradients I_x, I_y.
b = Vector containing negative temporal gradients $-I_t$
x = Vector representing the optical flow we solve for
$x = (u, v)$

We solve for x using: $x = (A^T A)^{-1} A^T b$

$I_x = \frac{\partial I}{\partial x}$ filter kernel -1 0 1

$$I_x = \begin{bmatrix} (-1+0+2) & (-1+0+2) & (-2+2+0) \\ (-1+0+5) & (-7+0+5) & (-5+0+5) \\ (-1+0+4) & (-5+0+4) & (-4+0+4) \end{bmatrix} =$$

$$I_x = \begin{bmatrix} 1 & 1 & 0 \\ 4 & -2 & 0 \\ 3 & -1 & 0 \end{bmatrix} \qquad I_t = \begin{bmatrix} 0 & -1 & -1 \\ -6 & -4 & -3 \\ -4 & 3 & 1 \end{bmatrix}$$

$$I_y = \begin{bmatrix} 6 & 4 & 4 \\ 4 & 2 & 2 \\ -2 & 0 & -1 \end{bmatrix}$$

$Ax = b$ where:

$$A = \begin{bmatrix} 1 & 6 \\ 1 & 4 \\ 0 & 4 \\ 4 & 4 \\ -2 & 2 \\ 0 & 2 \\ 3 & -2 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \quad x = \begin{bmatrix} u \\ v \end{bmatrix} \quad b = \begin{bmatrix} 0 \\ -1 \\ -1 \\ -6 \\ -4 \\ -3 \\ -4 \\ 3 \\ 1 \end{bmatrix}$$

$$A^T A \hat{x} = A^T b$$

$$A^T A = \begin{bmatrix} \sum_{p \in P} I_x^2 & \sum_{p \in P} I_x I_y \\ \sum_{p \in P} I_y I_x & \sum_{p \in P} I_y^2 \end{bmatrix}$$

$$A^T b = -\begin{bmatrix} \sum_{p \in P} I_x I_t \\ \sum_{p \in P} I_y I_t \end{bmatrix}$$

$$\sum_{P\in P} I_x^2 = 1^2 + 1^2 + 0^2 + 4^2 + (-2)^2 + 0^2 + 3^2 + (-1)^2 + 0^2 = 32$$

$$\sum_{P\in P} I_x I_y = (1\cdot 6) + (1\cdot 4) + (0\cdot 4) + (4\cdot 4) + (-2\cdot 2) + (0\cdot 2) + (3\cdot -2) + 0 + 0 = 16$$

$$\sum_{P\in P} I_y^2 = 6^2 + 4^2 + 4^2 + 4^2 + 2^2 + 2^2 + (-2)^2 + 0^2 + 1^2 = 97$$

$$\sum_{P\in P} I_x I_t = (0\cdot 1) + (1\cdot -1) + (0\cdot -1) + (4\cdot -6) + (-2\cdot -4) + (0\cdot -3) + (3\cdot -4) + (-1\cdot 3) + (0\cdot 1) = -32$$

$$\sum_{P\in P} I_y I_t = (6\cdot 0) + (4\cdot -1) + (4\cdot -1) + (4\cdot -6) + (2\cdot -4) + (2\cdot -3) + (-2\cdot -4) + (0\cdot 3) + (1\cdot 1) = -39$$

$$A^TA\tilde{x} = A^Tb$$

$$\begin{bmatrix} 32 & 16 \\ 16 & 97 \end{bmatrix} \cdot \begin{bmatrix} U \\ V \end{bmatrix} = - \begin{bmatrix} -32 \\ -39 \end{bmatrix}$$

$$x = (A^TA)^{-1} A^Tb$$

$$(A^TA)^{-1} = \frac{1}{Det(A^TA)} \begin{bmatrix} 97 & -16 \\ -16 & 32 \end{bmatrix}$$

$$Det(A^TA) = 32\cdot 92 - 16\cdot 16 = 3104 - 256 = 284$$

$$(A^TA)^{-1} = \frac{1}{2848} \begin{bmatrix} 97 & -16 \\ -16 & 32 \end{bmatrix} = \begin{bmatrix} 0.0341 & -0.0056 \\ -0.0056 & 0.0112 \end{bmatrix}$$

$$X = \begin{bmatrix} 0.0341 & -0.0056 \\ -0.0056 & 0.0112 \end{bmatrix} \cdot \begin{bmatrix} 32 \\ 39 \end{bmatrix} =$$

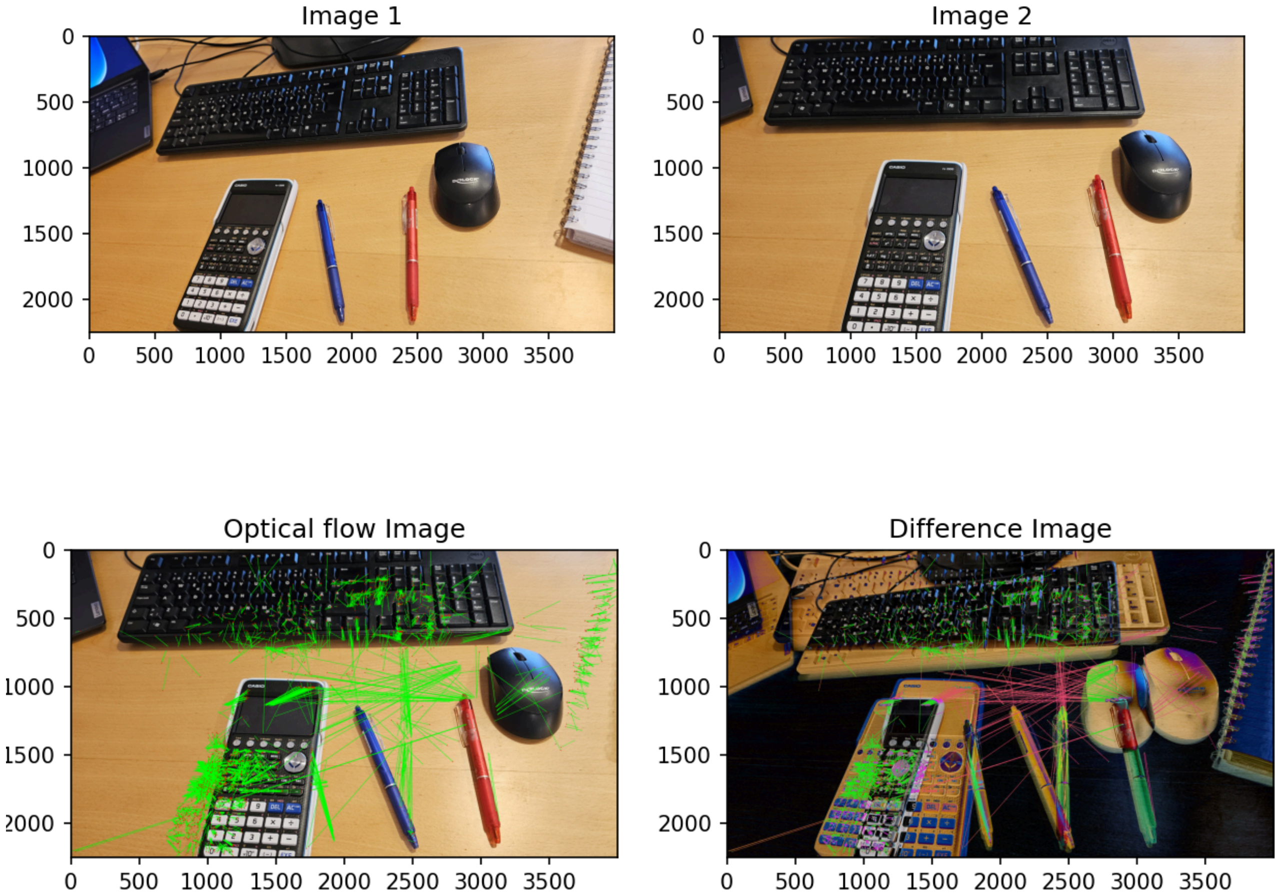
$$= \begin{bmatrix} (0.0341 \cdot 32) + (-0.0056 \cdot 39) \\ (-0.0056 \cdot 32) + (0.0112 \cdot 39) \end{bmatrix} =$$

$$= \begin{bmatrix} 1.0912 - 0.2184 \\ -0.1792 + 0.4368 \end{bmatrix} = \begin{bmatrix} 0.872 \\ 0.2576 \end{bmatrix}$$

$$X = \begin{bmatrix} 0.872 \\ 0.2576 \end{bmatrix}$$

$$(u, v) = (0.872, 0.2576)$$

5.3 Optical flow computation [7] (inspiration)



References

[1] https://en.wikipedia.org/wiki/Kalman_filter
[2] https://pmc.ncbi.nlm.nih.gov/articles/PMC7826670/
[3] https://www.geeksforgeeks.org/ml-mean-shift-clustering/
[4] https://en.wikipedia.org/wiki/Template_matching
[5] https://en.wikipedia.org/wiki/Optical_flow
[6]https://www.researchgate.net/publication/368286434_Feature-Based_Object_Detection_and_Tracking_A_Systematic_Literature_Review
[7] https://github.com/Utkal97/Object-Tracking