Tracking

Objectives

To learn and appreciate the following concepts:

✓Introduction to Optical flow

✓Lucas Kanade Method

✓KLT Tracking Method

✓Mean Shift Method & Dense Motion Estimation

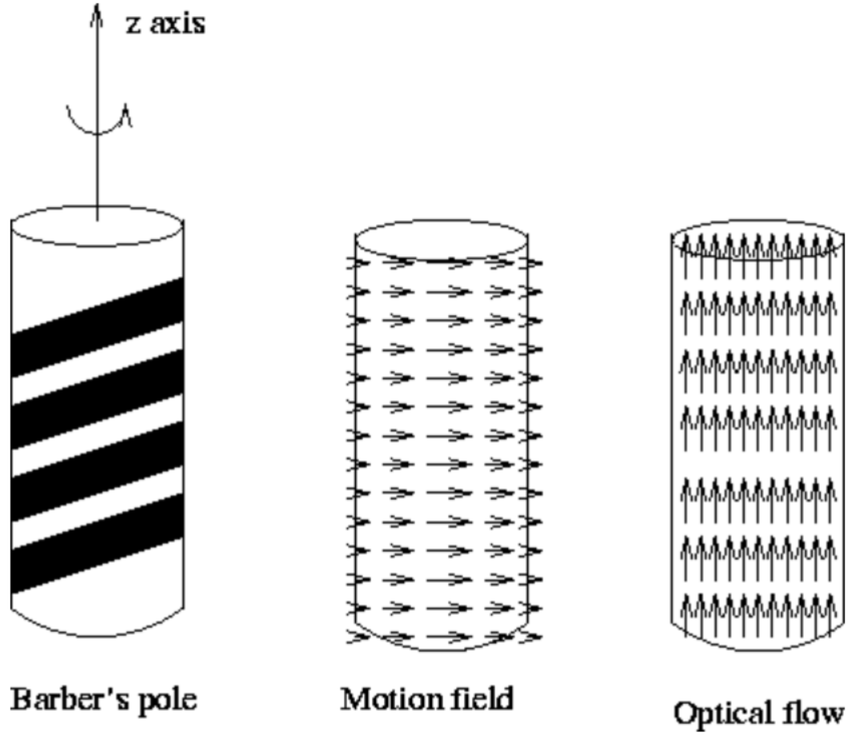
Session outcome

- At the end of session the student will be able to understand:
  - What is Optical Flow
  - Fundamentals of few methods like: Lucas Kanade Method, KLT Tracking Method, Mean Shift Method, Dense Motion Estimation

Optical Flow

- *Optical flow* is the apparent motion of brightness patterns in the image.
- Generally, optical flow corresponds to the motion field.
- For example, the motion field and optical flow of a rotating barber's pole are different:

Figure: The motion field and optical flow of a barber's pole.



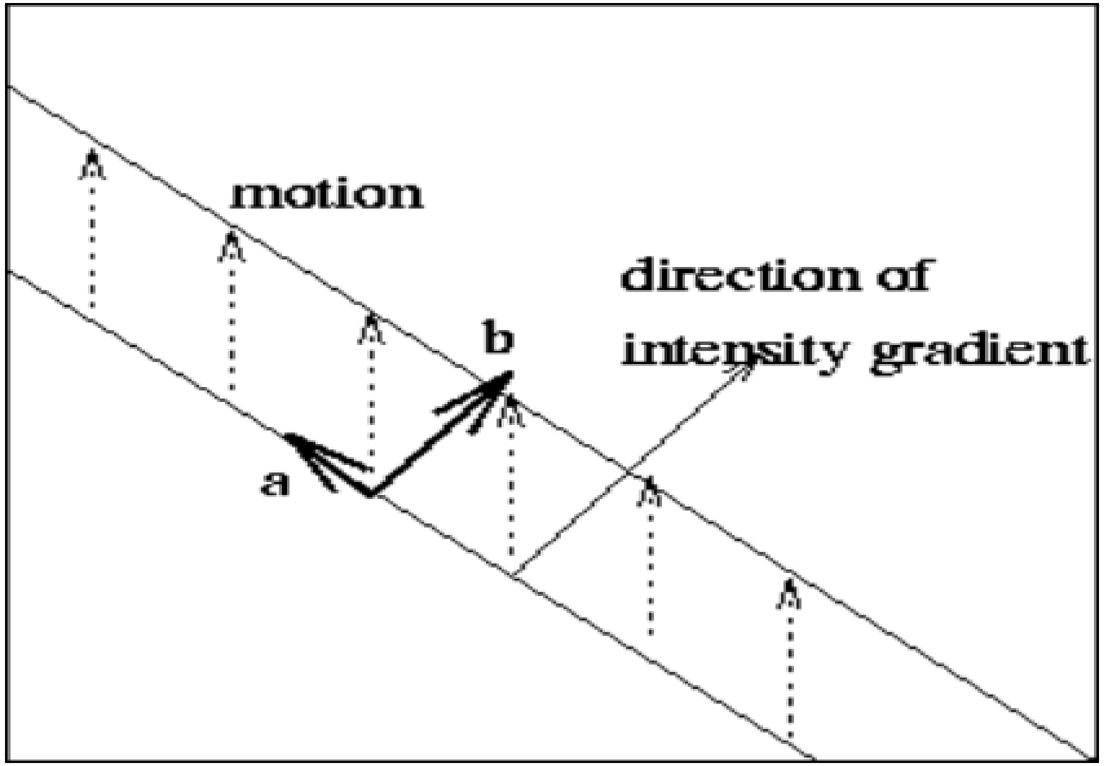
One main problem is, we are unable to measure the component of optical flow that is in the direction of intensity gradient.

And also we are unable to measure the component tangential to the intensity gradient.

Optical Flow

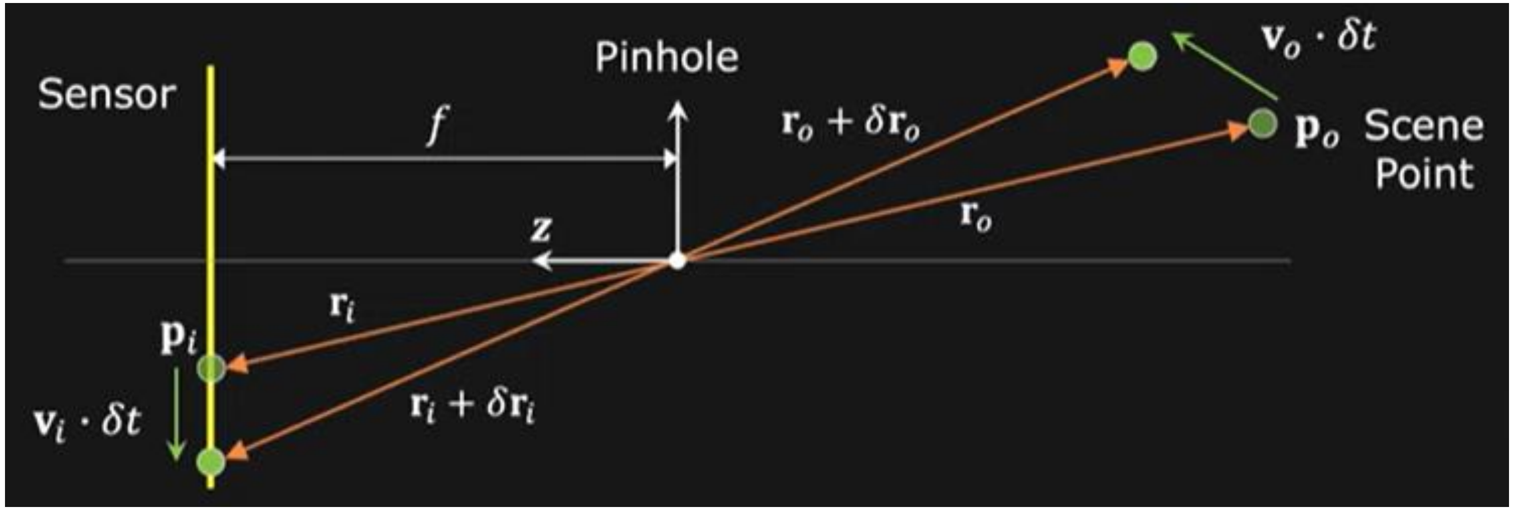
- Method to estimate apparent motion of scene points from a sequence of images.

Figure: The aperture problem. We can only measure the component b.



Motion Field and Optical Flow

- **Image velocity of a point** that is moving in the scene.



f -> focal length
r -> radius
v -> velocity
z -> distance between camera and calibration object.

Image Point Velocity: $\mathbf{v}_i = \dfrac{d\mathbf{r}_i}{dt}$
(Motion Field)

Scene Point Velocity: $\mathbf{v}_o = \dfrac{d\mathbf{r}_o}{dt}$

Perspective projection: $\dfrac{\mathbf{r}_i}{f} = \dfrac{\mathbf{r}_o}{\mathbf{r}_o \cdot \mathbf{z}}$

Image Point Velocity: $\mathbf{v}_i = \dfrac{d\mathbf{r}_i}{dt} = f\,\dfrac{(\mathbf{r}_o \cdot \mathbf{z})\mathbf{v}_0 - (\mathbf{v}_o \cdot \mathbf{z})\mathbf{r}_0}{(\mathbf{r}_o \cdot \mathbf{z})^2}$
(Motion Field)

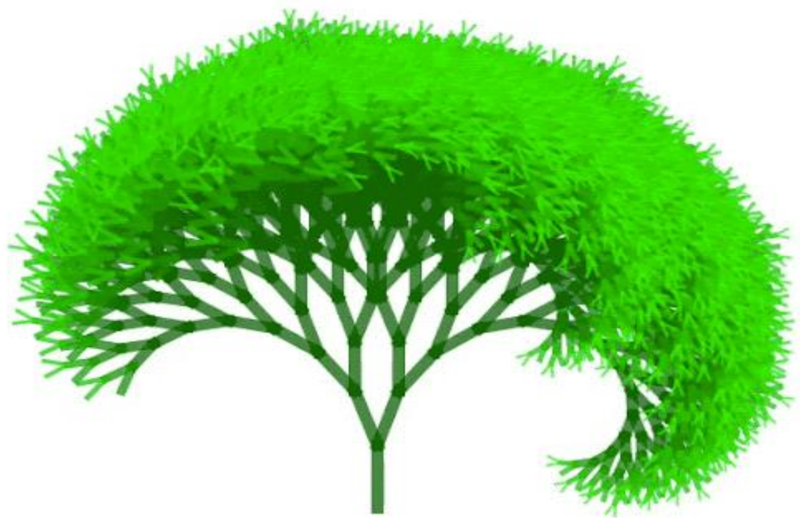
Motion Field and Optical Flow

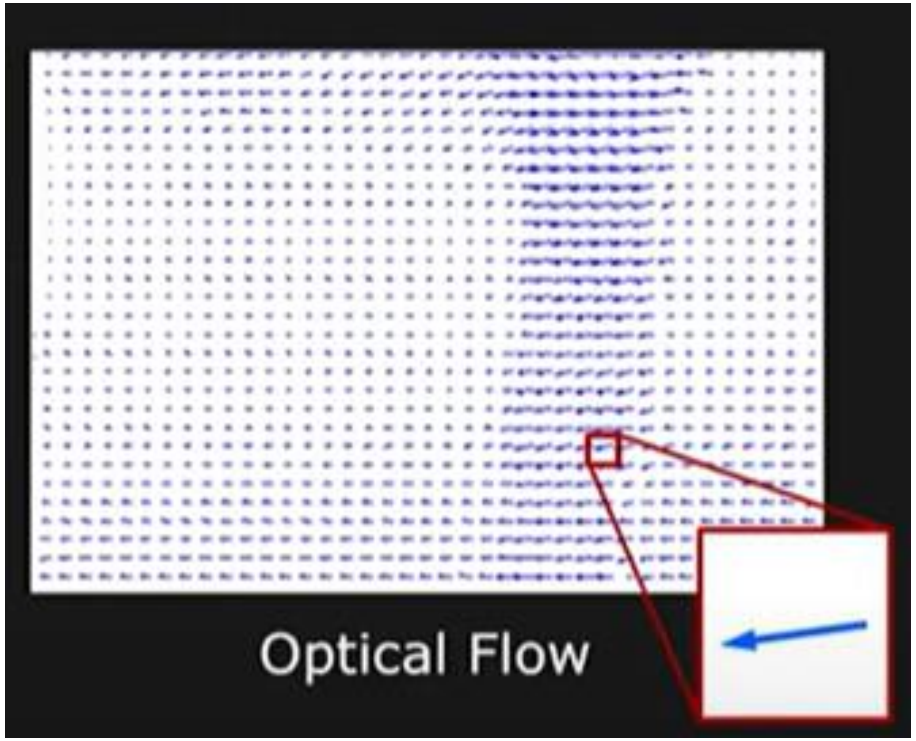
- **Image velocity of a point:**

$$\mathbf{v}_i = f \frac{(\mathbf{r}_o \times \mathbf{v}_0) \times \mathbf{z}}{(\mathbf{r}_o \cdot \mathbf{z})^2}$$

Motion Field and Optical Flow

- Motion of brightness patterns in the image:

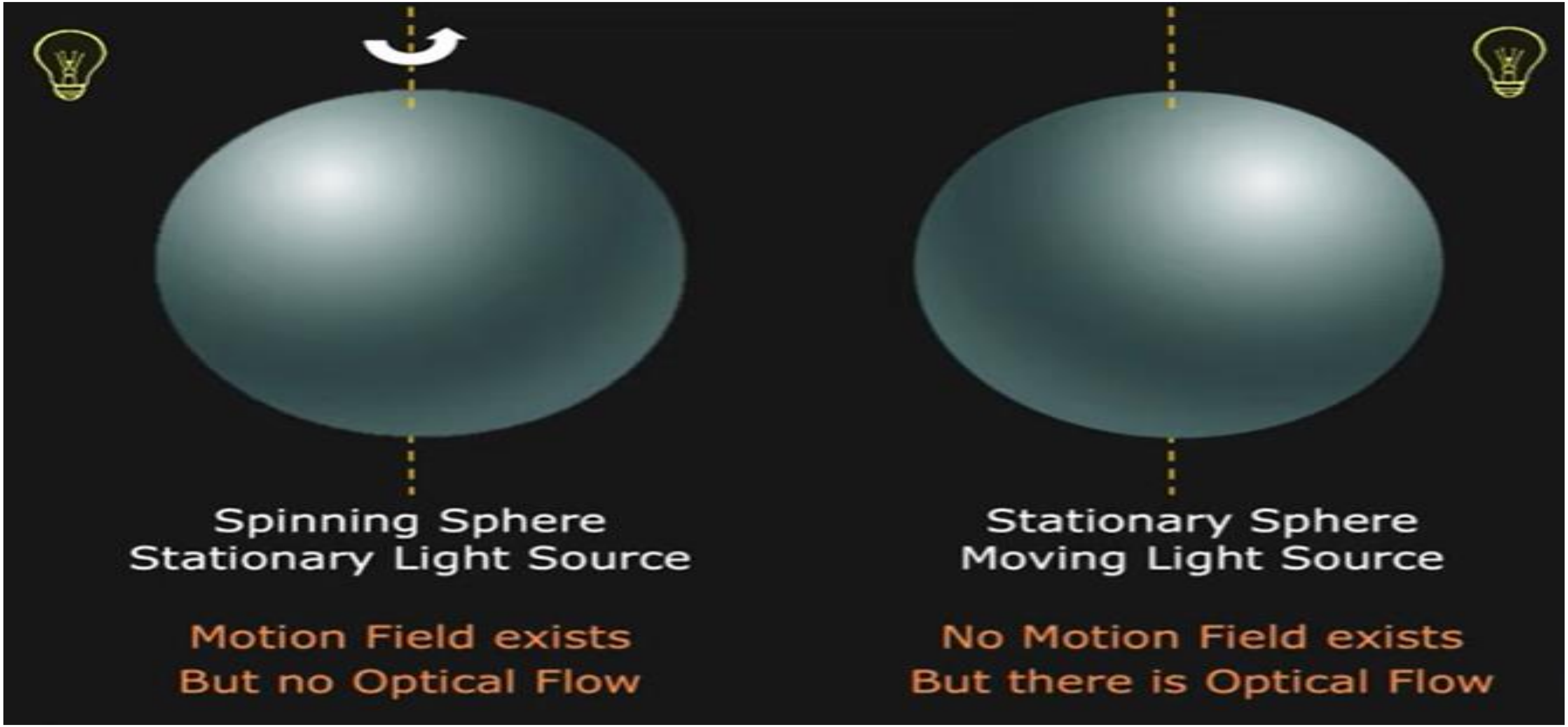




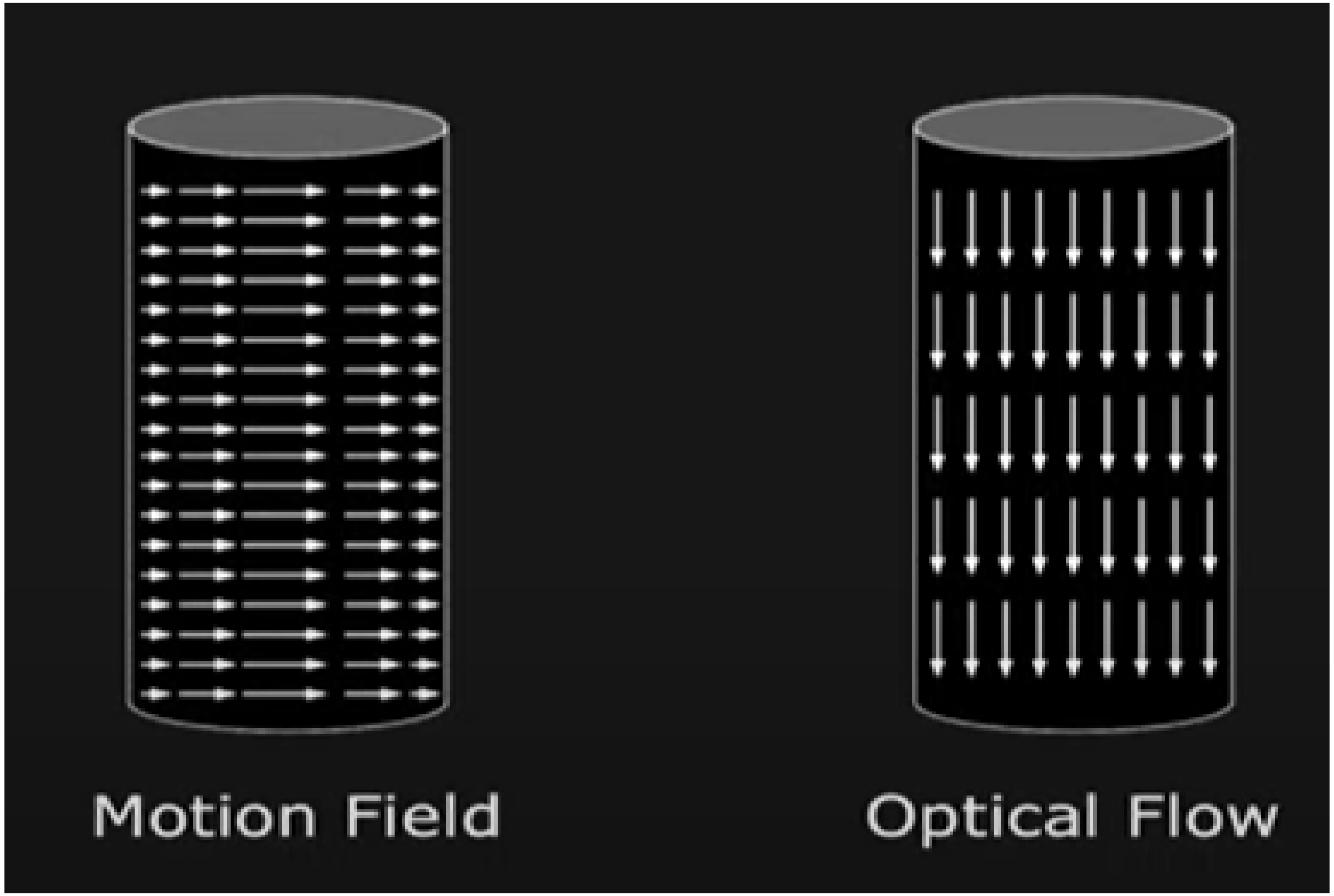
- Image Sequence (2 frames)

Velocity of different patterns

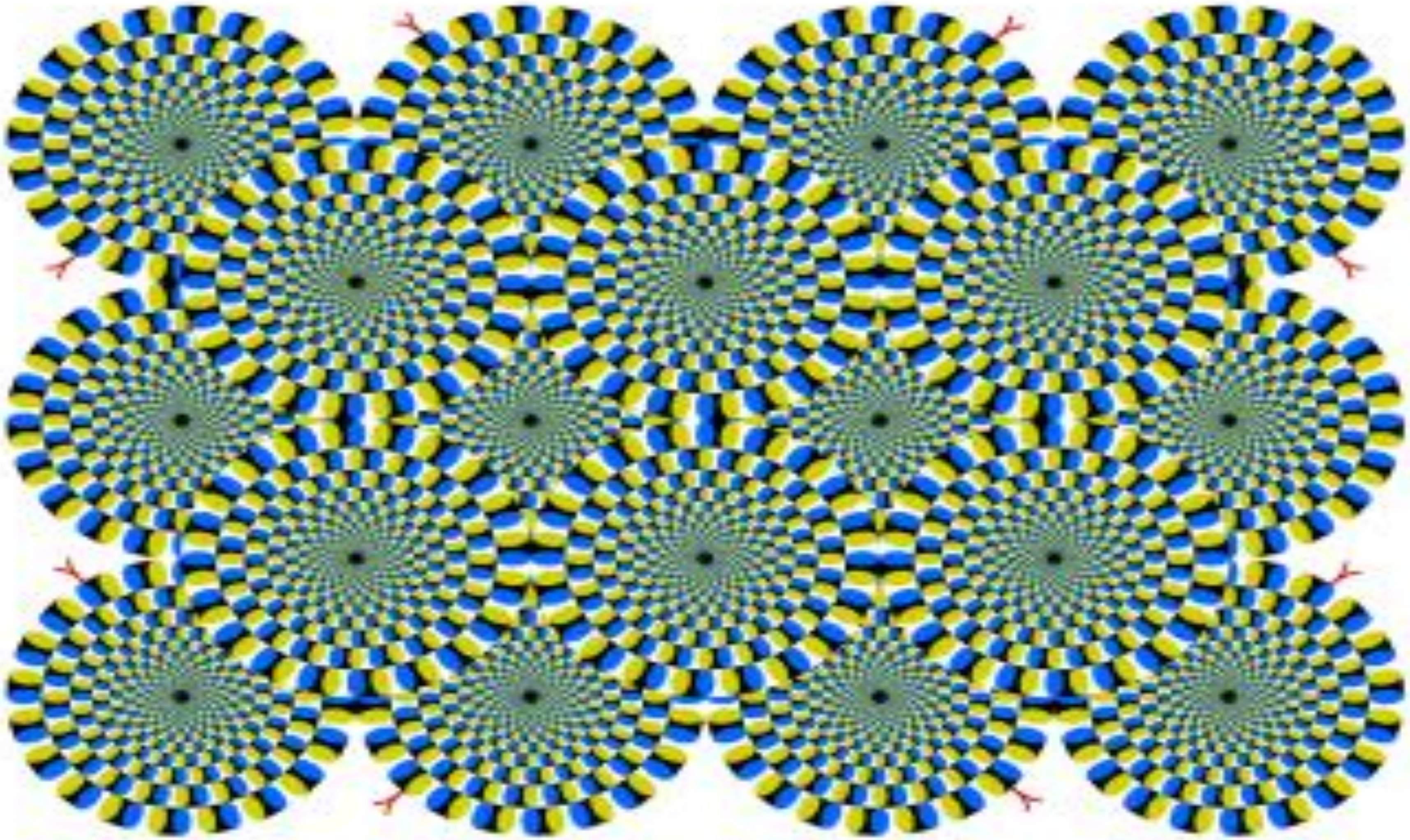
When Optical Flow is not equal to motion field?



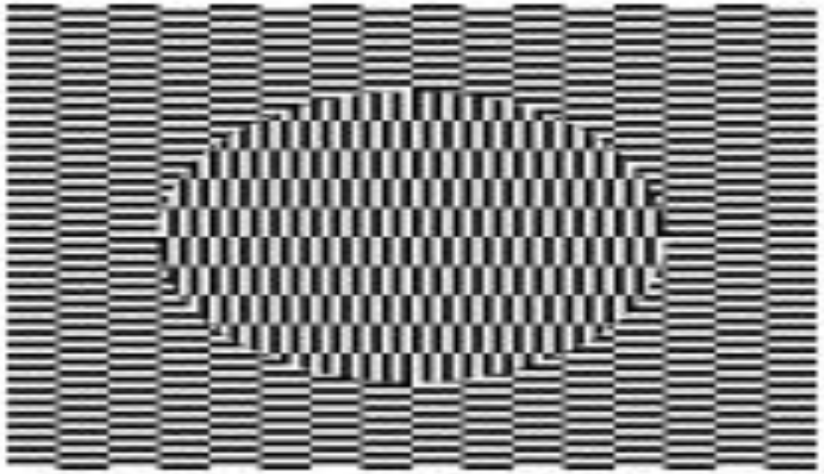
When Optical Flow is not equal to motion field?

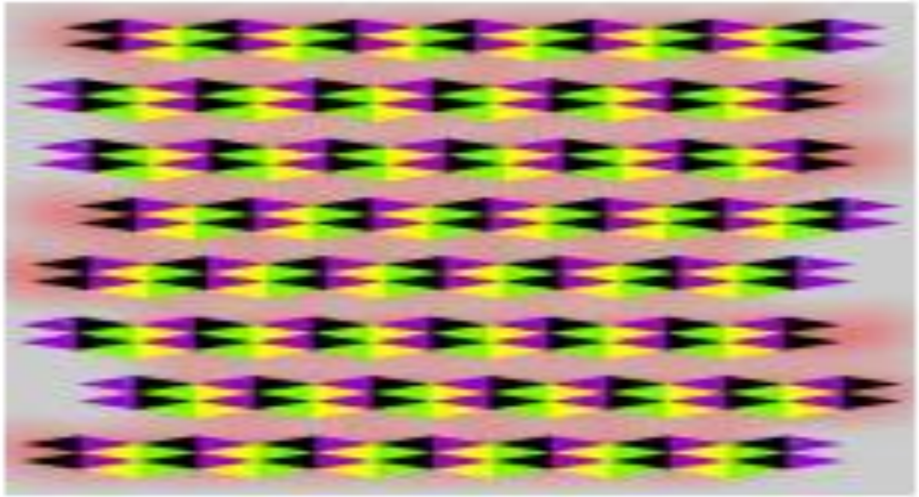


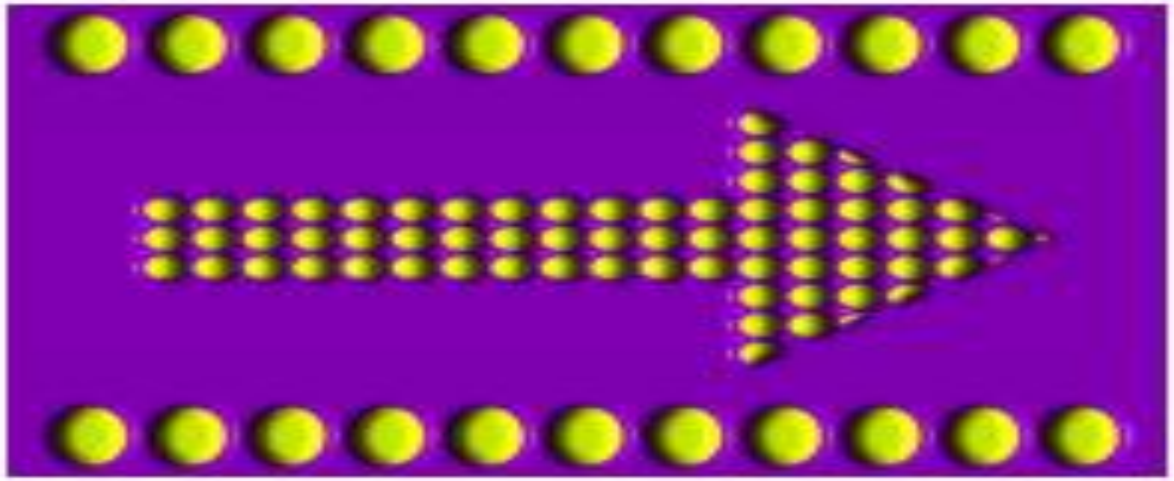
Seeing motion from a static picture?

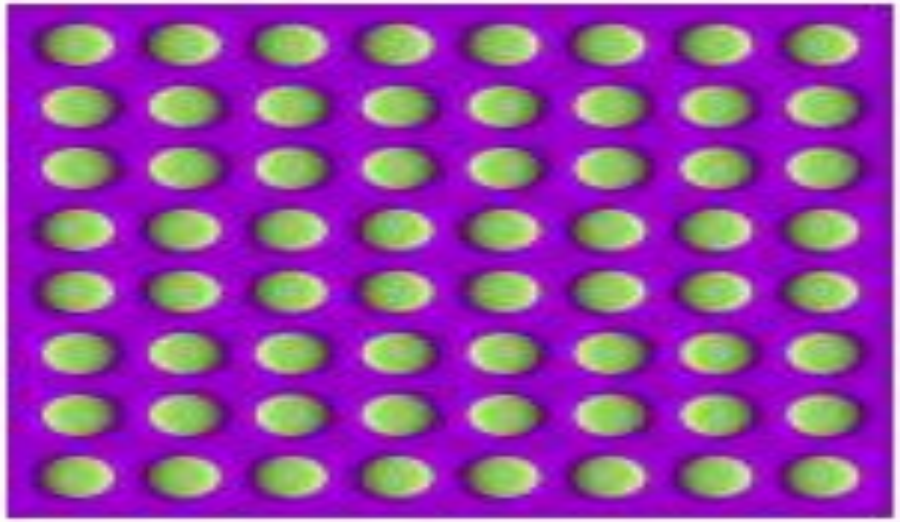


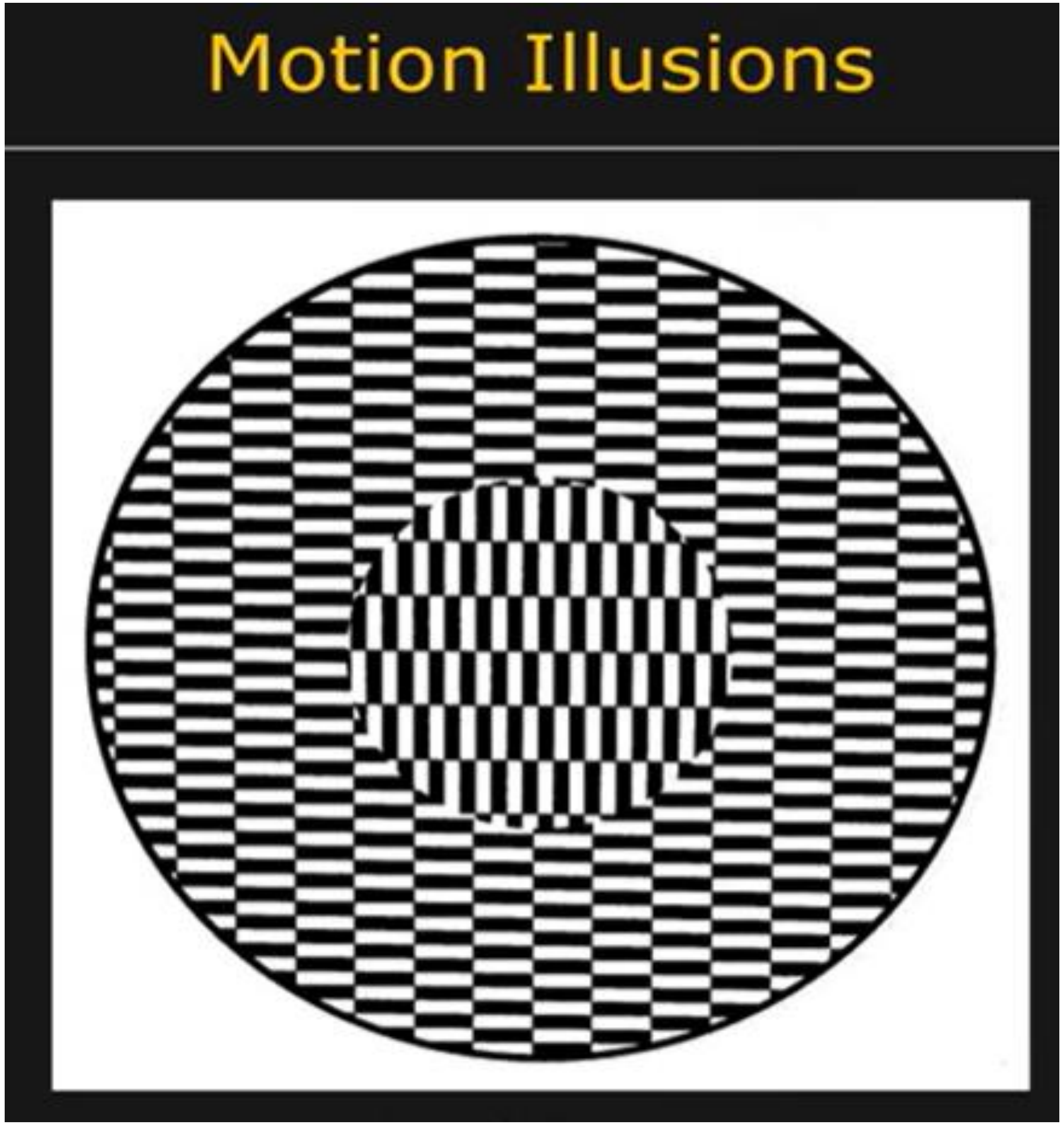
More examples

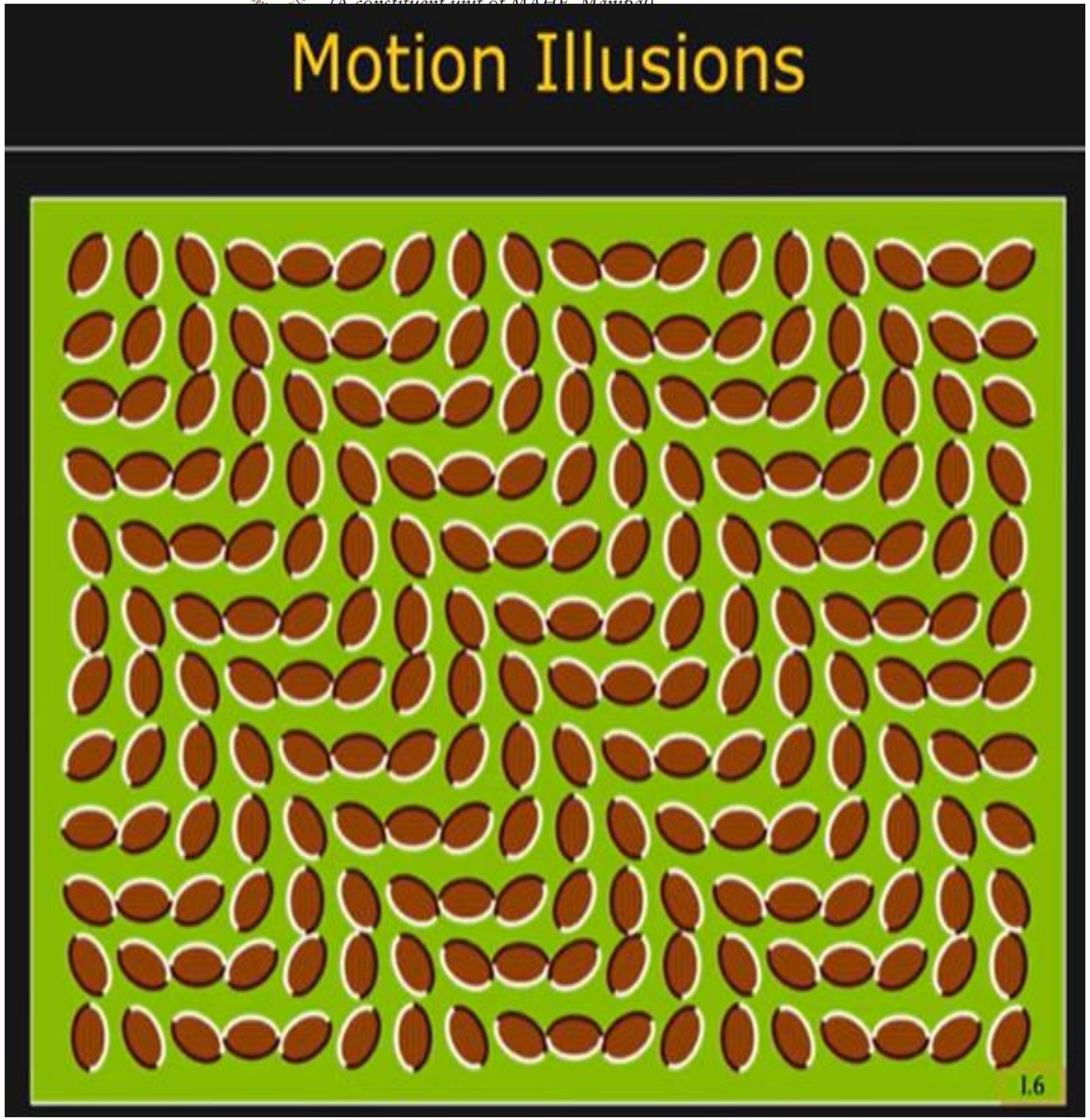






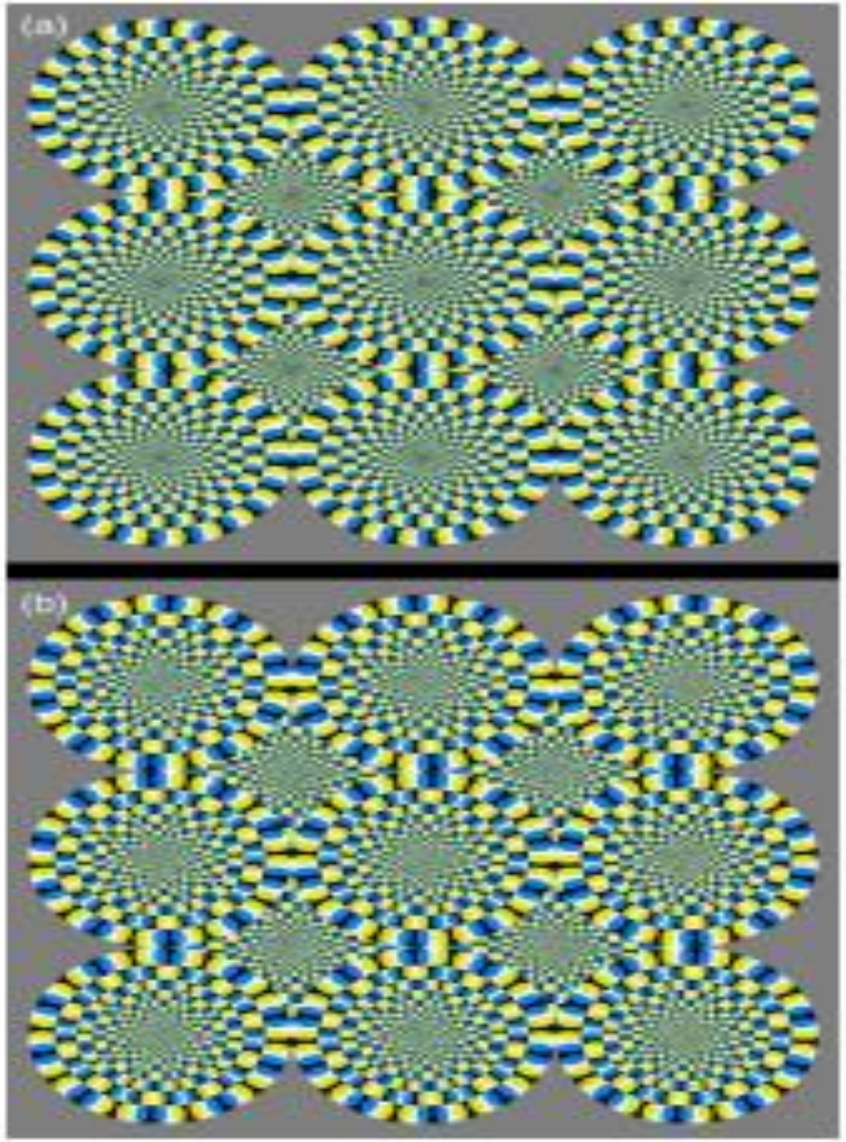



Motion Illusions


Motion Illusions
1.6

How is this possible?

- The true mechanism is to be revealed
- FMRI data suggest that illusion is related to some component of eye movements
- We don't expect computer vision to "see" motion from these stimuli, yet



The cause of motion

- Three factors in imaging process
  - Light
  - Object
  - Camera

- Varying either of them causes motion
  - Static camera, moving objects (surveillance)
  - Moving camera, static scene (3D capture)
  - Moving camera, moving scene (sports, movie)
  - Static camera, moving objects, moving light (time lapse)

Optical Flow

(Problem definition)

$I(x, y, t)$ $I(x, y, t')$

Estimate the motion (flow) between these two consecutive images

How is this different from estimating a 2D transform?

Key Assumptions

(unique to optical flow)

Color Constancy

(Brightness constancy for intensity images)

Implication: allows for pixel to pixel comparison (not image features)

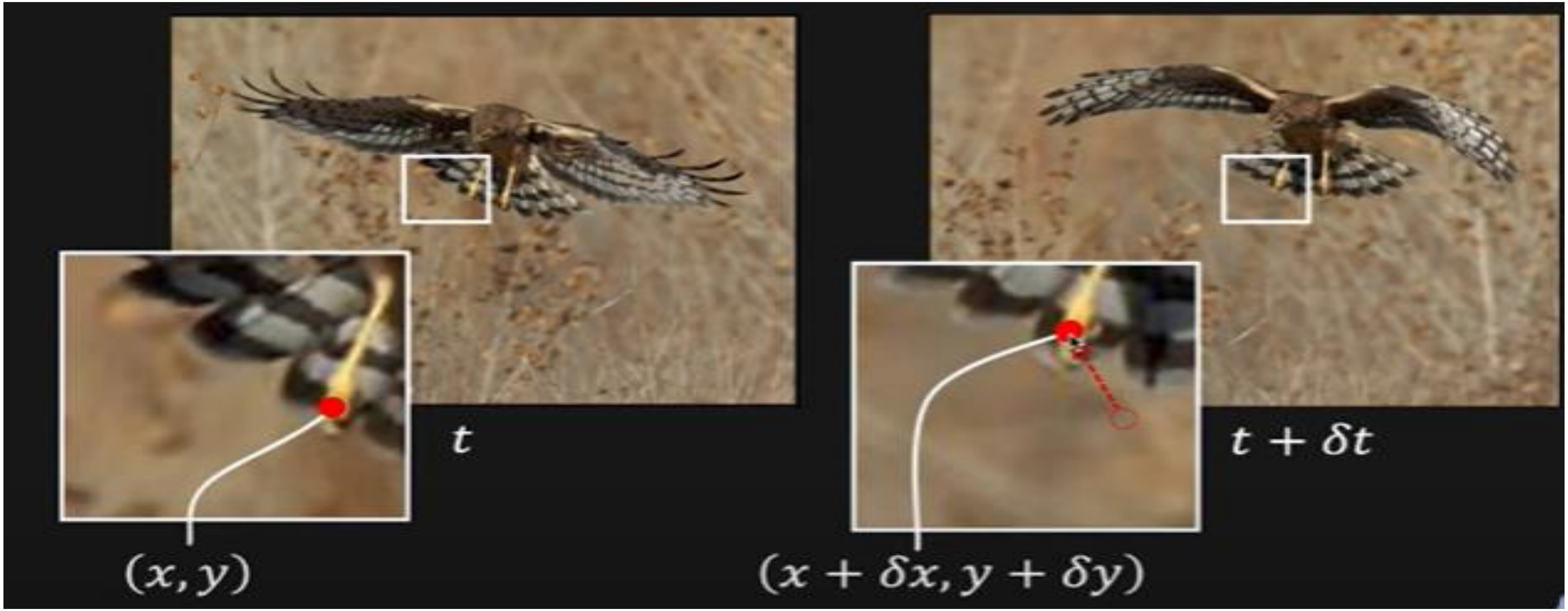
Small Motion

(pixels only move a little bit)

Implication: linearization of the brightness constancy constraint

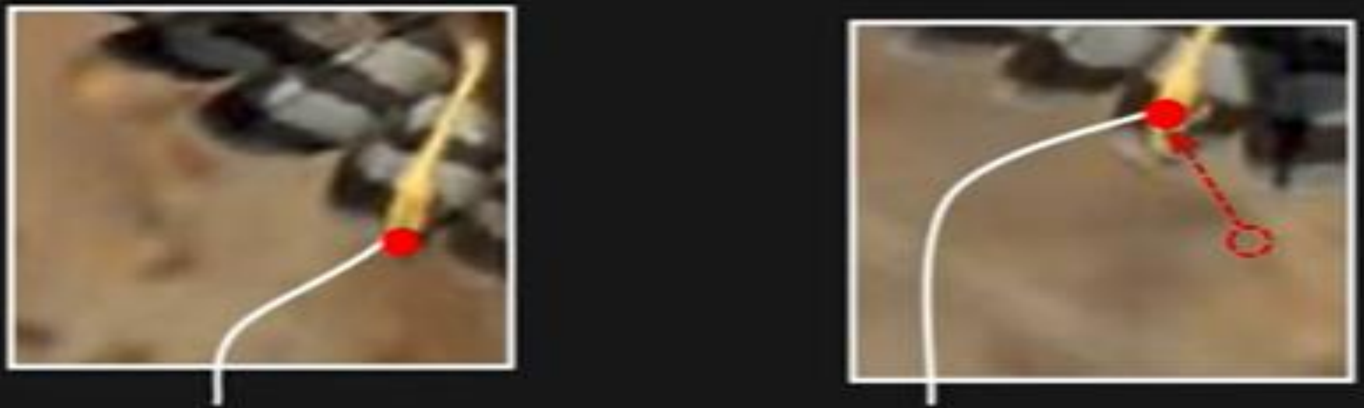
Optical Flow Constraint Equation

Optical Flow Constraint Equation



Displacement: $(\delta x, \delta y)$ Optical Flow: $(u, v) = \left(\frac{\delta x}{\delta t}, \frac{\delta y}{\delta t}\right)$

Optical Flow Constraint Equation



$I(x, y, t)$ $\qquad$ $I(x + \delta x, y + \delta y, t + \delta t)$

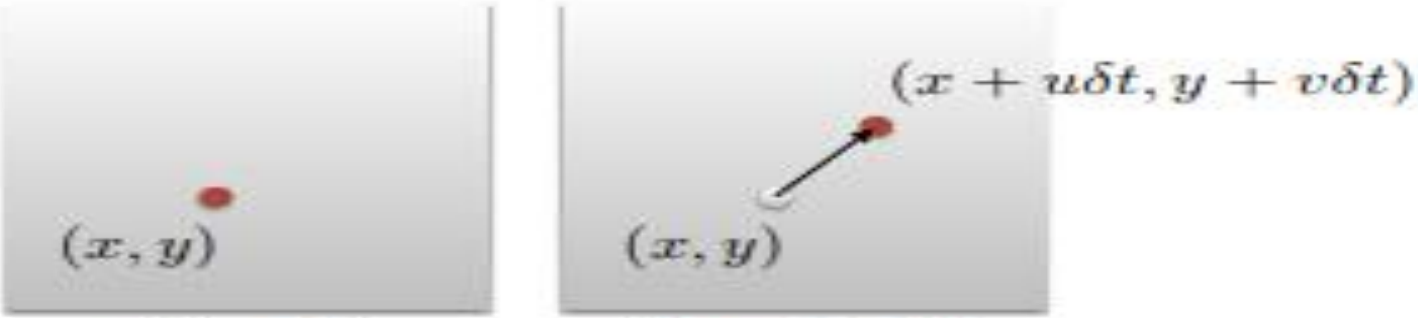
Assumption #1:

Brightness of image point remains constant over time

$$\boxed{I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t)}$$

Optical Flow Constraint Equation

Brightness constancy



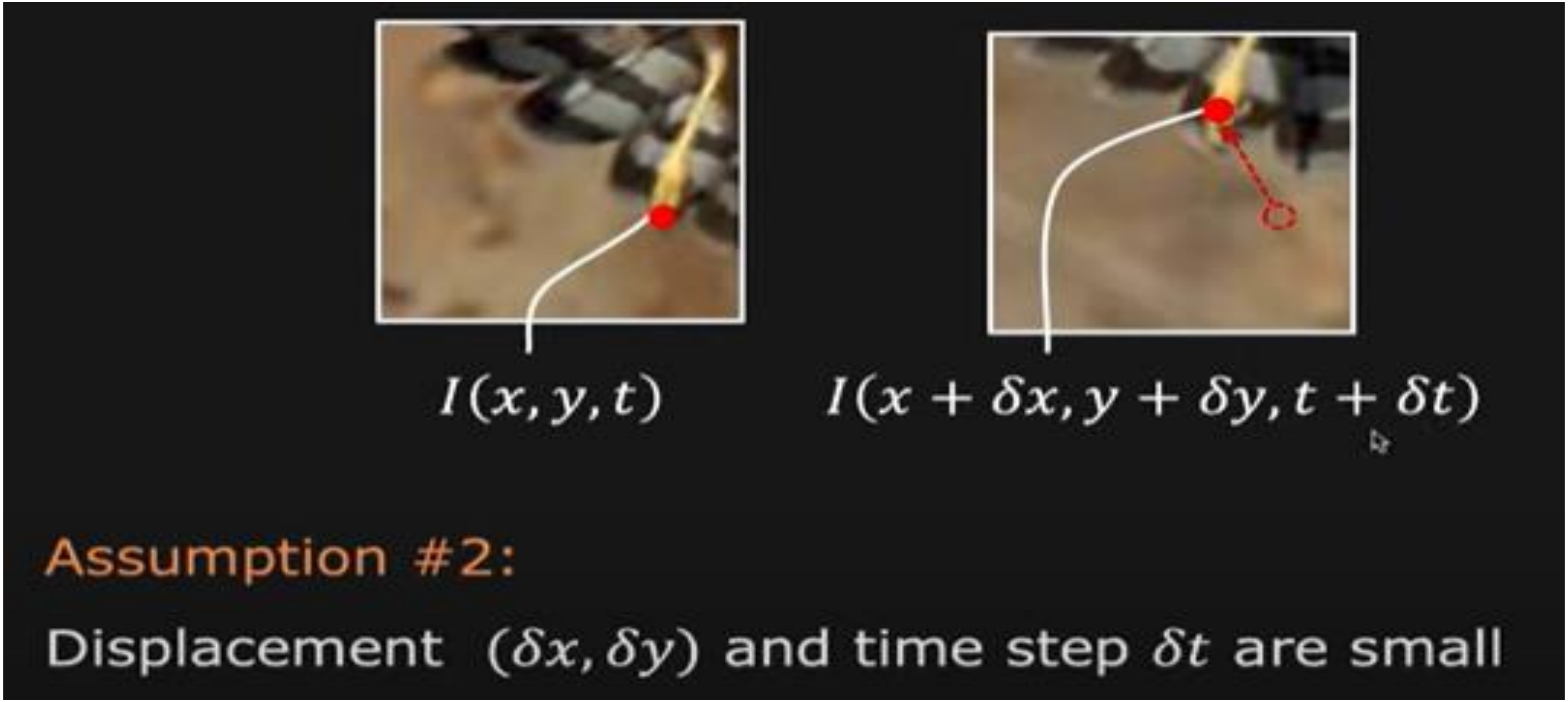
$I(x, y, t)$ $\quad$ $I(x, y, t + \delta t)$

Optical flow (velocities): (u, v) Displacement: $(\delta x, \delta y) = (u\delta t, v\delta t)$

$$I(x + u\delta t, y + v\delta t, t + \delta t) = I(x, y, t)$$

For a really small time step...

Optical Flow Constraint Equation



Optical Flow Constraint Equation

Optical Flow Constraint Equation

$I(x, y, t)$

$I(x + \delta x, y + \delta y, t + \delta t)$

Assumption #2:

Displacement $(\delta x, \delta y)$ and time step δt are small

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t$$

$$\boxed{I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + I_x \delta x + I_y \delta y + I_t \delta t}$$

Optical Flow Constraint Equation

$$I(x+\delta x, y+\delta y, t+\delta t) = I(x,y,t) \quad \text{.............} (1)$$

$$I(x+\delta x, y+\delta y, t+\delta t) = I(x,y,t) + I_x\delta x + I_y\delta y + I_t\delta t \quad \text{.............} (2)$$

Subtract (1) from (2): $I_x\delta x + I_y\delta y + I_t\delta t = 0$

Divide by δt and take limit as $\delta t \to 0$: $I_x\frac{\partial x}{\partial t} + I_y\frac{\partial y}{\partial t} + I_t = 0$

Constraint Equation: $\boxed{I_x u + I_y v + I_t = 0}$ (u, v): Optical Flow

(I_x, I_y, I_t) can be easily computed from two frames

Optical Flow Constraint Equation

$$I(x + u\delta t, y + v\delta t, t + \delta t) = I(x, y, t)$$

$$\cancel{I(x, y, t)} + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = \cancel{I(x, y, t)}$$

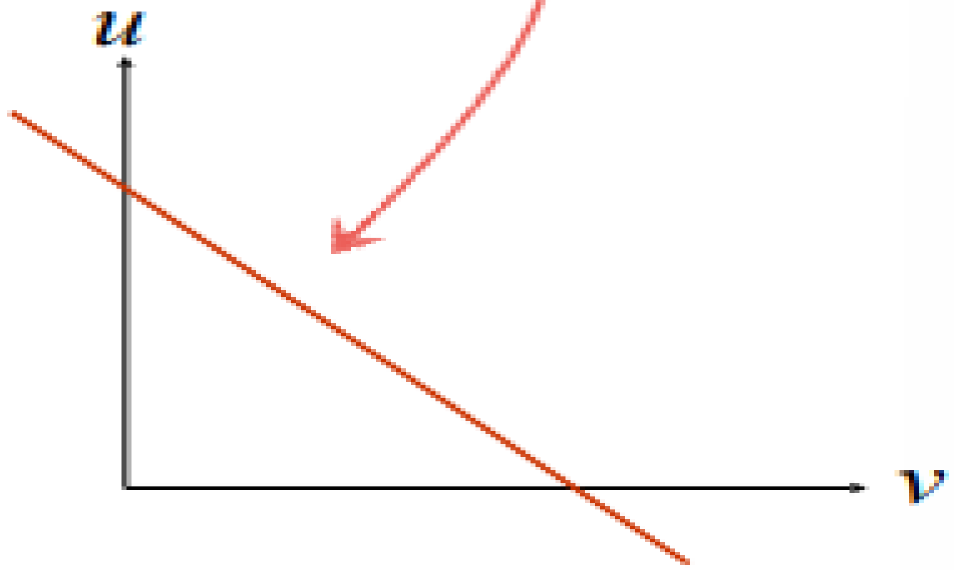
$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0$$

$$\boxed{I_x u + I_y v + I_t = 0}$$

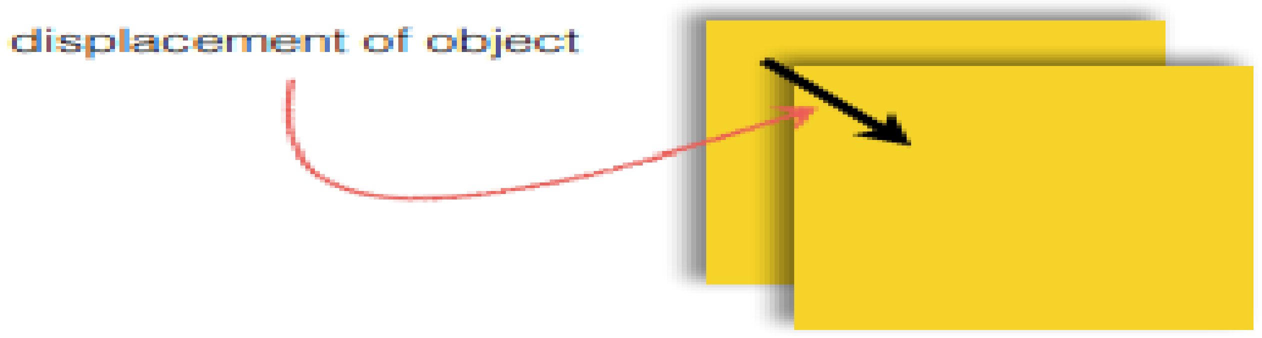
Optical Flow Constraint Equation

Solution lies on a straight line

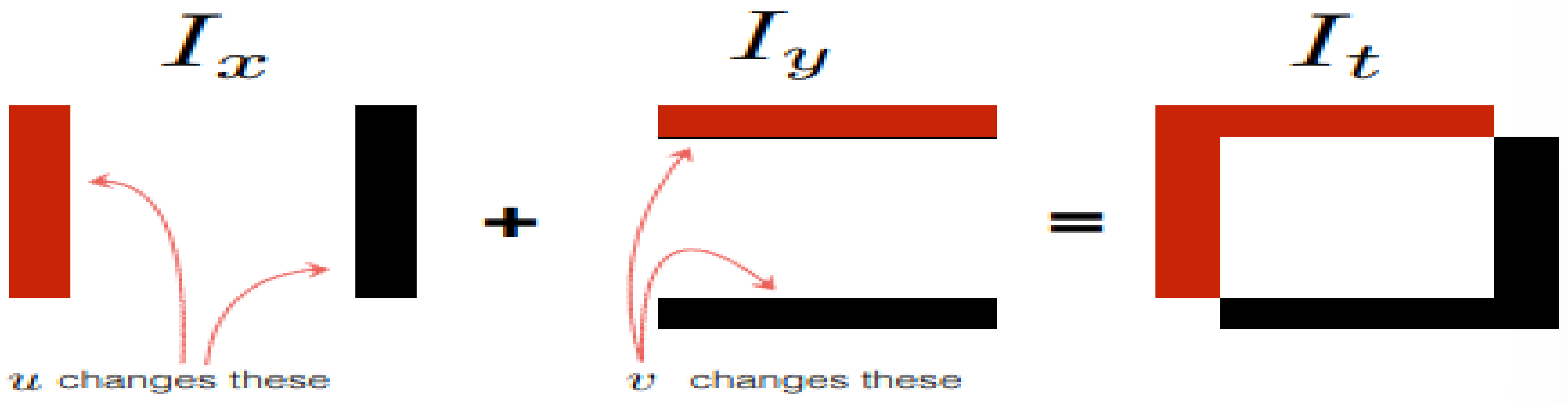
$$\boxed{I_x u + I_y v + I_t = 0}$$



The solution cannot be determined uniquely with a single constraint (a single pixel)



Find the optical flow such that it satisfies:



Optical Flow is Under Constrained

Constraint Equation: $\boxed{I_x u + I_y v + I_t = 0}$

2 unknowns, 1 equation.

Where can we get an additional constraint?

Lucas Kanade Method

$$I_x u + I_y v + I_t = 0$$

$$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y}$$

spatial derivative

$$\boxed{u = \frac{dx}{dt} \quad v = \frac{dy}{dt}}$$

optical flow

$$I_t = \frac{\partial I}{\partial t}$$

temporal derivative

How can we use the brightness constancy equation to estimate the optical flow?

Lucas Kanade Method

unknown

$$I_x u + I_y v + I_t = 0$$

known

We need at least _____ equations to solve for 2 unknowns.

Lucas Kanade Method

unknown

$$I_x u + I_y v + I_t = 0$$

known

Where do we get more equations (constraints)?

Lucas Kanade Method

- Motion Tracking is recognizing a salient feature and monitoring its motion across multiple frames.
- Applications like facial detection, video surveillance, traffic estimation, etc.
- Object is monitored for spatial and temporal changes through a sequence of frames

Lucas Kanade Method - Challenges

- Rapid movement of object across frames
- Changing object orientation
- Changing Illumination
- Complex feature tracking like facial expressions
- Interfering background

Methodology for Object Tracking

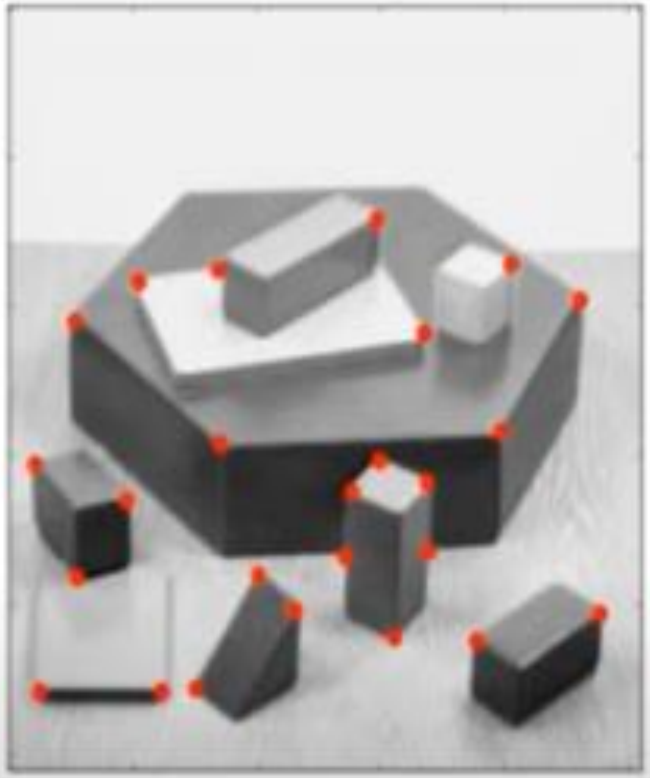
Step 1 : Use a suitable feature detection algorithm to detect salient features in an image, that will then be tracked. In this project, Harris Corner Detector

Step 2: Optical Flow computation using Lucas-Kanade algorithm

Harris Corner Detector – Recall the concept

- Corner is an intersection of two edges
- Harris Corner gives a mathematically representation for this concept.

$$\hat{M}(x,y) = \sum_{x,y} w(x,y) \otimes \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \frac{\partial I}{\partial x}\cdot\frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial x}\cdot\frac{\partial I}{\partial y} & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix}$$



Harris Corner Detector – Recall the concept

- Cornerness value greater than a threshold is a Corner.

$$R = \det M - k(trace\, M)^2$$

$$\det M = \lambda_1 \lambda_2$$

$$trace\, M = \lambda_1 + \lambda_2$$

Algorithm

Step 1 : Find the image x and y derivatives, Ix and Iy.

$$I_x = G_\sigma^x * I \quad I_y = G_\sigma^y * I$$

Step 2 : Using Ix and Iy, find Ix2, Iy2 and Ixy.

$$I_{x2} = I_x . I_x \quad I_{y2} = I_y . I_y \quad I_{xy} = I_x . I_y$$

Step 3 : For every pixel, find the sum of product of derivatives.

$$S_{x2} = G_{\sigma'} * I_{x2} \quad S_{y2} = G_{\sigma'} * I_{y2} \quad S_{xy} = G_{\sigma'} * I_{xy}$$

Step 4 : Find the Cornerness using the Harris Corner Detector Equation.

$$\hat{M}(x,y) = \sum_{x,y} w(x,y) \otimes \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \frac{\partial I}{\partial x} \cdot \frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial x} \cdot \frac{\partial I}{\partial y} & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix}$$

$$R = \det M - k(\operatorname{trace} M)^2$$

$$\det M = \lambda_1 \lambda_2$$

$$\operatorname{trace} M = \lambda_1 + \lambda_2$$

Lucas-Kanade for Optical Flow

- Motion of features across frames due to the relative motion between the scene and the camera.
- Computing Optical Flow of an image can help in Motion Tracking

Lucas Kanade Method for Optical Flow

Where do we get more equations (constraints)?

$$I_x u + I_y v + I_t = 0$$

Assume that the surrounding patch (say 5x5) has **constant flow**

Lucas Kanade Method for Optical Flow - Algorithm

Step 1 : Compute the Image x and Image y derivatives.

Step 2 : Compute the difference Image It = Image 1 – Image 2.

Step 3 : Smoothen the image components Ix, Iy and It.

Step 4 : Solve the Linear Equations for each pixel and calculate the Eigen values.

Step 5 : Depending on Eigen values obtained, solve the equations using Cramer's rule.

Step 6 : Plot the optical Flow vectors.

Lucas Kanade Method for Optical Flow

Assumptions:

Flow is locally smooth

Neighboring pixels have same displacement

Using a 5 x 5 image patch, gives us equations

Lucas Kanade Method for Optical Flow

Assumptions:

Flow is locally smooth

Neighboring pixels have same displacement

Using a 5 x 5 image patch, gives us 25 equations

$$\begin{aligned}
I_x(\boldsymbol{p}_1)u + I_y(\boldsymbol{p}_1)v &= -I_t(\boldsymbol{p}_1) \\
I_x(\boldsymbol{p}_2)u + I_y(\boldsymbol{p}_2)v &= -I_t(\boldsymbol{p}_2) \\
&\vdots \\
I_x(\boldsymbol{p}_{25})u + I_y(\boldsymbol{p}_{25})v &= -I_t(\boldsymbol{p}_{25})
\end{aligned}$$

Lucas Kanade Method

Assumptions:

Flow is locally smooth

Neighboring pixels have same displacement

Using a 5 x 5 image patch, gives us 25 equations

$$\begin{bmatrix} I_x(\boldsymbol{p}_1) & I_y(\boldsymbol{p}_1) \\ I_x(\boldsymbol{p}_2) & I_y(\boldsymbol{p}_2) \\ \vdots & \vdots \\ I_x(\boldsymbol{p}_{25}) & I_y(\boldsymbol{p}_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\boldsymbol{p}_1) \\ I_t(\boldsymbol{p}_2) \\ \vdots \\ I_t(\boldsymbol{p}_{25}) \end{bmatrix}$$

Matrix form

Lucas Kanade Method for Optical Flow

Assumptions:

Flow is locally smooth

Neighboring pixels have same displacement

Using a 5 x 5 image patch, gives us 25 equations

$$\underbrace{\begin{bmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2) & I_y(\mathbf{p}_2) \\ \vdots & \vdots \\ I_x(\mathbf{p}_{25}) & I_y(\mathbf{p}_{25}) \end{bmatrix}}_{\substack{A \\ 25 \times 2}} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{\substack{x \\ 2 \times 1}} = - \underbrace{\begin{bmatrix} I_t(\mathbf{p}_1) \\ I_t(\mathbf{p}_2) \\ \vdots \\ I_t(\mathbf{p}_{25}) \end{bmatrix}}_{\substack{b \\ 25 \times 1}}$$

How many equations? How many unknowns? How do we solve this?

Lucas Kanade Method for Optical Flow

- Consider an Image I(x,y). For smaller motion, the new image can be represented as:

  H(x,y) = I(x+u,y+v)

- Where (u,v) represents the displacement of the pixel.
- Solving this equation using Taylors Expansion we obtain the Lucas-Kanade equation :

$$\underbrace{\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}}_{A^T A} \begin{bmatrix} u \\ v \end{bmatrix} = - \underbrace{\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}}_{A^T b}$$

Lucas Kanade Method for Optical Flow

Least squares approximation

$\hat{x} = \arg\min_x ||Ax - b||^2$ is equivalent to solving $A^\top A\hat{x} = A^\top b$

To obtain the least squares solution solve:

$$\underbrace{\begin{bmatrix} \sum_{p\in P} I_x I_x & \sum_{p\in P} I_x I_y \\ \sum_{p\in P} I_y I_x & \sum_{p\in P} I_y I_y \end{bmatrix}}_{A^\top A} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{\hat{x}} = - \underbrace{\begin{bmatrix} \sum_{p\in P} I_x I_t \\ \sum_{p\in P} I_y I_t \end{bmatrix}}_{A^\top b}$$

where the summation is over each pixel $\boldsymbol{p}$ in patch $\boldsymbol{P}$

Sometimes called '**Lucas-Kanade Optical Flow**'
(special case of the LK method with a translational warp model)

Lucas Kanade Method for Optical Flow

When is this solvable?

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

$A^\top A$ should be invertible

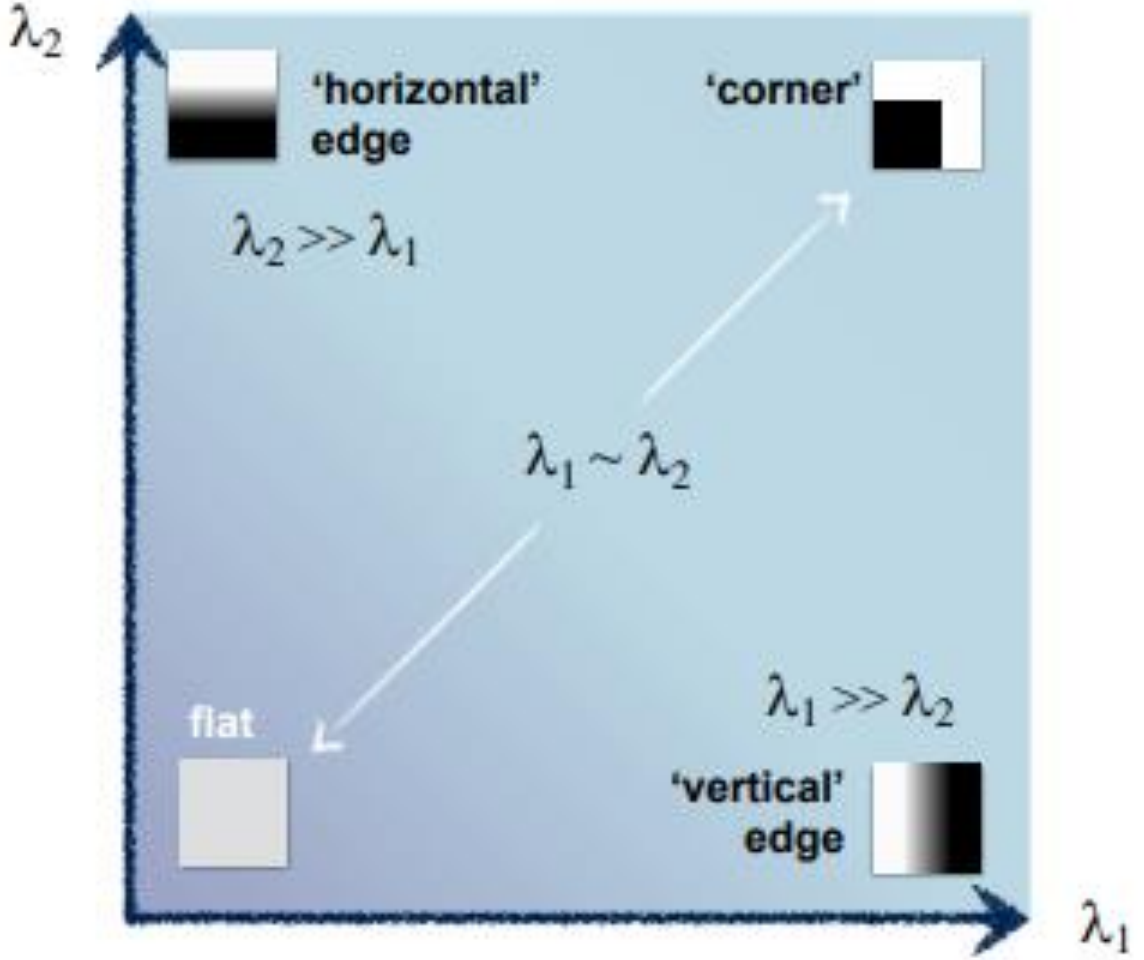
$A^\top A$ should not be too small

λ_1 and λ_2 should not be too small

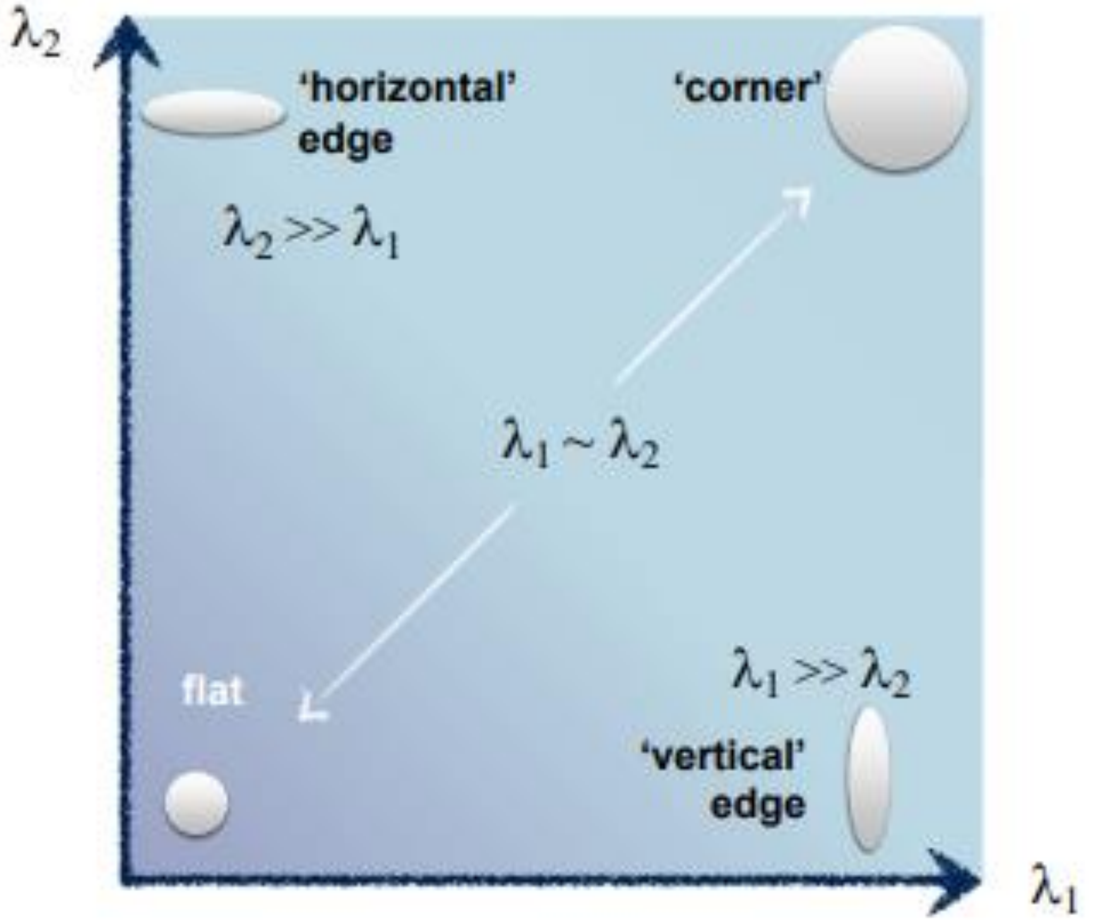
$A^\top A$ should be well conditioned

λ_1/λ_2 should not be too large (λ_1=larger eigenvalue)

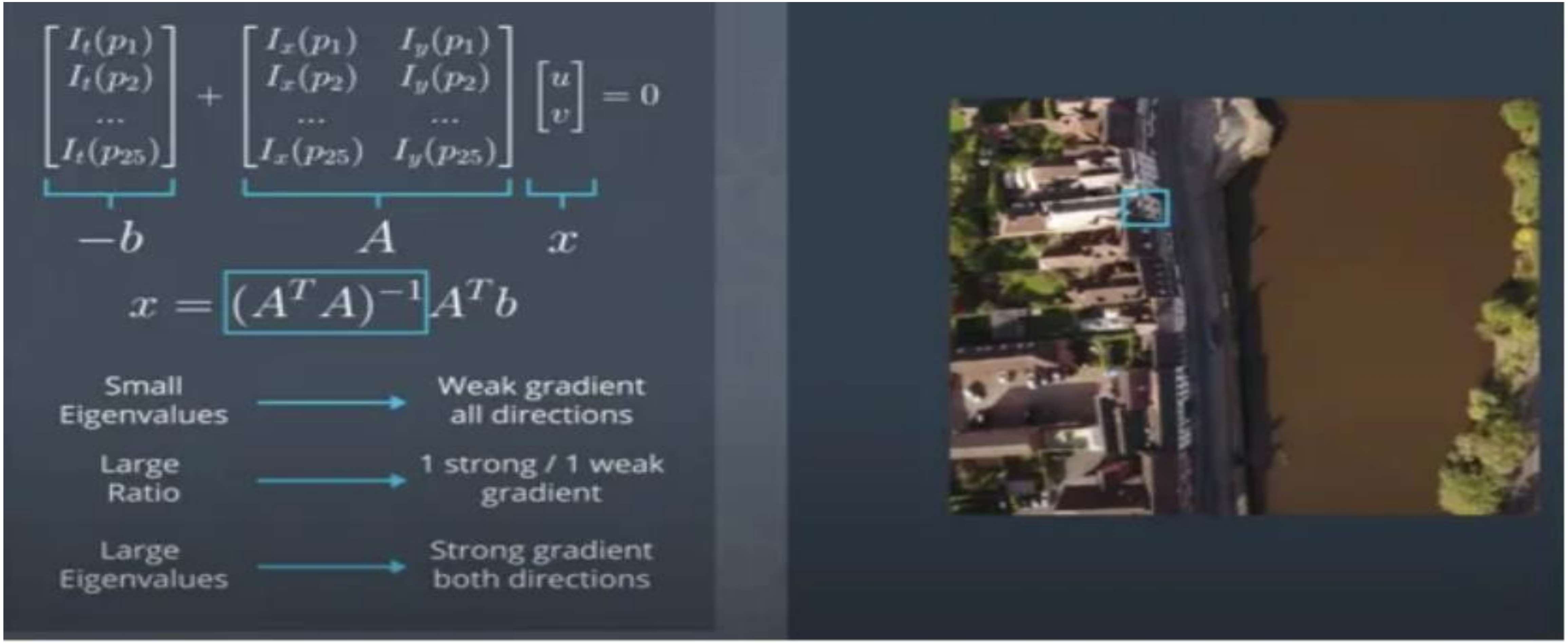
interpreting eigenvalues



interpreting eigenvalues



Lucas Kanade Method for Optical Flow



Solving the least-squares problem with the -b + A x = 0

Lucas Kanade Method for Optical Flow

- Lucas Kanade Method is based on something known as Brightness constancy assumption. The key idea here is that pixel level brightness won't change a lot in just one frame. It assumes that the color of an object does not change significantly and significantly in the previous two frames.

$$I(x, y, t) = I(x + u, y + v, t + 1)$$

Brightness constancy constraint

$$I_t + I_x u + I_y v = 0$$

Simplified Brightness constancy constraint

Optical flow is only valid in regions where

$$A^T A = \begin{pmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_y I_x & \sum I_y^2 \end{pmatrix}$$

KLT Method

Kanade-Lucas-Tomasi

How should we track them from frame to frame?

Lucas-Kanade

Method for aligning (tracking) an image patch

How should we select features?

Tomasi-Kanade

Method for choosing the best feature (image patch) for tracking

KLT Method

KLT algorithm

1. Find corners satisfying $\min(\lambda_1, \lambda_2) > \lambda$

2. For each corner compute displacement to next frame using the Lucas-Kanade method

3. Store displacement of each corner, update corner position

4. (optional) Add more corner points every M frames using 1

5. Repeat 2 to 3 (4)

6. Returns long trajectories for each corner point

KLT Method

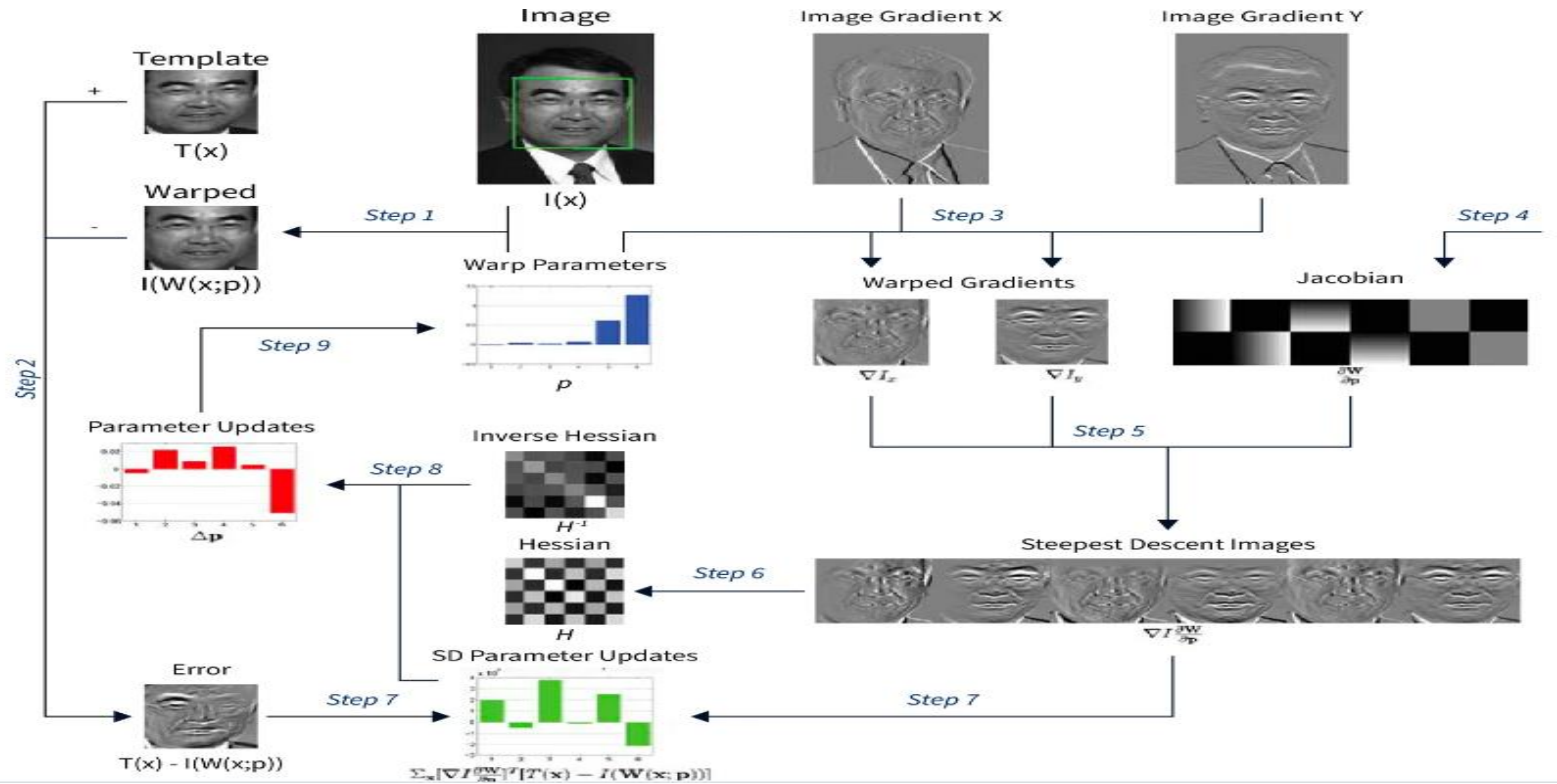
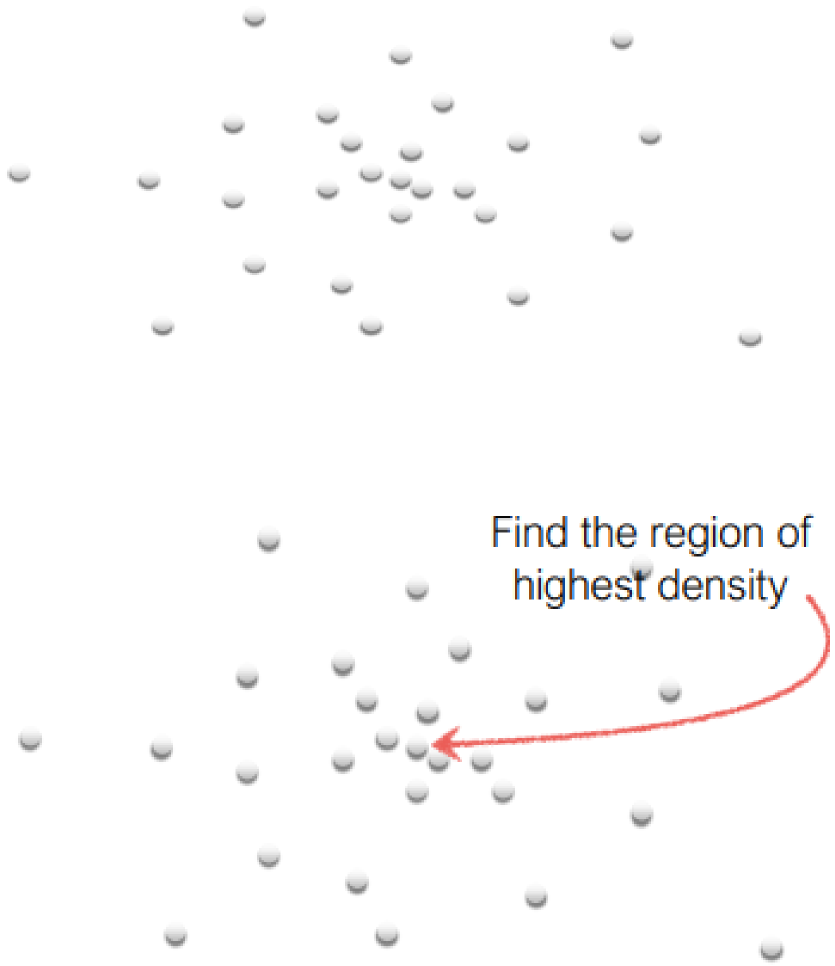


Image
Image Gradient X
Image Gradient Y
Template
T(x)
I(x)
Warped
I(W(x;p))
Step 1
Step 3
Step 4
Warp Parameters
Warped Gradients
Jacobian
Step 9
p
Step 2
Parameter Updates
Inverse Hessian
Step 5
Step 8
Δp
Hessian
Steepest Descent Images
Step 6
H
SD Parameter Updates
Error
Step 7
Step 7
T(x) - I(W(x;p))

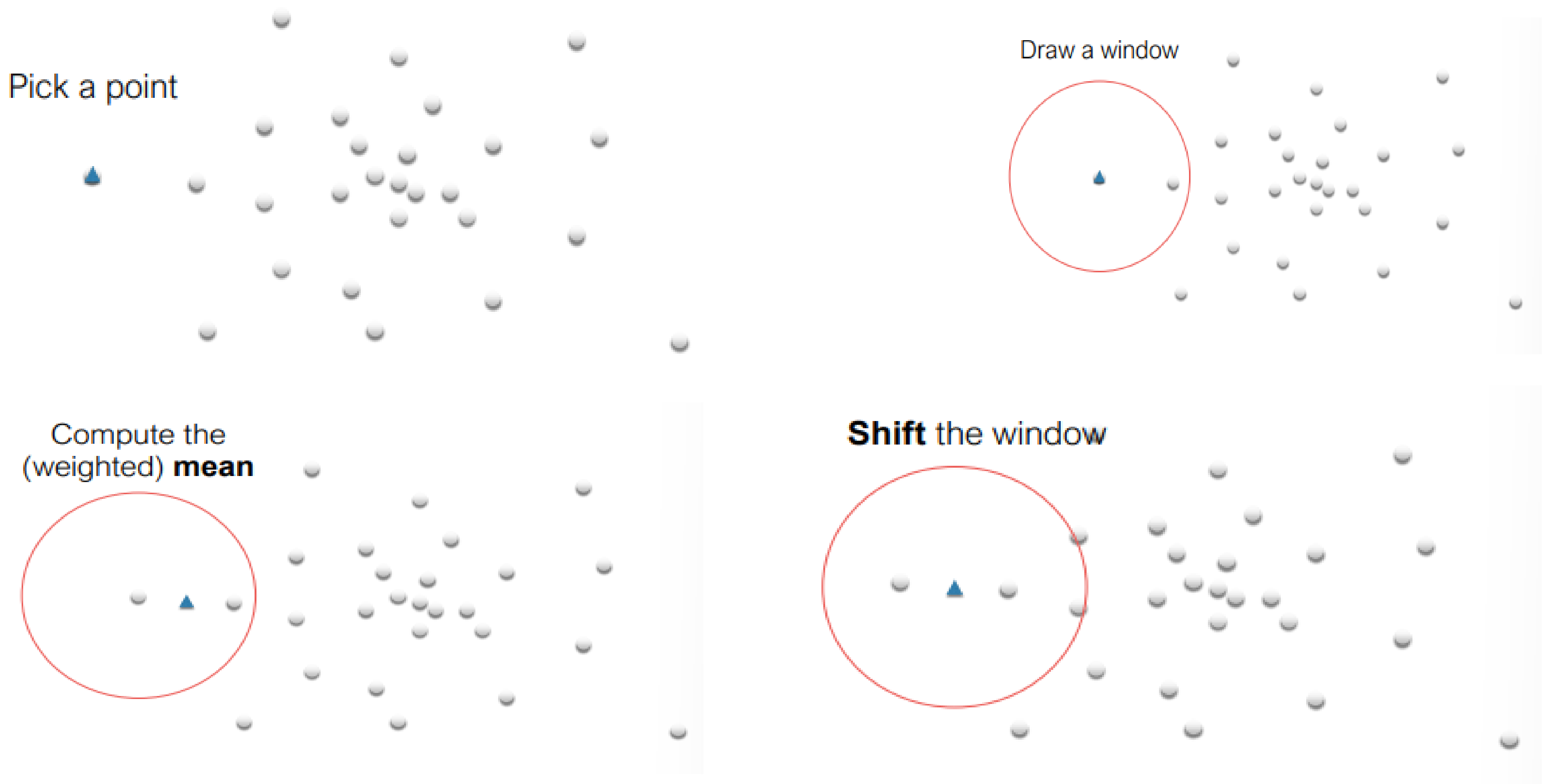
What are good features for tracking?

Intuitively, we want to avoid smooth regions and edges. But is there a more is principled way to define good features?

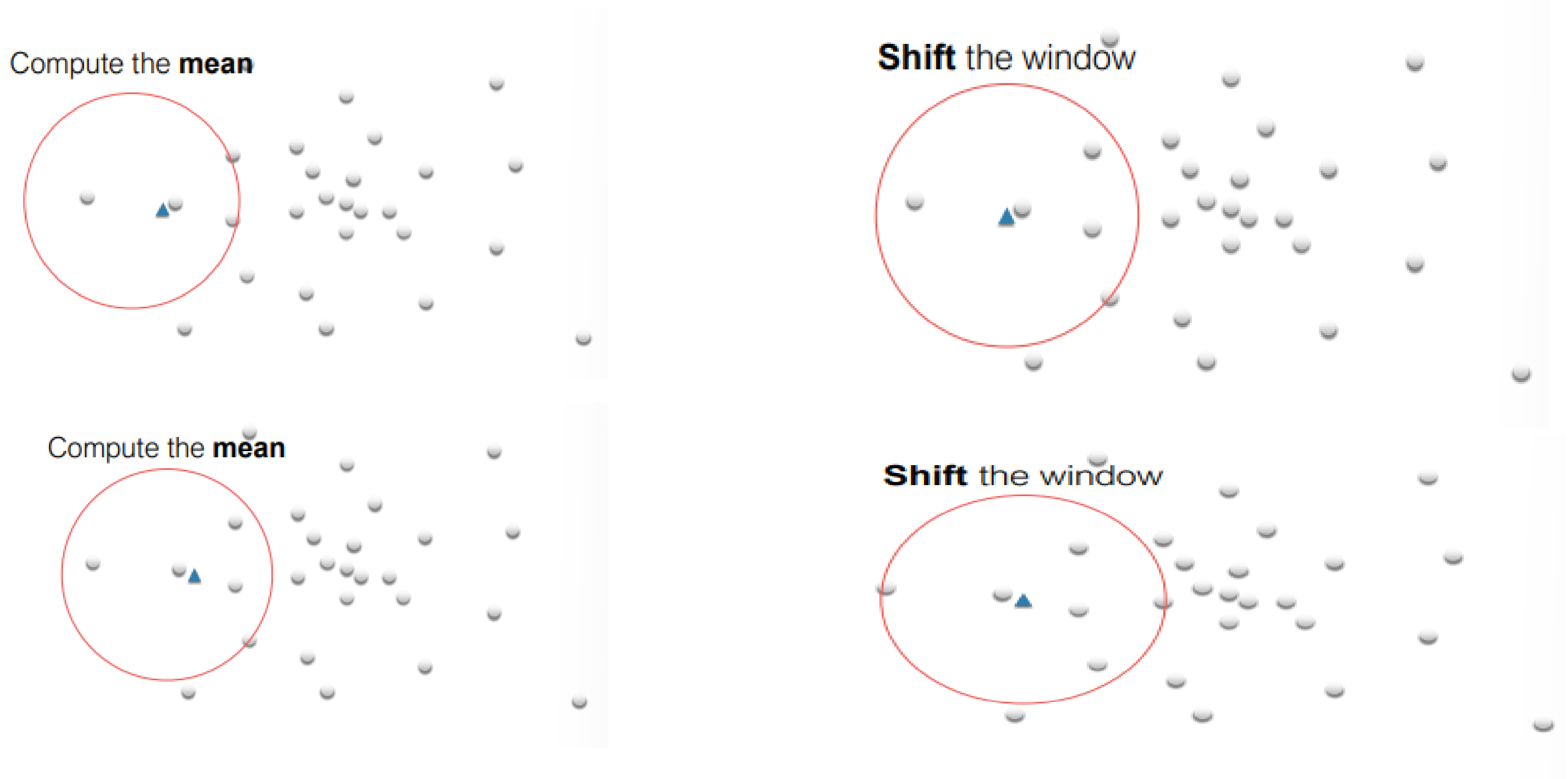
Mean Shift Method



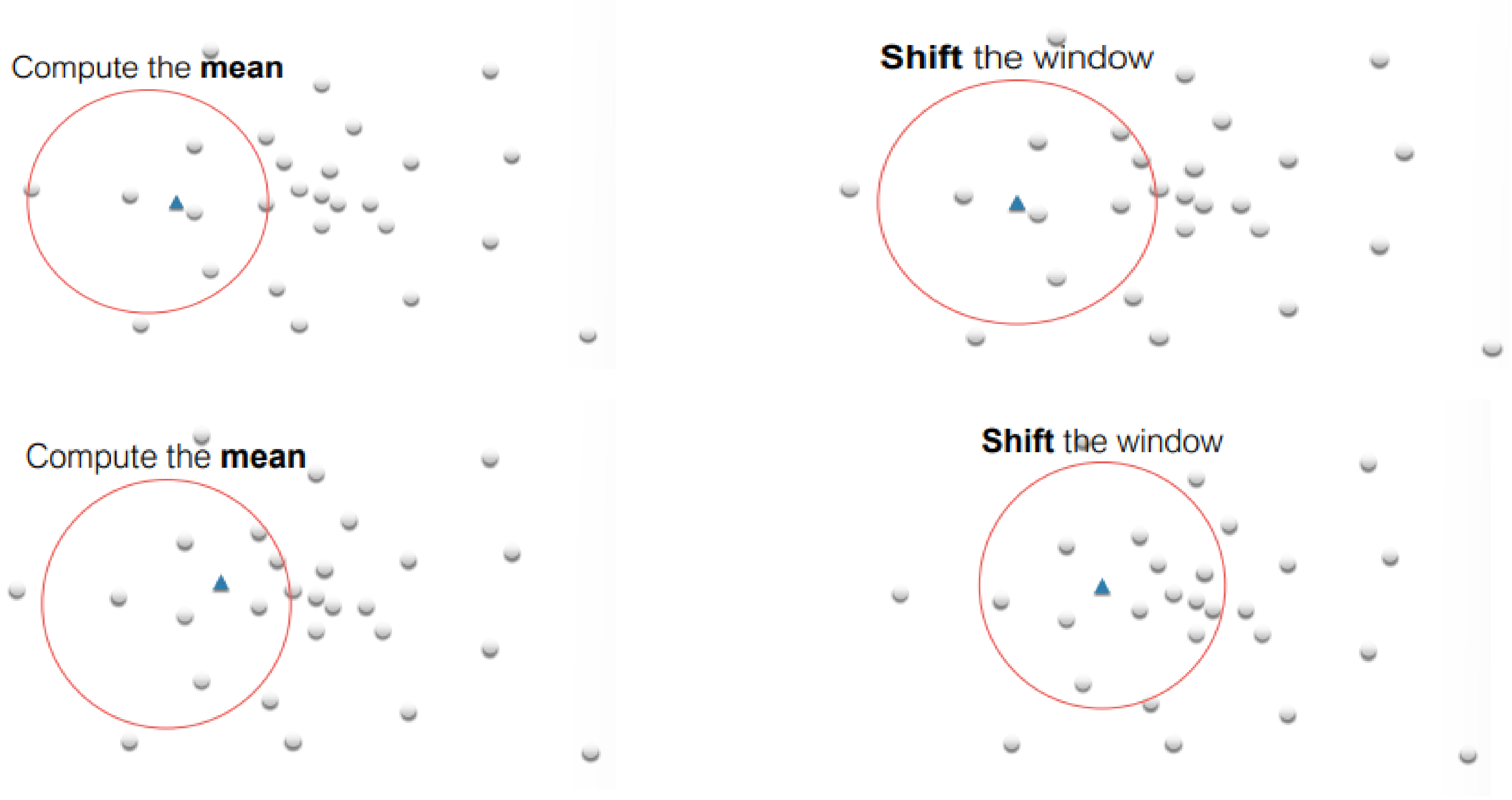
Mean Shift Method



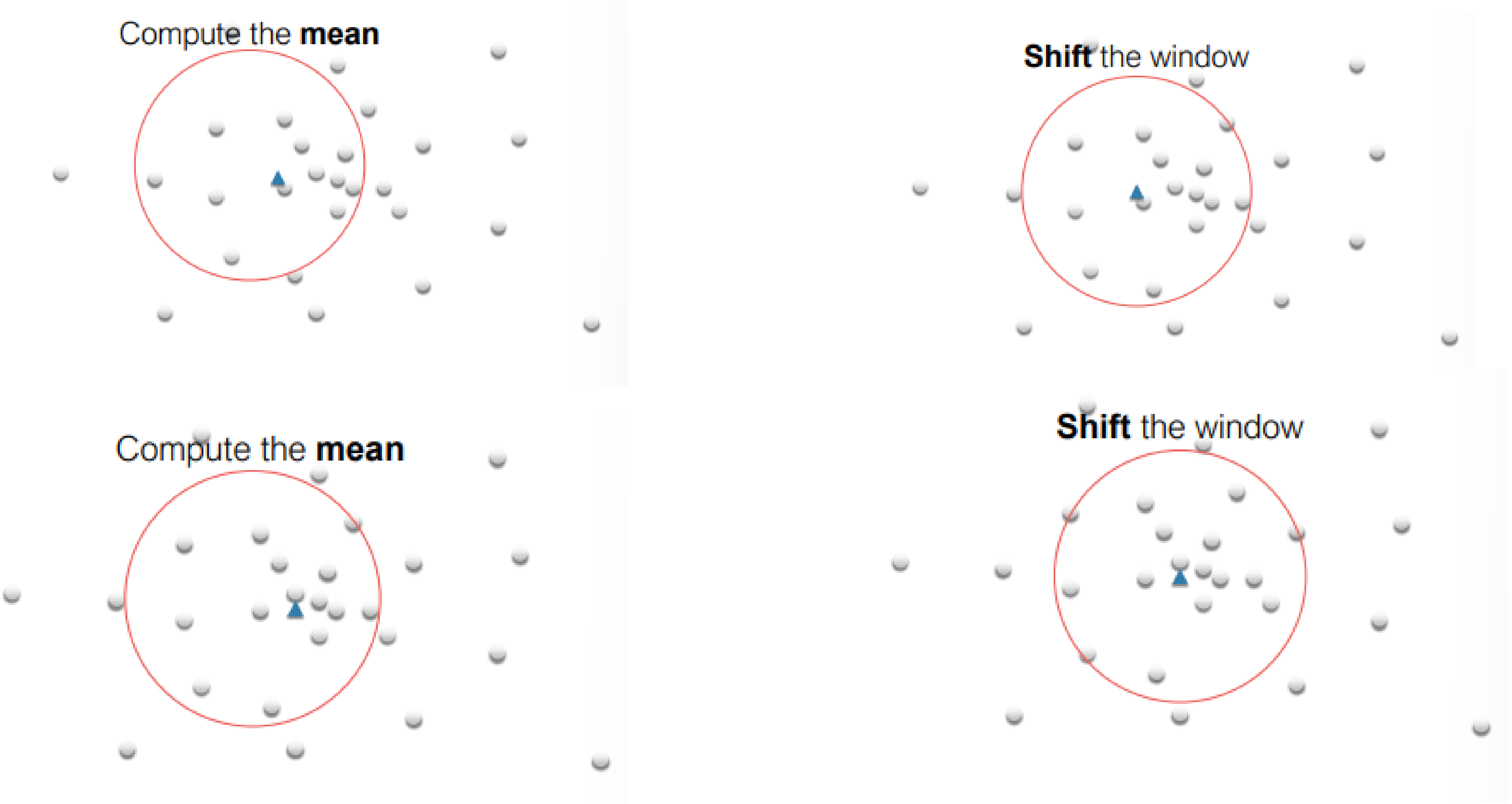
Mean Shift Method



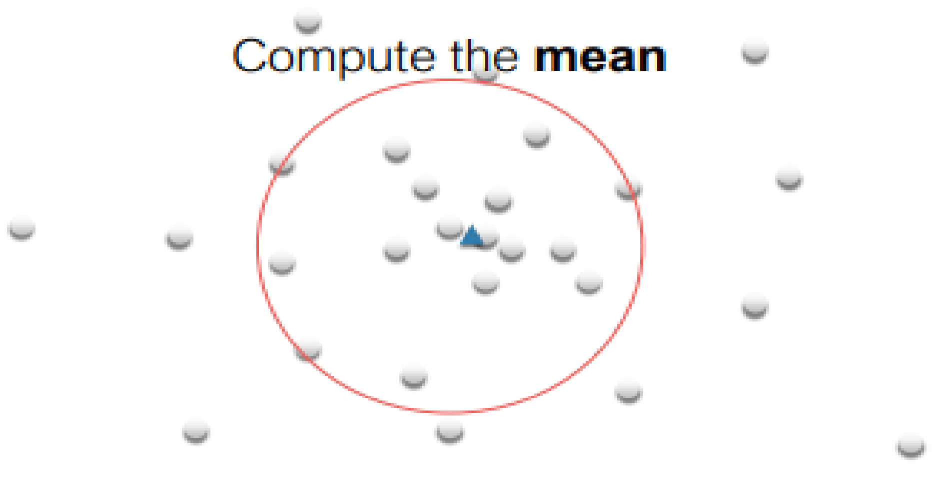
Mean Shift Method

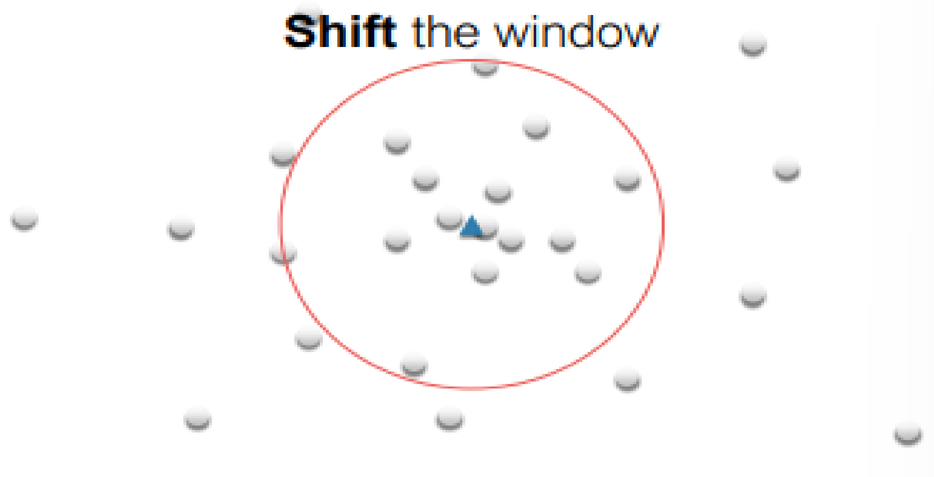


Mean Shift Method



Mean Shift Method





Mean Shift Method

Initialize $\boldsymbol{x}$ — place we start

While $v(\boldsymbol{x}) > \epsilon$ — shift values becomes really small

1. Compute mean-shift

$$m(\boldsymbol{x}) = \frac{\sum_s K(\boldsymbol{x}, \boldsymbol{x}_s)\boldsymbol{x}_s}{\sum_s K(\boldsymbol{x}, \boldsymbol{x}_s)}$$

compute the 'mean'

$$v(\boldsymbol{x}) = m(\boldsymbol{x}) - \boldsymbol{x}$$

compute the 'shift'

2. Update $\boldsymbol{x} \leftarrow \boldsymbol{x} + v(\boldsymbol{x})$ — update the point

Mean Shift Method

Initialize $\boldsymbol{x}$

While $v(\boldsymbol{x}) > \epsilon$

1. Compute mean-shift

$$m(\boldsymbol{x}) = \frac{\sum_s K(\boldsymbol{x}, \boldsymbol{x}_s)\boldsymbol{x}_s}{\sum_s K(\boldsymbol{x}, \boldsymbol{x}_s)}$$

$$v(\boldsymbol{x}) = m(\boldsymbol{x}) - \boldsymbol{x}$$

2. Update $\boldsymbol{x} \leftarrow \boldsymbol{x} + v(\boldsymbol{x})$

Where does this come from?

Kernel density estimate

(radially symmetric kernels)

$$P(\boldsymbol{x}) = \frac{1}{N} c \sum_n k(\|\boldsymbol{x} - \boldsymbol{x}_n\|^2)$$

can compute probability for any point using the KDE!

Mean Shift Method

Initialize $\boldsymbol{x}$

While $v(\boldsymbol{x}) > \epsilon$

1. Compute mean-shift

$$m(\boldsymbol{x}) = \frac{\sum_s K(\boldsymbol{x}, \boldsymbol{x}_s)\boldsymbol{x}_s}{\sum_s K(\boldsymbol{x}, \boldsymbol{x}_s)}$$

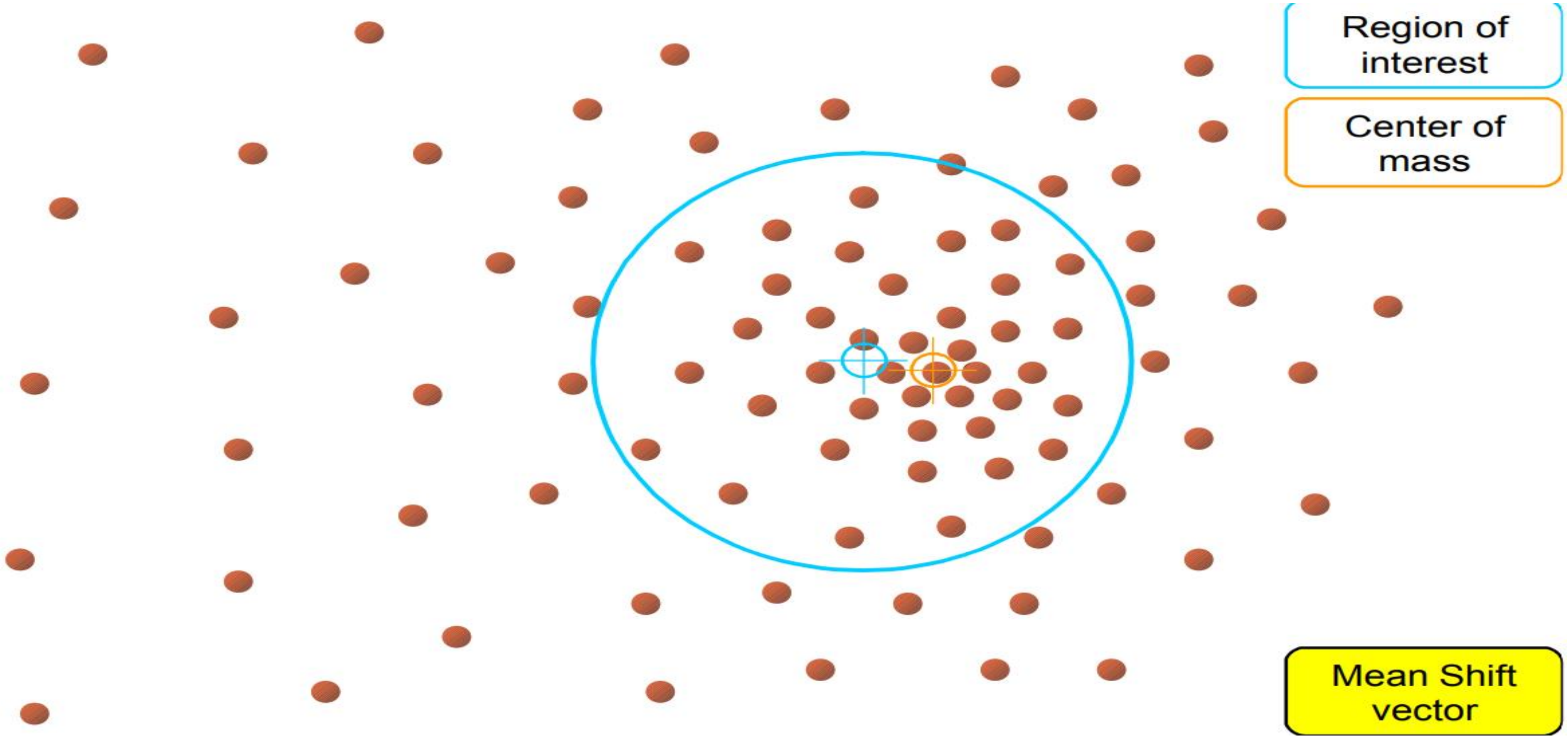
$$v(\boldsymbol{x}) = m(\boldsymbol{x}) - \boldsymbol{x}$$

2. Update $\boldsymbol{x} \leftarrow \boldsymbol{x} + \boldsymbol{v}(\boldsymbol{x})$

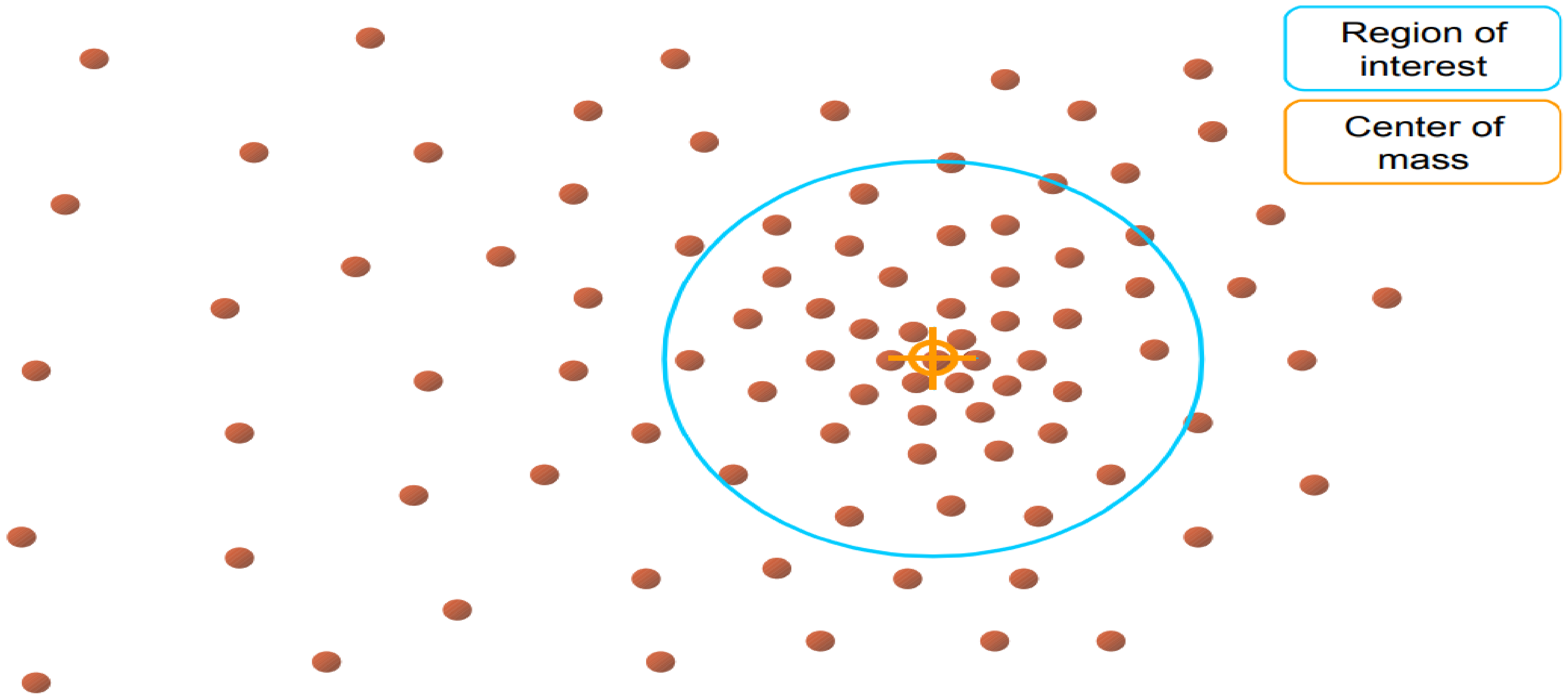
gradient with
adaptive step size

$$\frac{\nabla P(\boldsymbol{x})}{\frac{1}{N} 2c \sum_n g_n}$$

Mean Shift Method



Mean Shift Method

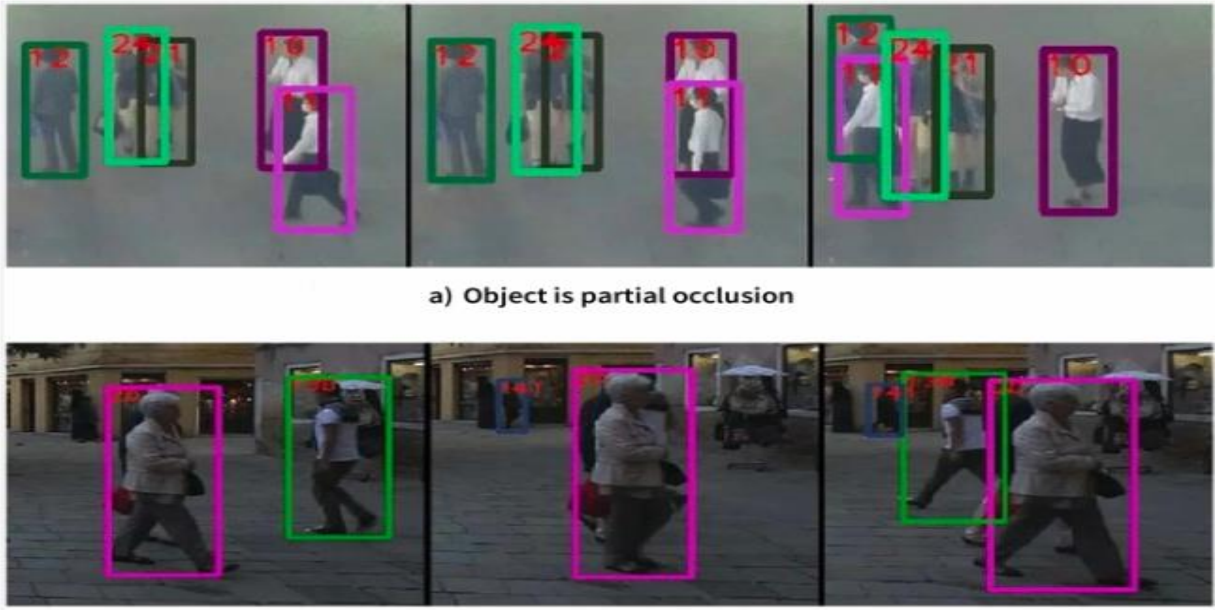


Objective : Find the densest region

Optical Flow Motion Estimation

Applications:

- **Security and Surveillance:** Object tracking can be used in security and surveillance applications to monitor the movements of people and objects of interest, such as vehicles or packages. For example, security cameras can use object tracking to track people or vehicles as they move through space and alert security personnel if any suspicious activity is detected.

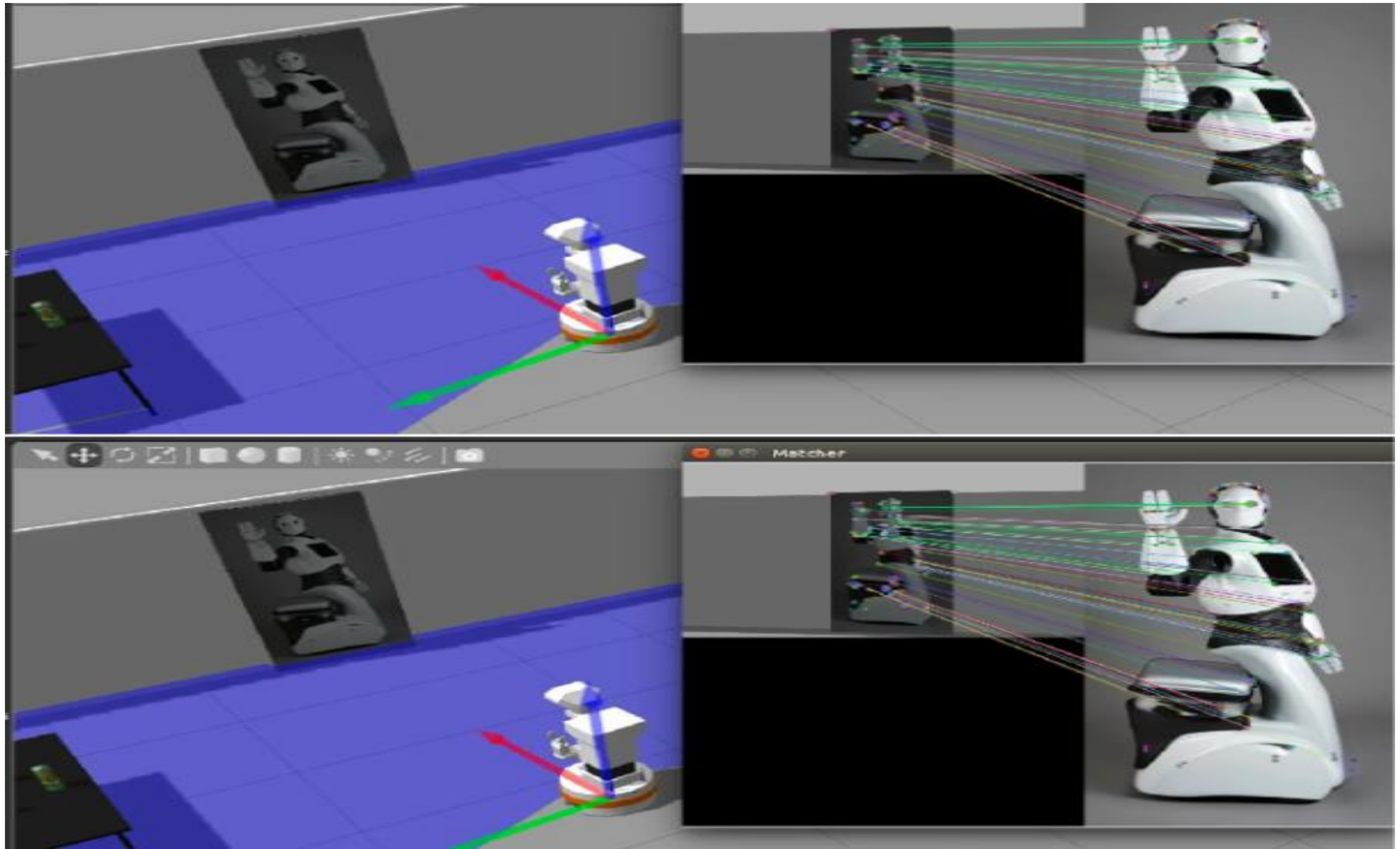


a) Object is partial occlusion

b) Object is full occlusion

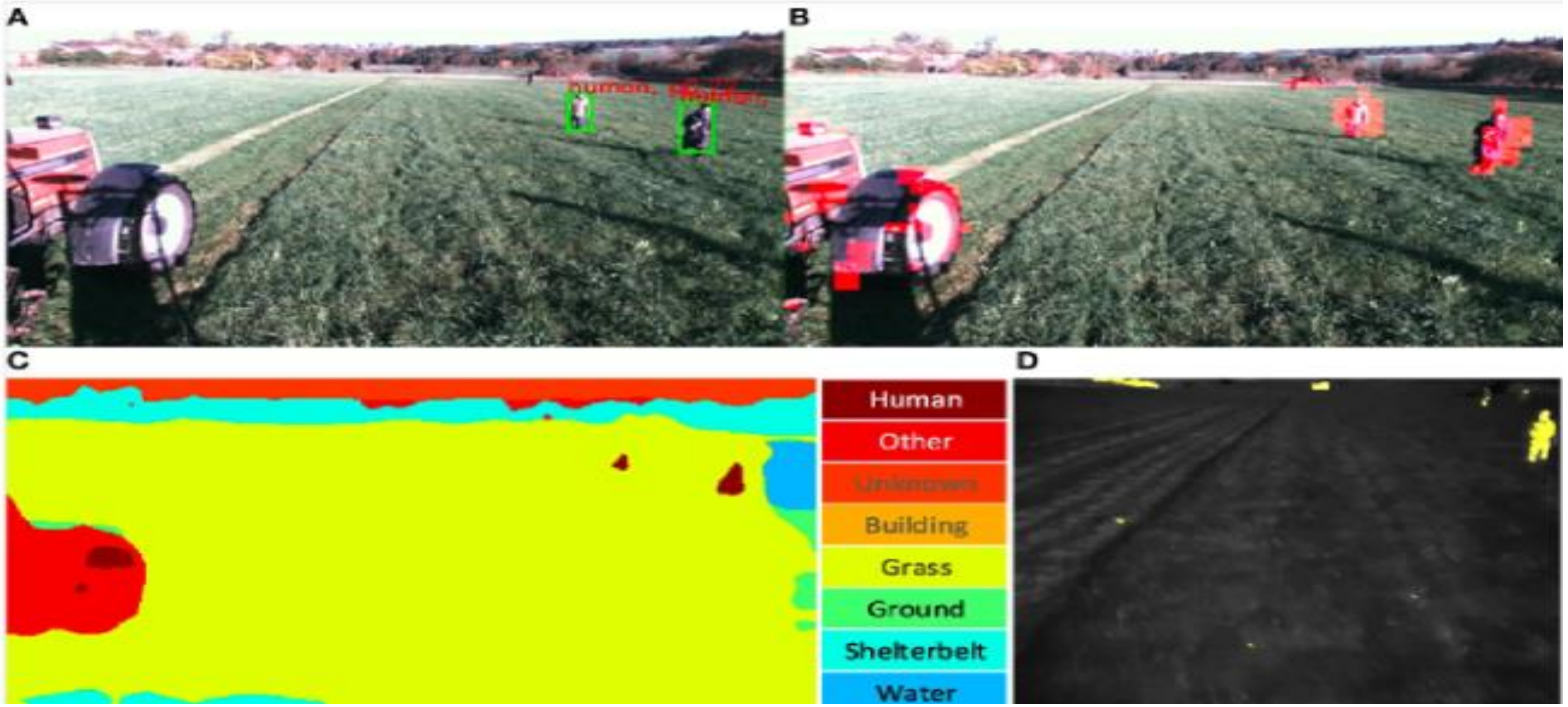
Optical Flow Motion Estimation

- **Robotics:** Object tracking can be used in robotics to help robots interact with their environment and perform object manipulation and navigation tasks. For example, a robot may use object tracking to locate and pick up an object in a cluttered environment.



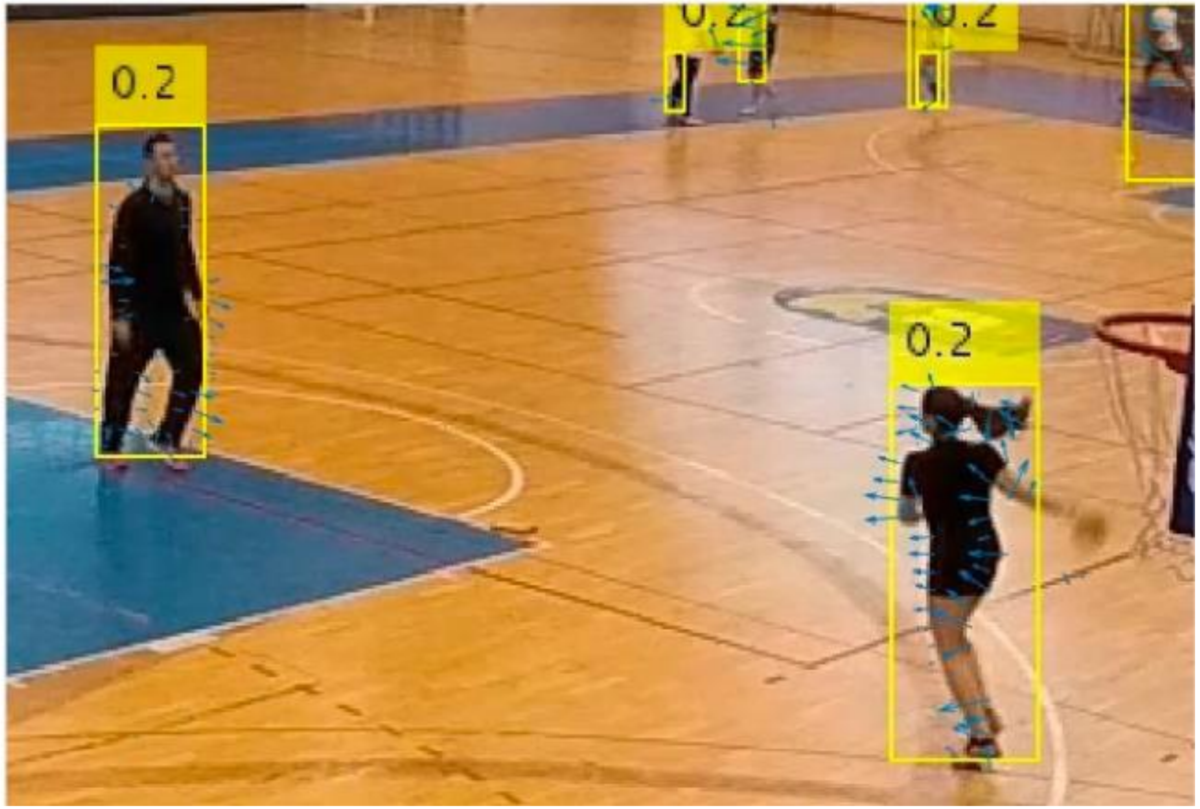
Optical Flow Motion Estimation

- **Agriculture:** Object tracking can be used in agriculture to track the movements of livestock or agricultural equipment. For example, a farmer may use object tracking to monitor the movements of cows in a field or to track the location of tractors and other agricultural equipment.



Optical Flow Motion Estimation

- **Sports:** Object tracking can be used in sports analysis to track players and their movements on the field. For example, a coach may use object tracking to analyze their players' performance and make adjustments to their strategy based on this information.



Dense Motion Estimation

- Robustness to changes in lighting conditions: Object tracking using optical flow can be sensitive to changes in lighting conditions, which can cause errors in object motion estimation. Future research could focus on developing more robust algorithms that can handle changes in lighting conditions and other environmental factors.

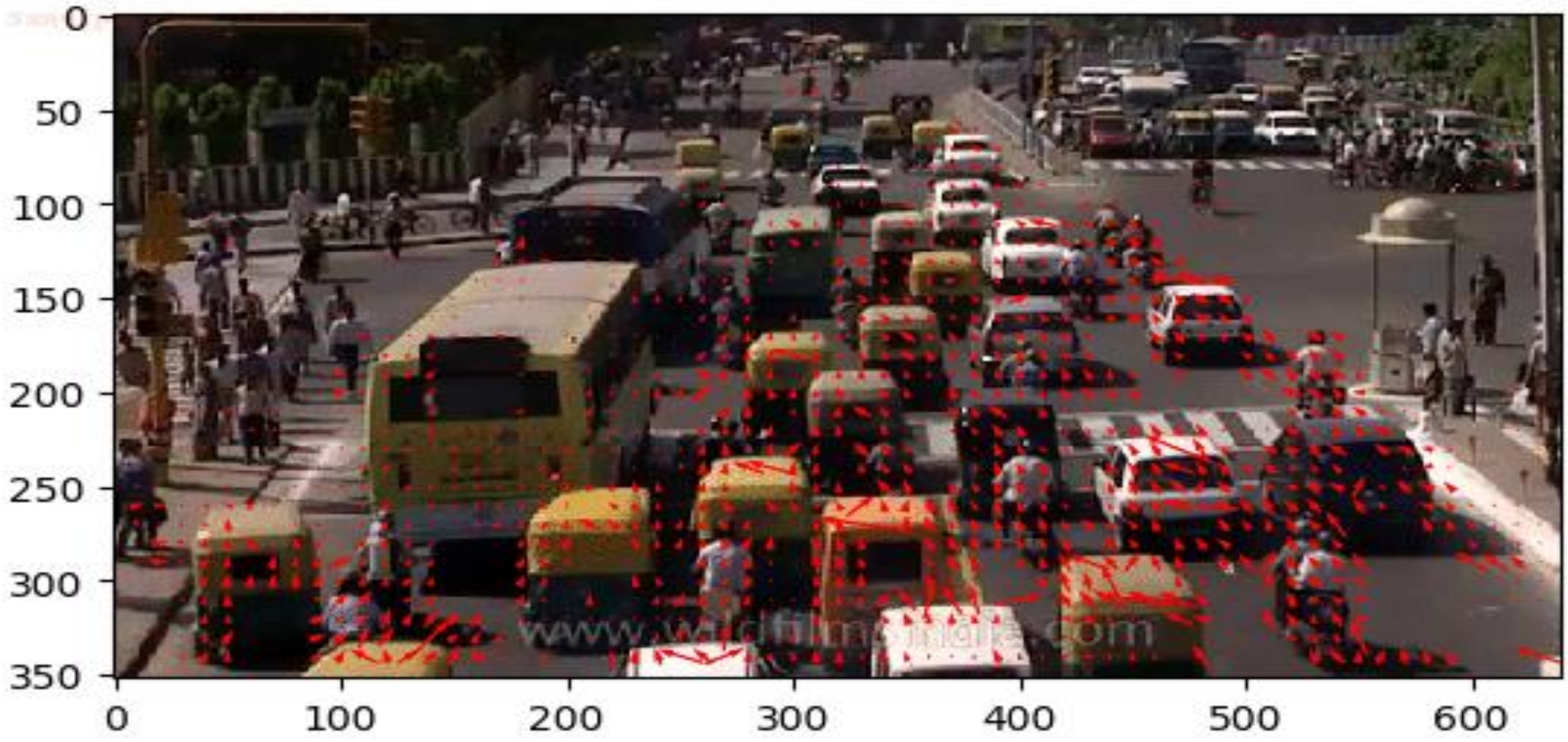
- Handling occlusions: Object tracking using optical flow can also be challenging when partially or fully occluded objects. Future research could focus on developing algorithms that can handle occlusions more effectively and accurately.

Dense Motion Estimation

- Integration with other computer vision techniques: Object tracking using optical flow can be integrated with other computer vision techniques, such as object detection and recognition, to improve tracking accuracy and robustness. Future research could focus on developing algorithms combining different computer vision techniques more effectively.

- Real-time performance: Object tracking using optical flow can be computationally intensive, limiting its real-time performance. Future research could focus on developing more efficient algorithms that can run in real-time on low-power devices such as mobile phones or drones.

Dense Motion Estimation



Conclusion

- In conclusion, optical flow motion estimation is a fundamental technique in computer vision that enables us to estimate the motion of objects in video sequences.
- Optical flow motion estimation has numerous applications, including object tracking, video compression, and 3D reconstruction.
- Some common optical flow algorithms include Lucas-Kanade, Farneback, and Horn-Schunck, which differ in their assumptions about the underlying motion and the computation of flow vectors.
- Overall, optical flow motion estimation is a powerful technique with a wide range of applications in computer vision.