Optical Flow

Fiseha B. Tesema, PhD

Recap

- Stereo Vision
- Geometry for a simple stereo system
- Epipolar Geometry
- Multiview Stereo

Outline

Optical Flow:

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

- Topics:
  - Motion Field and Optical Flow:
  - Optical Flow Constraint Equation
  - Lukas Kanade Method
  - Coarse to Fine Flow Estimation
  - Alternative Approach: Template Matching
  - Dense and Sparse Optical Flow
  - Application of Optical Flow

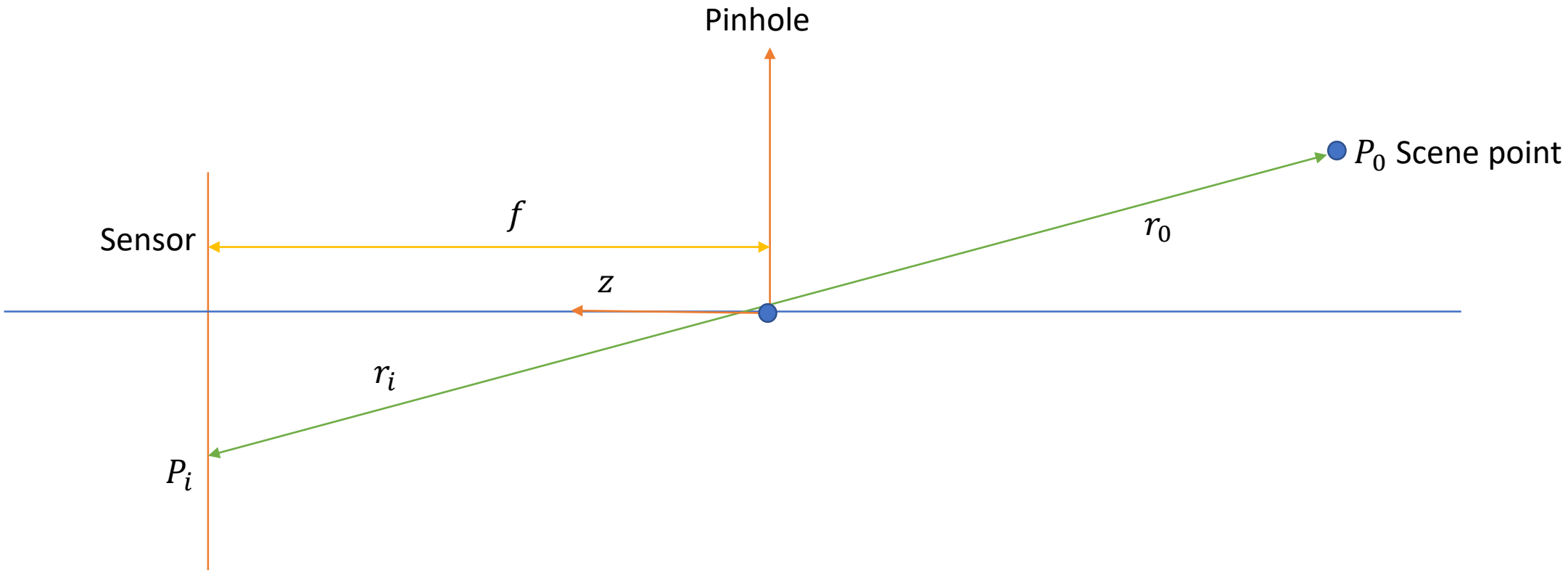
Motion Field and Optical Flow

Motion Field and Optical Flow

- Motion field is the projection of a point's movement in a 3D scene onto a 2D image through perspective projection.
- We cannot directly measure the motion field; instead, we measure the motion of brightness patterns in images, known as optical flow.
- Optical flow represents the motion of brightness patterns in an image.

Motion Field

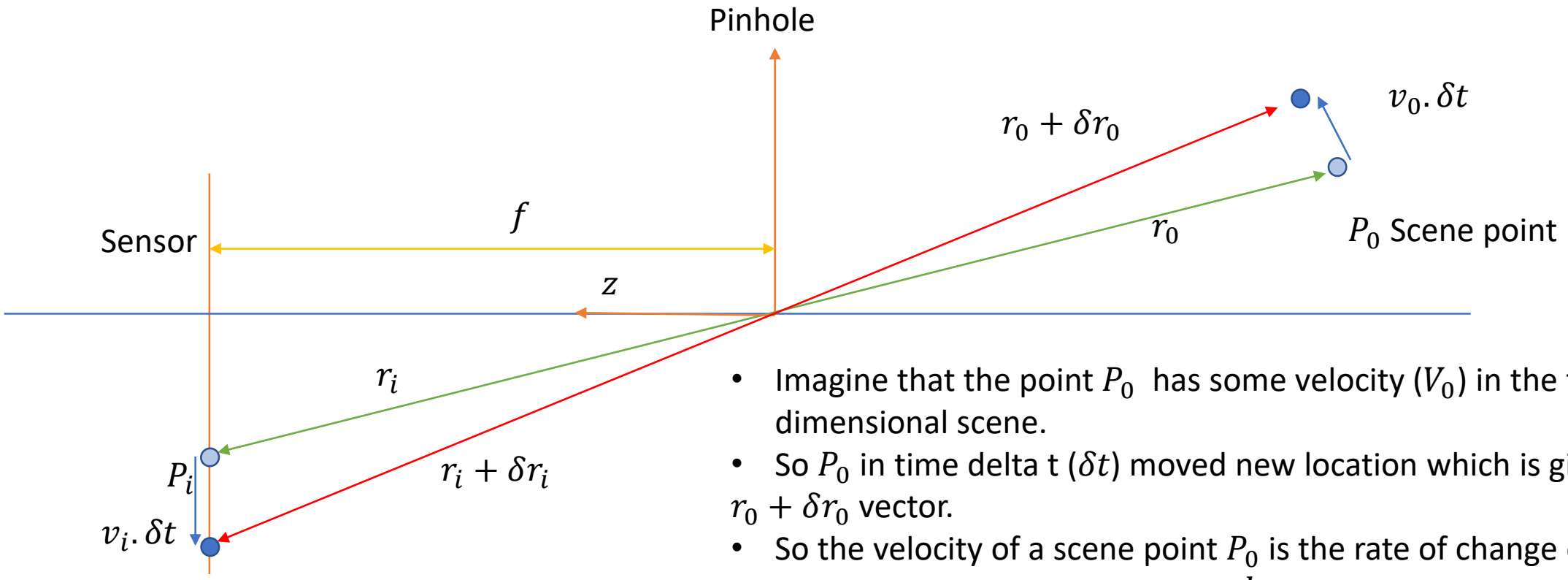
- Image velocity of a point that is moving in the scene



Horn 1981

Motion Field

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



(Motion Field)

- Imagine that the point P_0 has some velocity (V_0) in the three dimensional scene.
- So P_0 in time delta t (δt) moved new location which is given by the $r_0 + \delta r_0$ vector.
- So the velocity of a scene point P_0 is the rate of change of r_0 over d_t.

$$v_0 = \frac{dr_0}{d_t}$$

Horn 1981

Motion Field

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

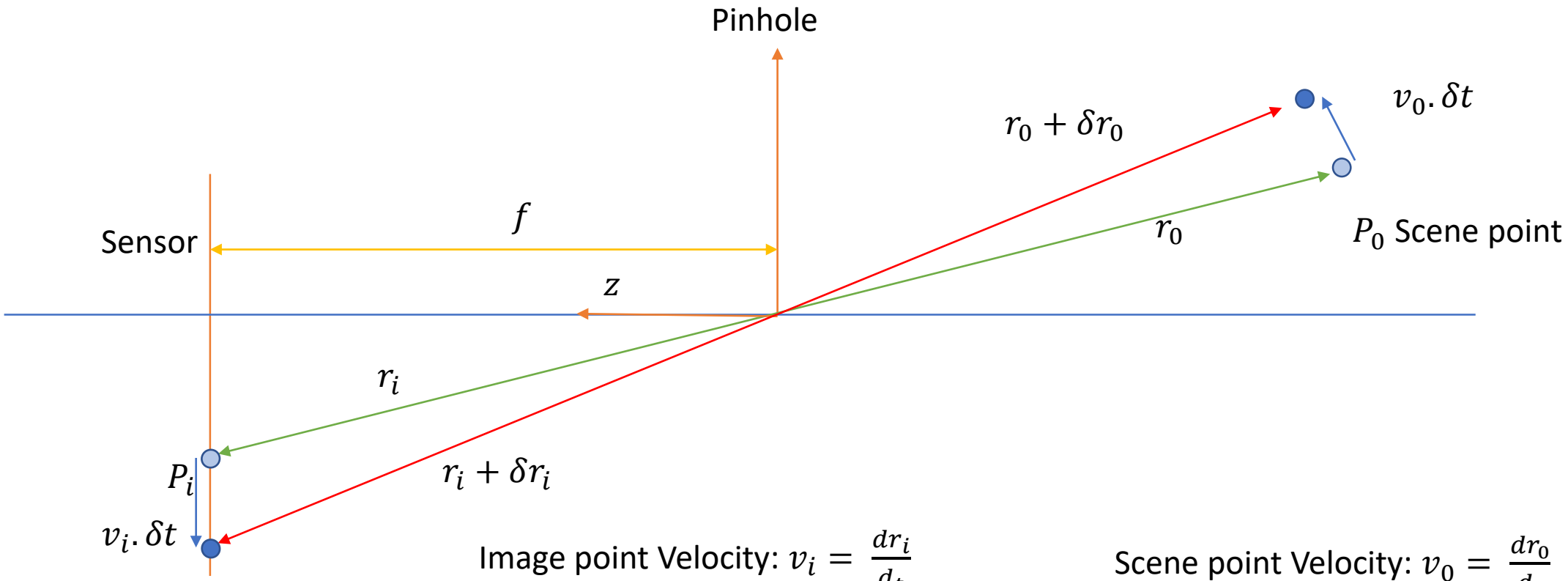


Image point Velocity: $v_i = \frac{dr_i}{d_t}$

Scene point Velocity: $v_0 = \frac{dr_0}{d_t}$

(Motion Field)

Horn 1981

Motion Field

- We want to relate v_i to v_0. How do we relate?

Image point Velocity: $v_i = \frac{dr_i}{d_t}$ Scene point Velocity: $v_0 = \frac{dr_0}{d_t}$

Perspective projection equation

Motion Field

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

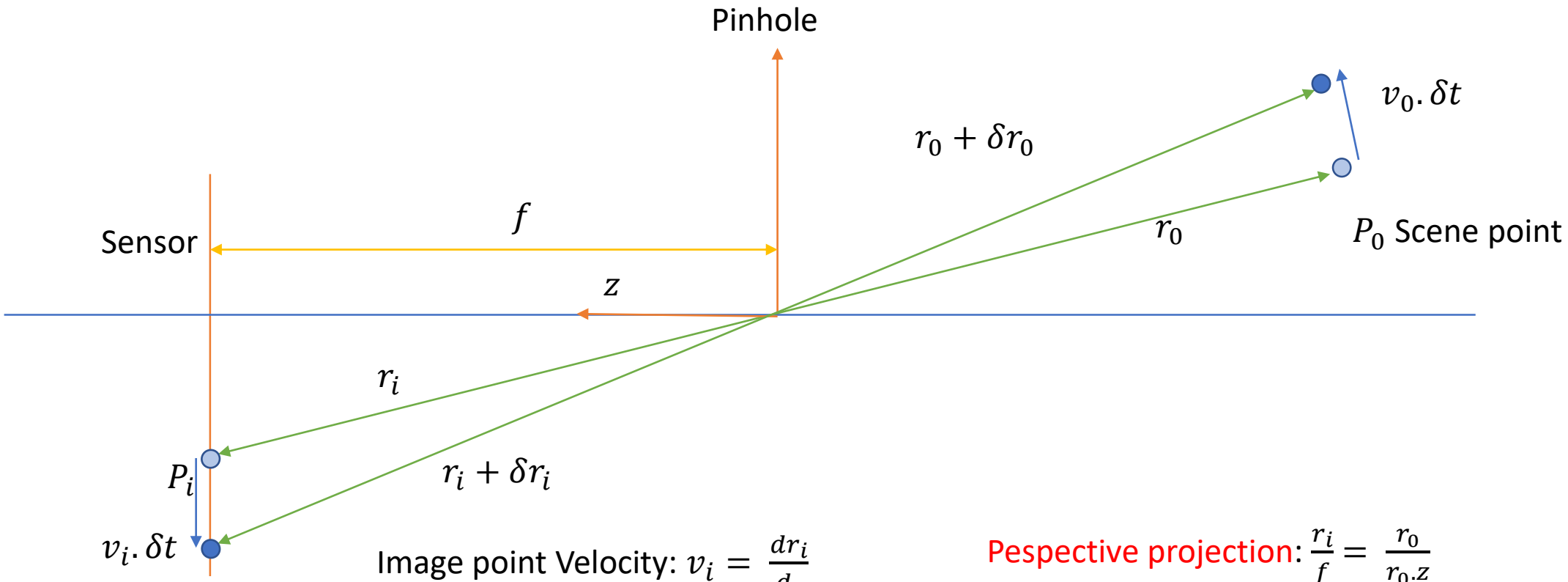


Image point Velocity: $v_i = \frac{dr_i}{d_t}$

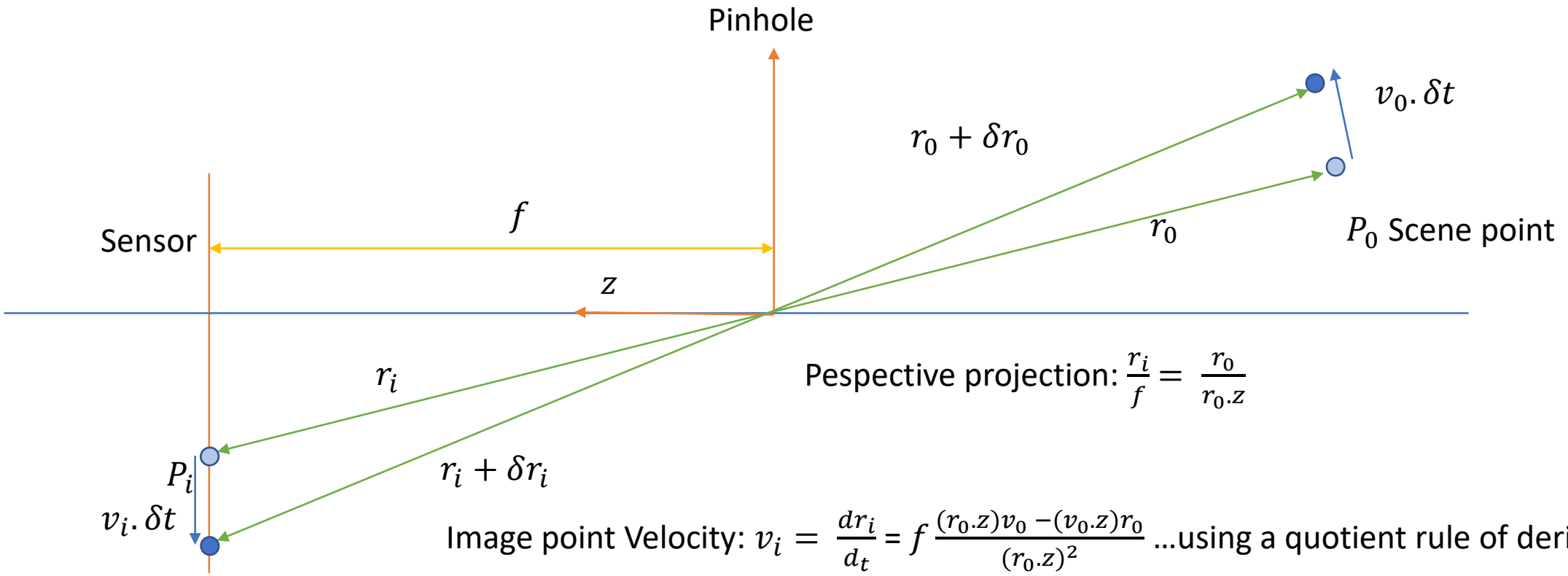
(Motion Field)

Pespective projection: $\frac{r_i}{f} = \frac{r_0}{r_0.z}$

[Horn 1981]

Motion Field

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



Pespective projection: $\frac{r_i}{f} = \frac{r_0}{r_0.z}$

$$v_0 = \frac{dr_0}{d_t}$$

Image point Velocity: $v_i = \frac{dr_i}{d_t} = f\frac{(r_0.z)v_0 - (v_0.z)r_0}{(r_0.z)^2}$ ...using a quotient rule of derivates

$v_i = f\frac{(r_0 \times v_0) \times z)}{(r_0.z)^2}$ .. Simplified format

(Motion Field)

Horn 1981

Motion Field

$$v_i = f\frac{(r_0 \times v_0) \times z)}{(r_0.z)^2}$$

- Then we can use this expression to figure out what is the velocity of the point is going to be in the image, which is a v_i.
- This v_i is the motion field corresponding to the moving point in the 3D scene.
- This is what we would like to measure.
  - Well, unfortunately for us, there's no guarantee that we can measure the v_i or the motion field because in images all we have are brightness patterns.
- What we can measure or hope to measure is a motion of brightness patterns in the scene.

Optical Flow

- Motion of brightness patterns in the image

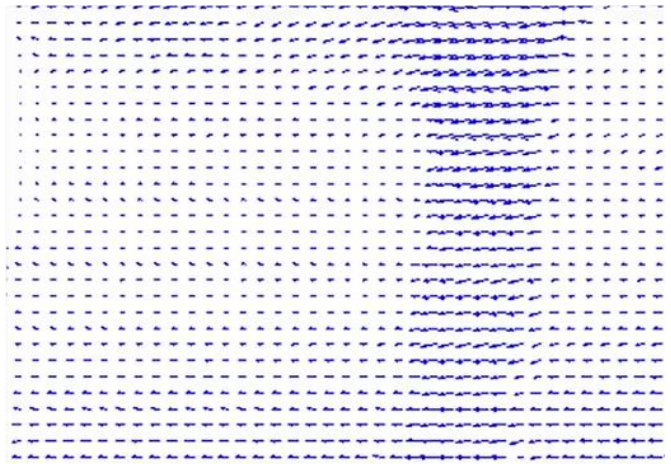
Image Sequence (2 frames)

- We have two images which is taken by a moving camera.
- Motion field: take each point in the first image and figure it out where it ended up in the second image.
- Optical flow: take a brightens patterns in one image and see where the brightens pattern ends up in the second image.
  - Our objective is to develop algorithm to do that!

Optical Flow

- Motion of brightness patterns in the image

Image Sequence (2 frames)



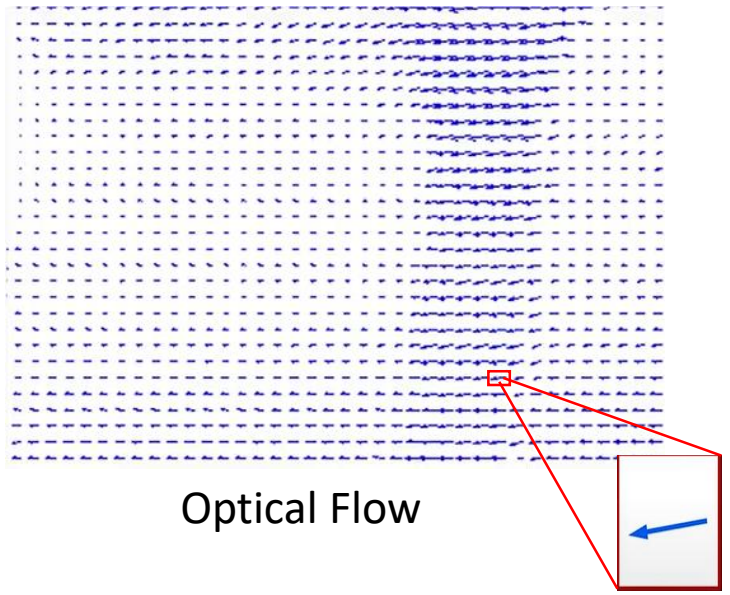
Optical Flow

- OF algorithm, Intended to develop algorithm which
  - Take the pattern, the brightness pattern in one image.
  - Observe where the brightness pattern ends up in the second image.
  - Generate the flow shown in the image

Optical Flow

- Motion of brightness patterns in the image

Image Sequence (2 frames)



Optical Flow

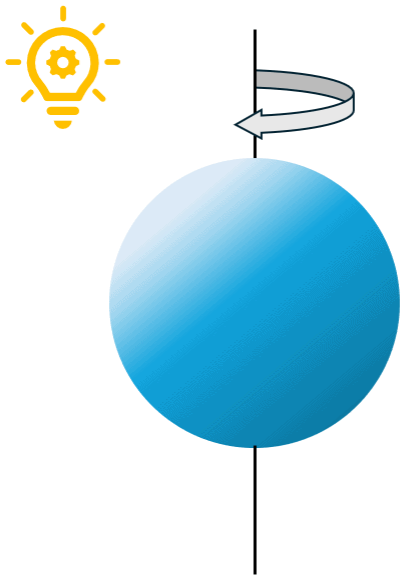
Velocity of brightness pattern

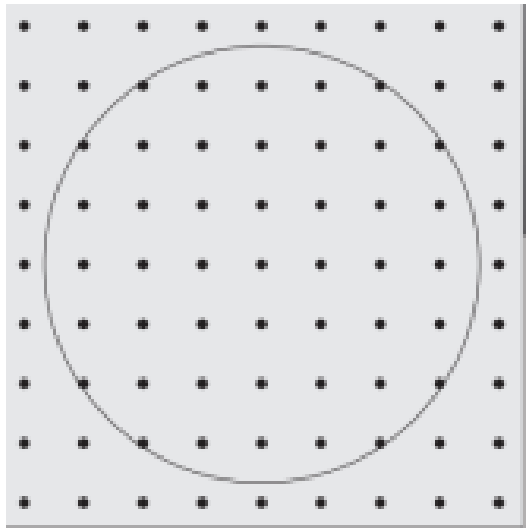
- The motion of brightness patterns is the optical flow.
- Each pixel you have a vector which tells you what the optical flow at that point.
  - The length of the vector tells you:
    - how fast it's moving and
  - Its direction tells you
    - Which direction it's moving in on the image plane.

Ideally, optical flow is equal to motion field. Unfortunately not all the time!

When is Optical Flow ≠ Motion Field?

- Example 1: Let's consider a rotating sphere and fixed light:

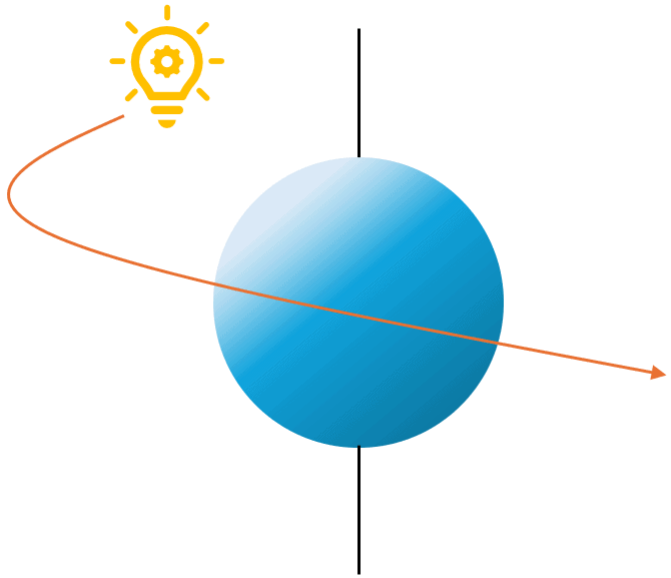


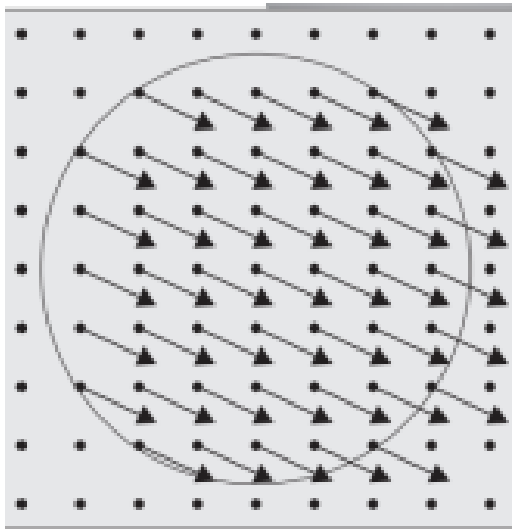


- A uniform rotating sphere with a fixed light source has no optical flow but a non-zero motion field.

When is Optical Flow ≠ Motion Field?

- Example 2: Let's now consider the same sphere but fixed and the light source moving around it:

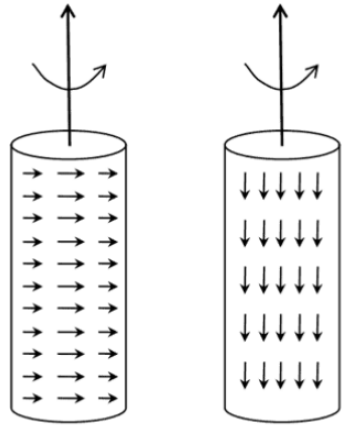




- A fixed uniform sphere with a light source moving around it has a non-zero optical flow but a zero motion field.

When is Optical Flow ≠ Motion Field?

- Example 3: An interesting example is the barber-pole illusion represented below. There is motion filed and optical flow but optical flow and motion field don't correspond to each other.



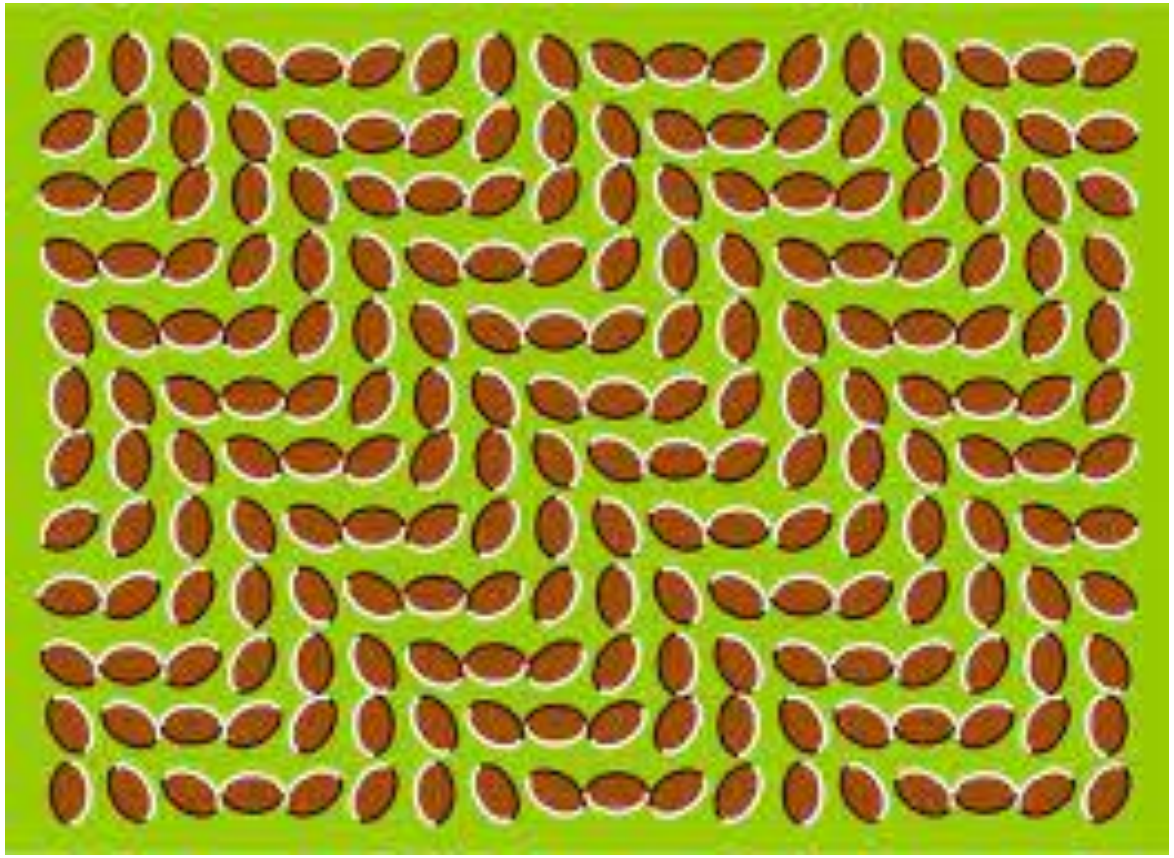
Barber Pole Illusion [https://www.baeldung.com/cs/motion-field-optical-flow]

- The barber pole rotates along the vertical axis; hence, all points on the cylinder move horizontally. Conversely, if you record the video and observe the brightness patterns move in a vertical direction.

Motion Illusion

- So this idea of incorrectly perceiving motion also applies to human beings.

You have optical flow in your visual system. But there is no motion field.



Donguri wave illusion

Optical Flow Constraint Equation

Optical Flow Constraint Equation

- Estimating optical flow?
  - It turns out it's a hard problem.
  - It's an under constraint problem.
- We will discuss:
  - Drive a constraint equation is known as an optical flow constraint equation,
  - Then develop an algorithm for estimating the optical flow at each point that uses the derived constrain equation.

Optical Flow

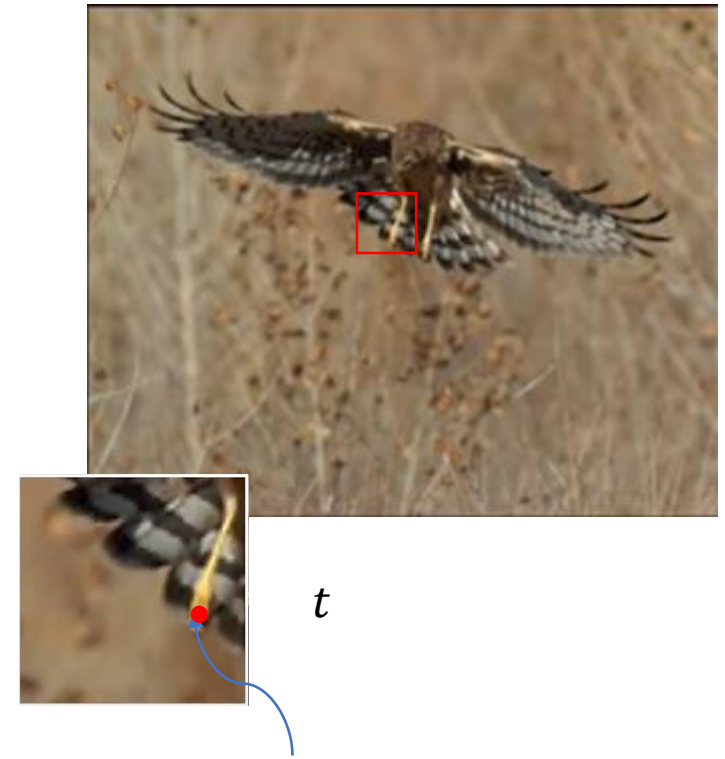
t

$t + \delta t$

- Let's say we have two images of a scene in quick succession where δt is small. Horn 1981

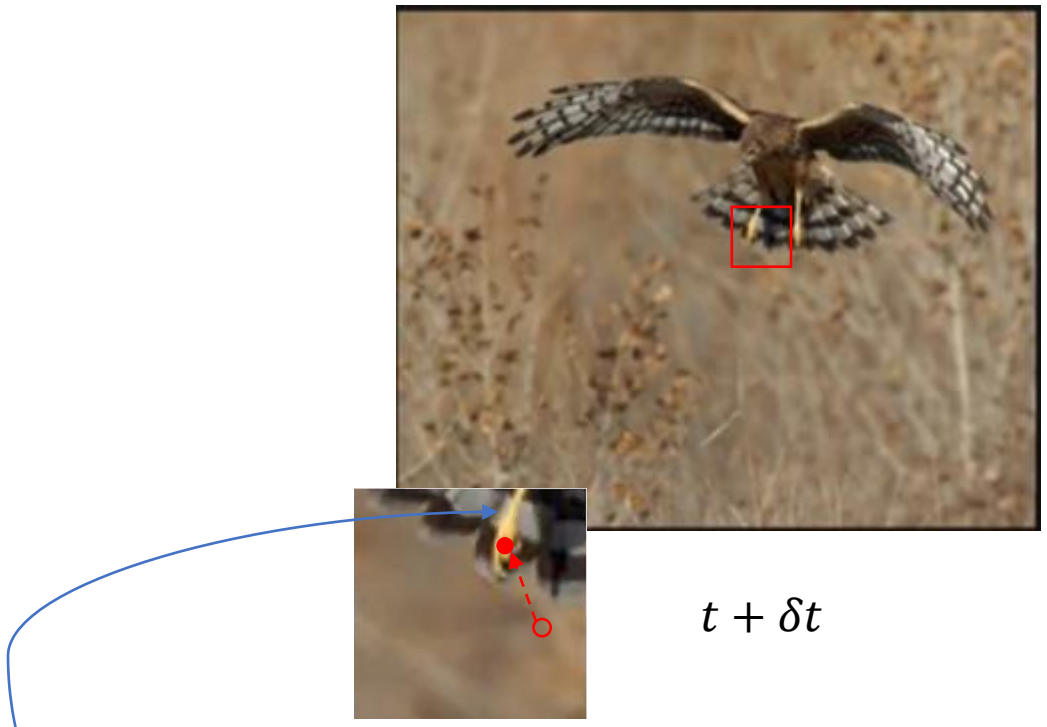
Optical Flow Constraint Equation

- Let's focus our attention on single point within this window.



t

1) Point location (x, y)



$t + \delta t$

2) Moved to New pixel location$(x + \delta x, y + \delta y)$

Displacement: $(\delta x, \delta y)$

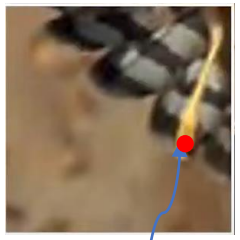
(Speed of the point in the x and y direction) Optical Flow $(u, v) = (\frac{\delta x}{\delta t}, \frac{\delta y}{\delta t})$

[Horn 1981]

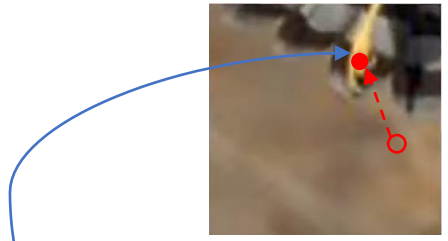
Optical Flow Constraint Equation

- Optical Flow (u, v)= $(\frac{\delta x}{\delta t}, \frac{\delta y}{\delta t})$, optical flow crossponding to the point.
- (u, v) what we are going to measure.
- How are we going to measure?
  - We need to make a couple of assumptions to solve the problem.

Optical Flow Constraint Equation



$I(x, y, t)$



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

Assumption #1: (Constant brightness assumption)

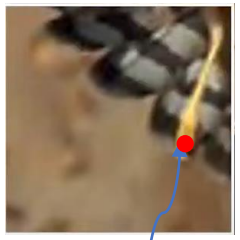
- Brightness of image point remains constant over time. At least between consecutive image which being captured in quick succession.

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

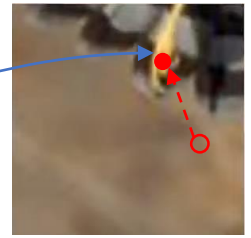
- But in reality as point move in space the brightness is not constant. That is what makes optical flow a difficult problem.
- So what that tells us is that the intensity of this image (I) point tried two different time is equal.

Horn 1981

Optical Flow Constraint Equation| Optical Flow



$I(x, y, t)$



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

Assumption #2: (Small displacement assumption)

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

$I(x + \delta x, y + \delta y, t + \delta t)$ ……Do linear approximation

Horn 1981

Taylor Series Expansion

- The Taylor series expansion applies to any function which is infinitely differentiable.
- Simply means that all its derivatives exists.

$$f(x+\delta x) = f(x) + \frac{\partial f}{\partial x}\delta x + \frac{\partial^2 f}{\partial x^2}\frac{\delta x^2}{2!} + \ldots + \frac{\partial^n f}{\partial x^n}\frac{\delta x^n}{n!}$$

Expand a function as infinite sum of its derivatives (∂)

In Tylor series If δx is small the higher order $\delta x^2 \ldots . \delta x^n$ is very small

$$f(x+\delta x) = f(x) + \frac{\partial f}{\partial x}\delta x + O(\delta x^2)$$

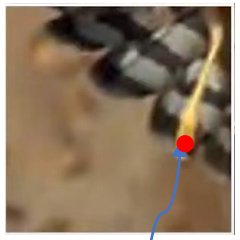
The higher ordered almost zero

- Linear Approximation(first order Tylor approximation)
- It is linear approximation because it is linear in displacement δx.

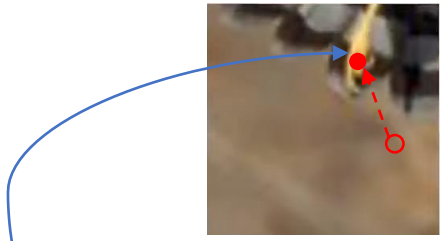
For a function of three variables with small $\delta x\ \delta y\ \delta t$:

$$f(x+\delta x, y+\delta y, t+\delta t) \approx f(x,y,t) + \frac{\partial f}{\partial x}\delta x + \frac{\partial f}{\partial y}\delta y + \frac{\partial f}{\partial t}\delta t$$

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

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

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

[Horn 1981]

Optical Flow Constraint Equation

$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t)$ …(1) Brightens Assum.

$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + I_x \delta x + I_y \delta y + I_t \delta t$…(2) Displacement Assum.

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

Divide by $\delta t \; and \; take \;\; limit \; as \; \delta t \; \rightarrow 0$: $I_x \frac{\delta x}{\delta t} + I_y \frac{\delta y}{\delta t} + I_t = 0$…

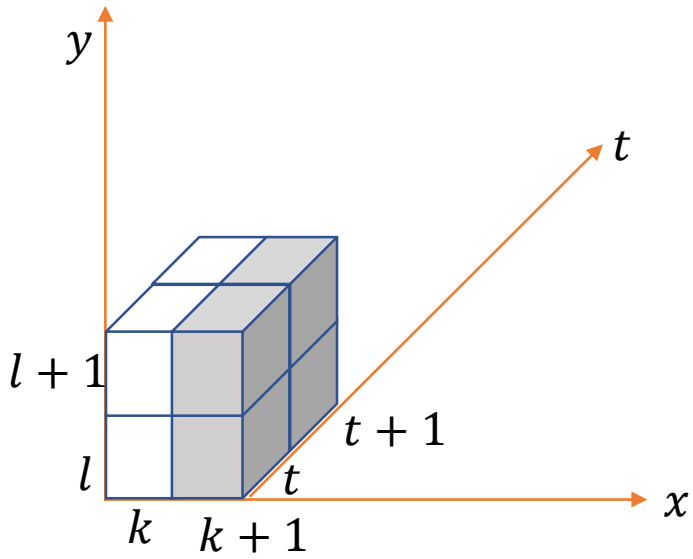
- replace $\frac{\delta x}{\delta t}$ with u component and
- $\frac{\delta y}{\delta t}$ with v component

Optical Flow Constrained Equation: $I_x u + I_y v + I_t = 0 \; (u, v): Optical \; Flow$

(I_x, I_y, I_t) Can be easily computed from two frames taken in **quick succession using finite differences.**

Computing Partial Derivatives (I_x, I_y, I_z)

The derivative of brightness in x direction



$$I_x(k,l,t) =$$

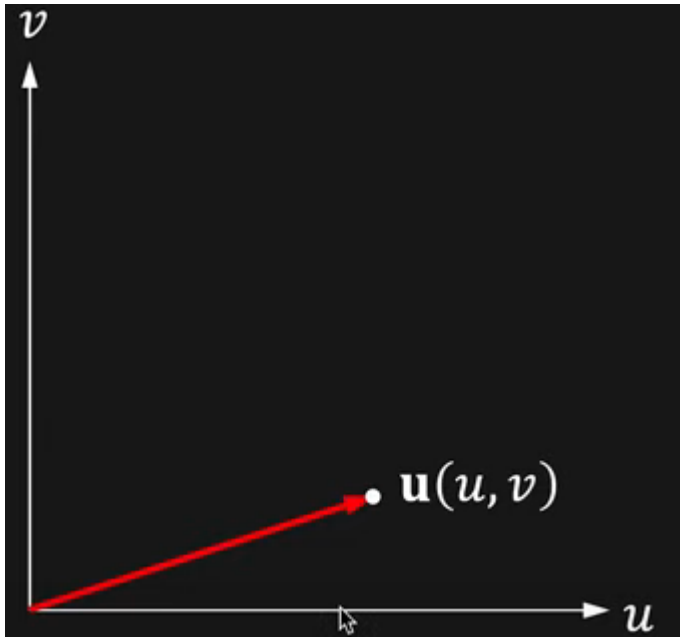
$$\frac{1}{4}[I(k+1,l,t) + I(k+1,l,t+1,) + I(k+1,l+1,t) + I(k+1,;l+1,t+1)]$$

$$-\frac{1}{4}[I(k,l,t) + I(k,l,t+1,) + I(k,l+1,t) + I(k,l+1,t+1)]$$

Similarly find $I_y(k,l,t)$ *and* $I_t(k,l,t)$

Geometric Interpretation of OF Constraint Equation

- Assume (u, v) space and the OF at particular point in the image is the vector $\boldsymbol{u}$.

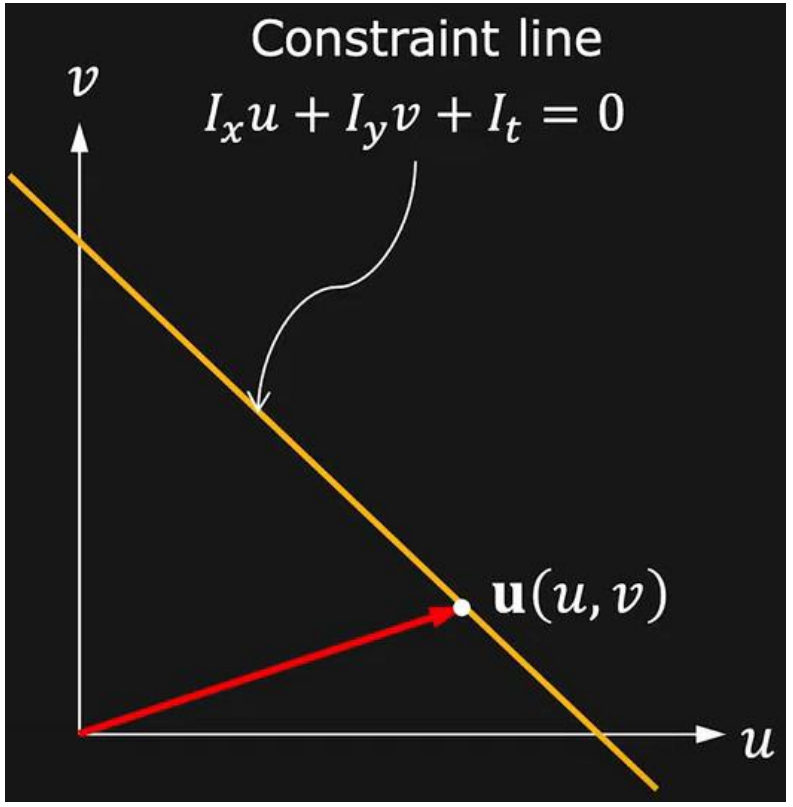


Geometric Interpretation of OF Constraint Equation

- What we have is only Constraint line, which is equation of straight line,

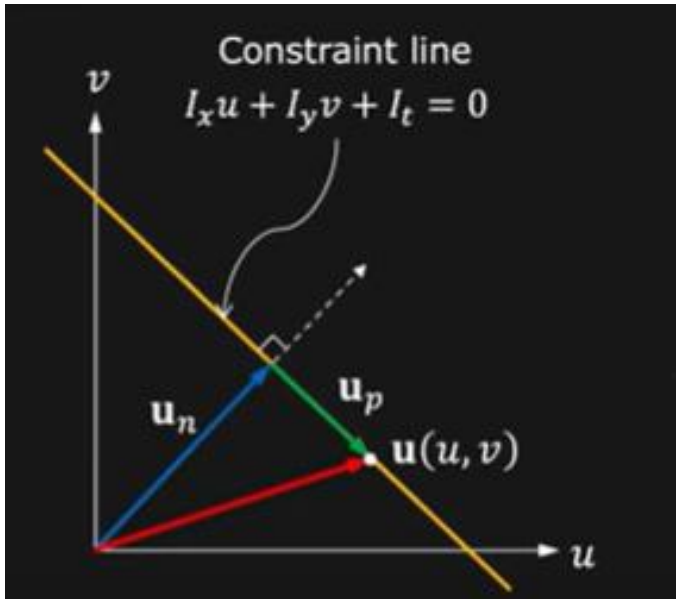
$$I_x u + I_y v + I_t = 0$$

- Given I_x, I_y and I_t we have straight line in (u, v).
- For any point (x,y) in the image, its optical flow (u,v) lies on this line but we don't know where exactly it lies.
  - This is what makes optical flow estimation problem **under constrained problem**.



Geometric Interpretation

- We can conceptually is just split up the optical flow vector you into two components.



$\boldsymbol{u} = u_n + u_p$

$\boldsymbol{u}_n$: $Normal\ Flow$: $Which\ is\ noraml\ to\ constriant\ line$

$\boldsymbol{u}_p$: $Parallel\ Flow$: Parallel to constriant line

what can be estimated?

Geometric Interpretation

- The Normal Flow u_n can be computed from the constraint line.
  - Direction of Normal Flow:
    - Unit vector in the direction that is perpendicular to the constrain/straight line:

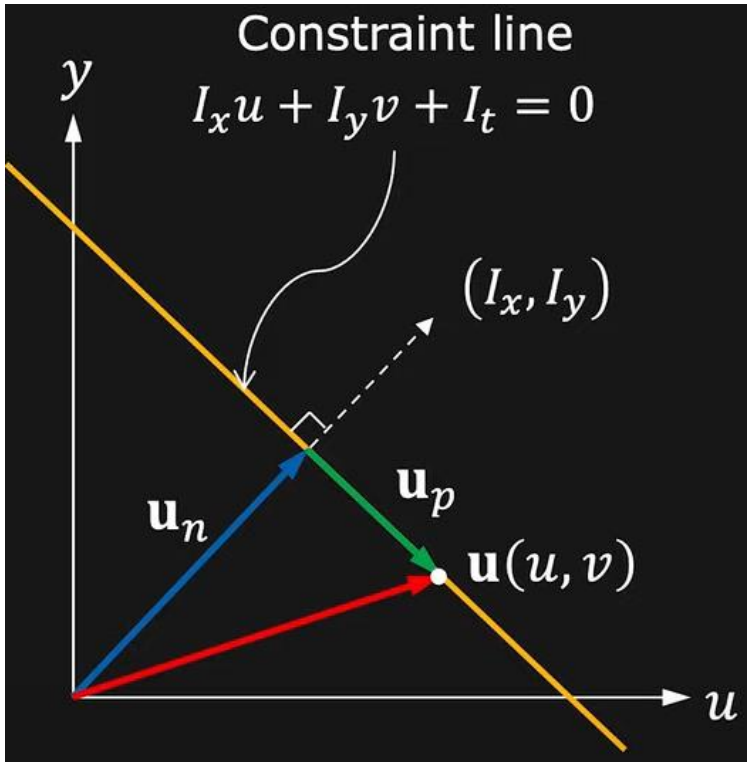
$\hat{u}_n = \frac{(I_x, I_y)}{\sqrt{I_x^2 + I_y^2}}$ … from the properties of straight line.

  - Magnitude of Normal Flow:
    - The shortest distance of origin from the constraint line:

$|u_n| = \frac{|I_t|}{\sqrt{I_x^2 + I_y^2}}$

  - Multiply the two equation to get u_n

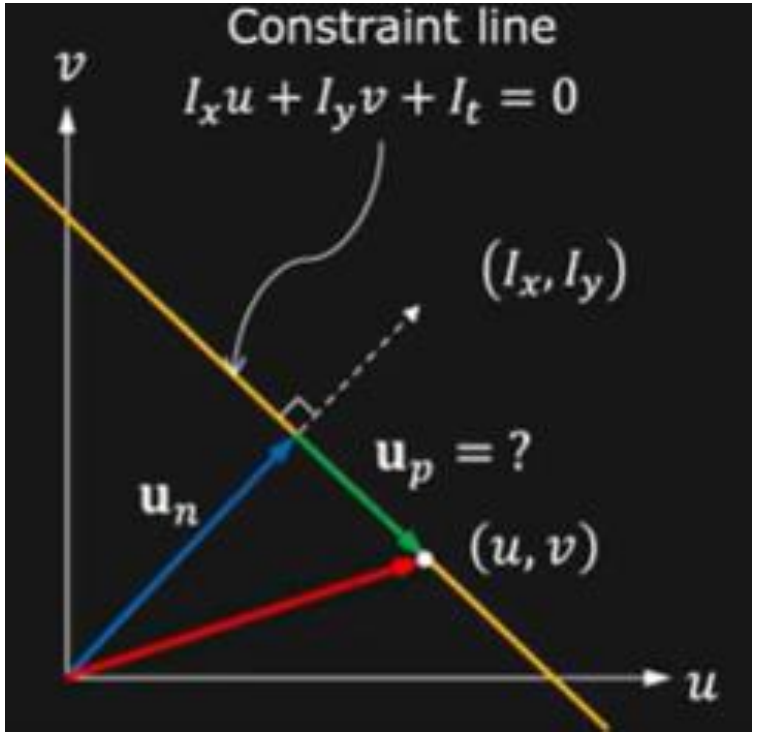
$u_n = \frac{|I_t|}{I_x^2 + I_y^2}(I_x, I_y)$



Parallel Flow

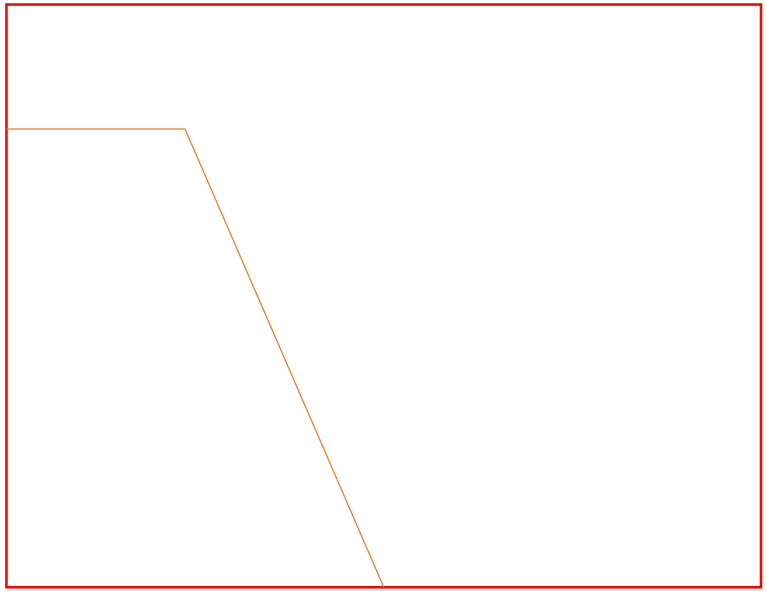
- We cannot determine u_p, the optical flow component parallel to the constraint line.

- The aforementioned issue is called the aperture problem – the motion of an edge as seen through an aperture is essentially ambiguous.
- How did this Manifest in terms of real situation in the algorthim?
- And, Humans



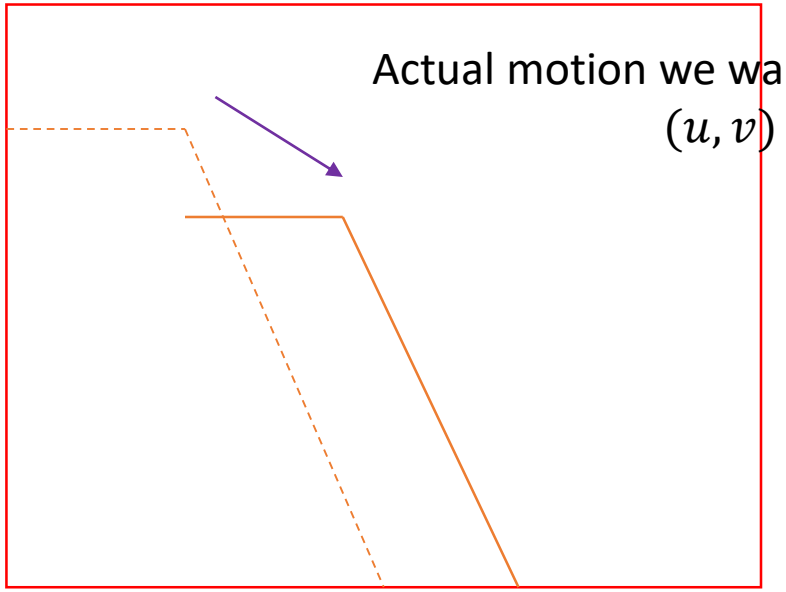
Aperture problem

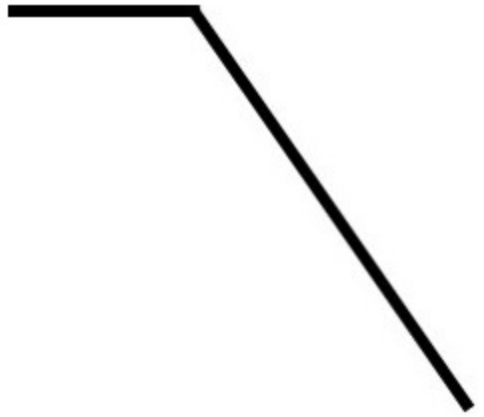
- Let’s say this is object of interest



Aperture problem

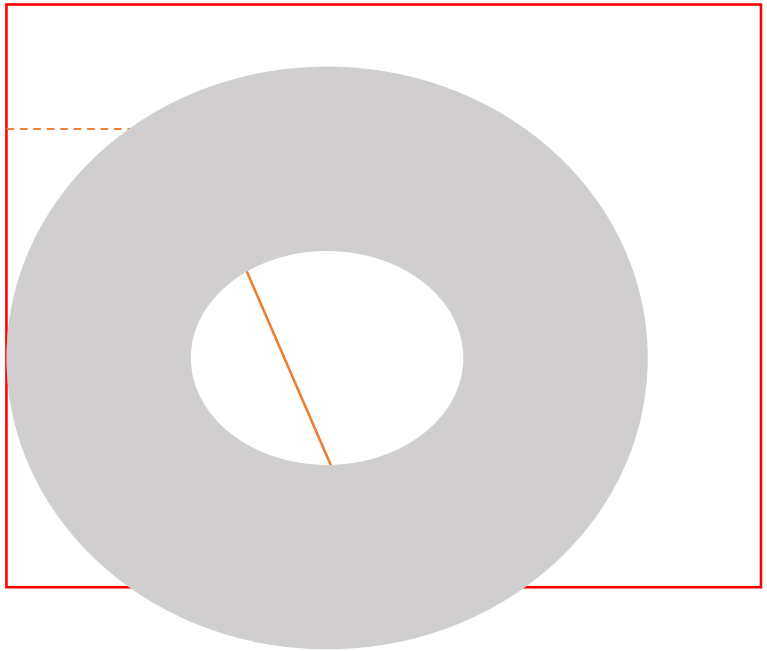
- Let's say this is object moves in this direction

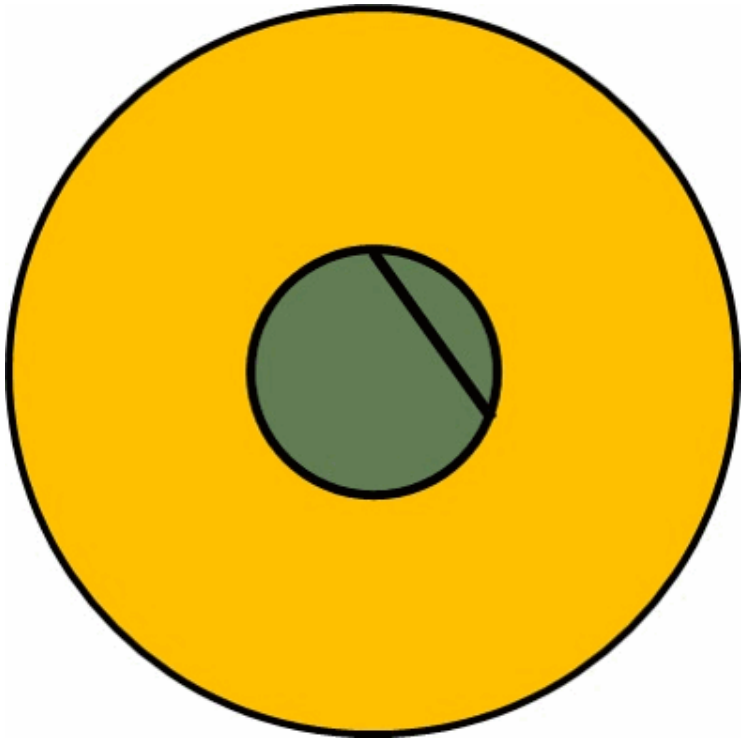




Aperture problem

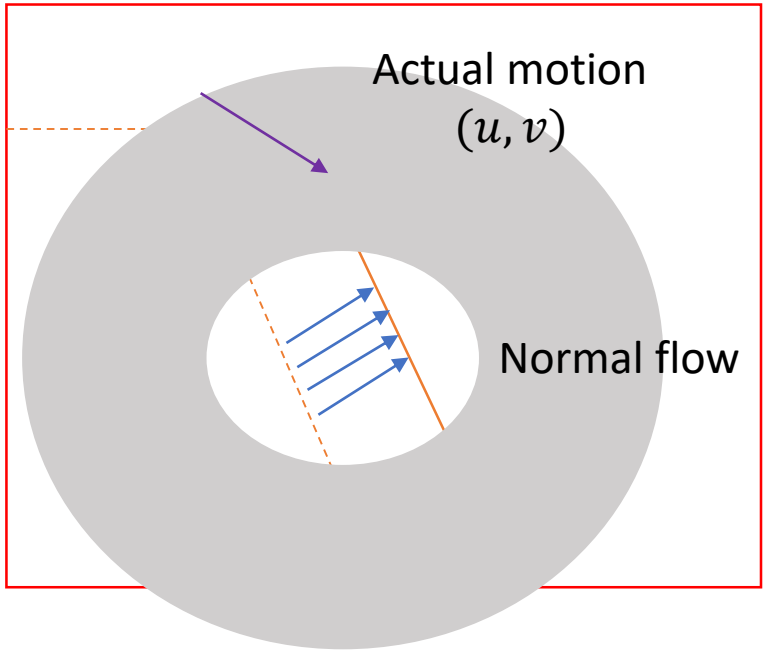
- Let us put an aperture over the line. If we can observe only the green area we will get a wrong impression that the line is moving in the direction perpendicular to the edge.

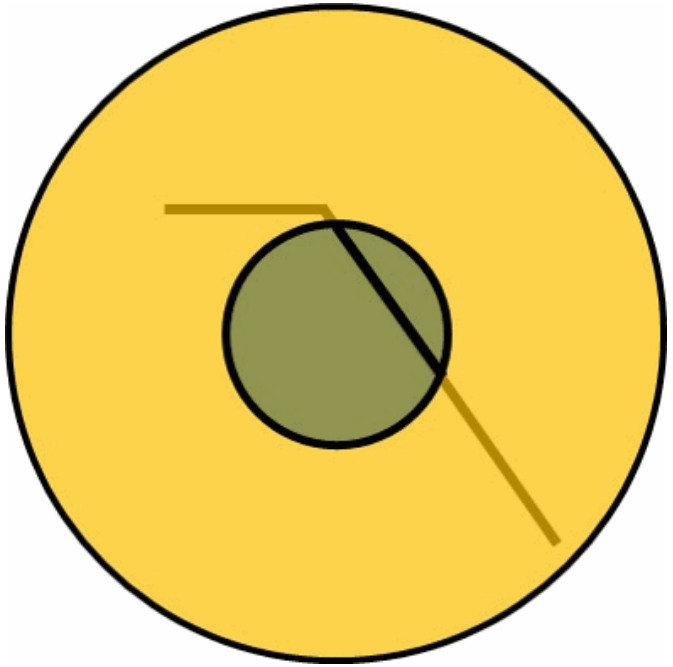




Aperture problem

- We are not able to measure the actual flow.
- We can only able to determine the normal flow.





[Aperature problem Demo](https://datahacker.rs/calculating-sparse-optical-flow-using-lucas-kanade-method/)
https://datahacker.rs/calculating-sparse-optical-flow-using-lucas-kanade-method/
https://elvers.us/perception/aperture/

Optical Flow is under constrained

- Constraint equation

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

- 2 unknowns 1 equation
- How are we going to solve this?
  - We need additional constraints.

Lukas Kanade Method

Lukas Kanade Solution

- **Lukas Kanade Assumption**: For each pixel, assume Motion Field, and hence Optical Flow (u, v), is constant within a small neighborhood w.

W

Small window is better. They produce the same motion field and hence the same optical flow in the image

$$That\ is\ for\ all\ points\ in\ the\ window\ (k,l) \in W:$$

$$I_x(k,l)u + I_y(k,l)v + I_t(k,l) = 0$$

[Lucas 1981]

Lucas-Kanade Solution

$For\ all\ points\ (k,l) \in W:\ I_x(k,l)u + I_y(k,l)v + I_t(k,l) = 0$

Let the size of window w be $n \times n$. So we get a large system of equations which is equal to the number of point inside the window.

In matrix form:

$$\begin{bmatrix} I_x(1,1) & I_y(1,1) \\ I_x(k,l) & I_y(k,l) \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ I_x(n.n) & I_y(n,n) \end{bmatrix} \quad \begin{bmatrix} u \\ v \end{bmatrix} \quad = \begin{bmatrix} I_t(1,1) \\ I_t(1,1) \\ \cdot \\ \cdot \\ \cdot \\ I_n(n,n) \end{bmatrix}$$

A (known) $n^2 \times 2$ | u (Unknown) 2×1 | B (known) $n^2 \times 1$

We have, $n^2\ Equations,\ 2\ Unknowns$:
So we will solve this problem using the Least Squares Solution

Least Squares Solution

$Solve\ linear\ System$: $Au = B$

$$A^T Au = A^T B$$ (Least-Squares using Pseudo-Invers)

In matrix form:

$$\begin{bmatrix} \sum_W I_x I_x & \sum_W I_x I_y \\ \sum_W I_x I_y & \sum_W I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum_W I_x I_t \\ -\sum_W I_y I_t \end{bmatrix}$$

$(A^T A)^{-1}$ (known) 2×2

u (unknown) 2×2

$A^T B$ (Known) 2×1

Indices (k, l) not written for simplicity

$$u = (A^T A)^{-1} A^T B$$

Fast and East to Solve

Given two consecutive frames compute the motion vector(OF) for each pixel in the image using LK method

1) Compute Gradients: Calculate spatial gradients $I_x \, and \, I_y (\; x \; and \; y \; derivatives)$ and temporal gradient I_t (frame difference).

2) Optical Flow Equation: For each pixel, set up the equation: $I_x u + I_y v = -I_t$ where u and v are the motion vectors in the $I_x \, and \, I_y$ directions, respectively.

3) Local Window System: For each pixel, gather the equations from its local window (e.g., 5×5) and form the system:

$$\mathrm{A} \begin{bmatrix} u \\ v \end{bmatrix} = b \; ,$$

where A contains the spatial gradients $I_x \, and \, I_y$ for all pixels in the window, and b contains the negative temporal gradients $-I_t$.

4) Solve for Motion Vectors: Use least squares to solve for $u \, and \, v$:

$$\begin{bmatrix} u \\ v \end{bmatrix} = (A^T A)^{-1} A^T b$$

5) Iterate: Repeat for all pixels to compute the flow vectors.

When Does Optical Flow Estimation Work?

$Au = B$ $\quad\quad$ $A^T Au = A^T B$

- $A^T A$ must be invertible. That is $\det(A^T)A \neq 0$
- $A^T A$ must be well-conditioned. What does it mean?

- Is where you have a significant enough change in the output for a change in the input.
- You can take the output and estimate the input well.
- If $A^T A$ is well-condition, then you will get robust, reliable solution for optical flow

When is $A^T A$ well-conditioned? Assume that this is two by two matrix.

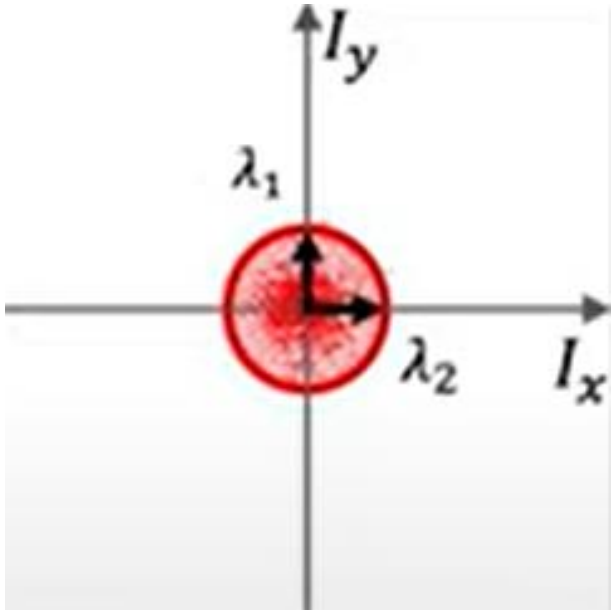
If λ_1 and λ_2 are eigen values of $A^T A$, then

$\lambda_1 > \in$ and $\lambda_2 > \in$ $\quad\quad$ Neither of one is close to zero. Should be significant enough.

$\lambda_1 \geq \lambda_2 \, but \, not \, \lambda_1 \ggg \lambda_2$ $\quad\quad$ One can be larger than the other one but not significantly larger than the other one

Smooth Regions (Bad)

- Equations for all pixels in window are more or less the same
- The spatial gradients are close to zero. Because there's not much texture in this window.
- Cannot reliably compute flow!



Spatial gradient $I_{x\ and}\ I_y$ plot

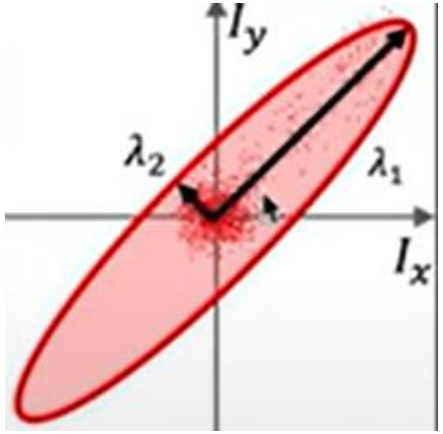
$\lambda_1 \sim \lambda_2$, Both are small

Smooth Regions (Bad)

- This should not surprise you when you have no texture to hang on to.
- If the derivatives are all small estimating optical flow is very difficult.
- And that applies to human as well.
  - If I you see scene or a surface patch which has no texture moving it looks pretty much the same even after the motion.

Edges (Bad)

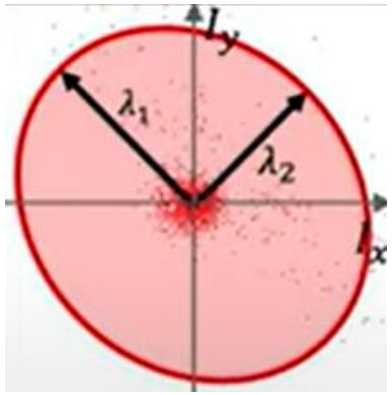
- Badly conditioned. Prominent gradient in one direction
- Cannot reliably compute flow!
- Same as Aperture problem



$\lambda_1 \ggg \lambda_2$

Textured Region (Good)

- Well conditioned. Large and diverse gradient magnitudes
- Can reliably compute optical flow.



$\lambda_1 \sim \lambda_2$
Both are large

Coarse to Fine Flow Estimation

What if we have large Motion?

- When we derived the optical flow constrained equation, we made the assumption that the displacement is $(\delta x, \delta y)$ are really small.
- So what happens if you have large motion between consecutive images?

What if we have large Motion?

- So consider you have two images taken in quick succession.
- Let's assume that in this case, the camera's moving.
- And that's the cause of the motion. And because the tree is close to the camera, its motion is going to be substantial.
- As per perspective, projection, maybe by tens of pixels.
- Optical flow constrained equation no longer valid!

What if we have large Motion?

Taylor Series approximation of

$I(x + \delta x, y + \delta y, t + \delta t)$ is not valid

Our simple linear constraint equation not valid

$I_x u + I_y v + I_t \neq 0$

Large Motion: Coarse-to-Fine Estimation

t

$t + \delta t$

Resolution $N \times N$

Resolution $\frac{N}{2} \times \frac{N}{2}$

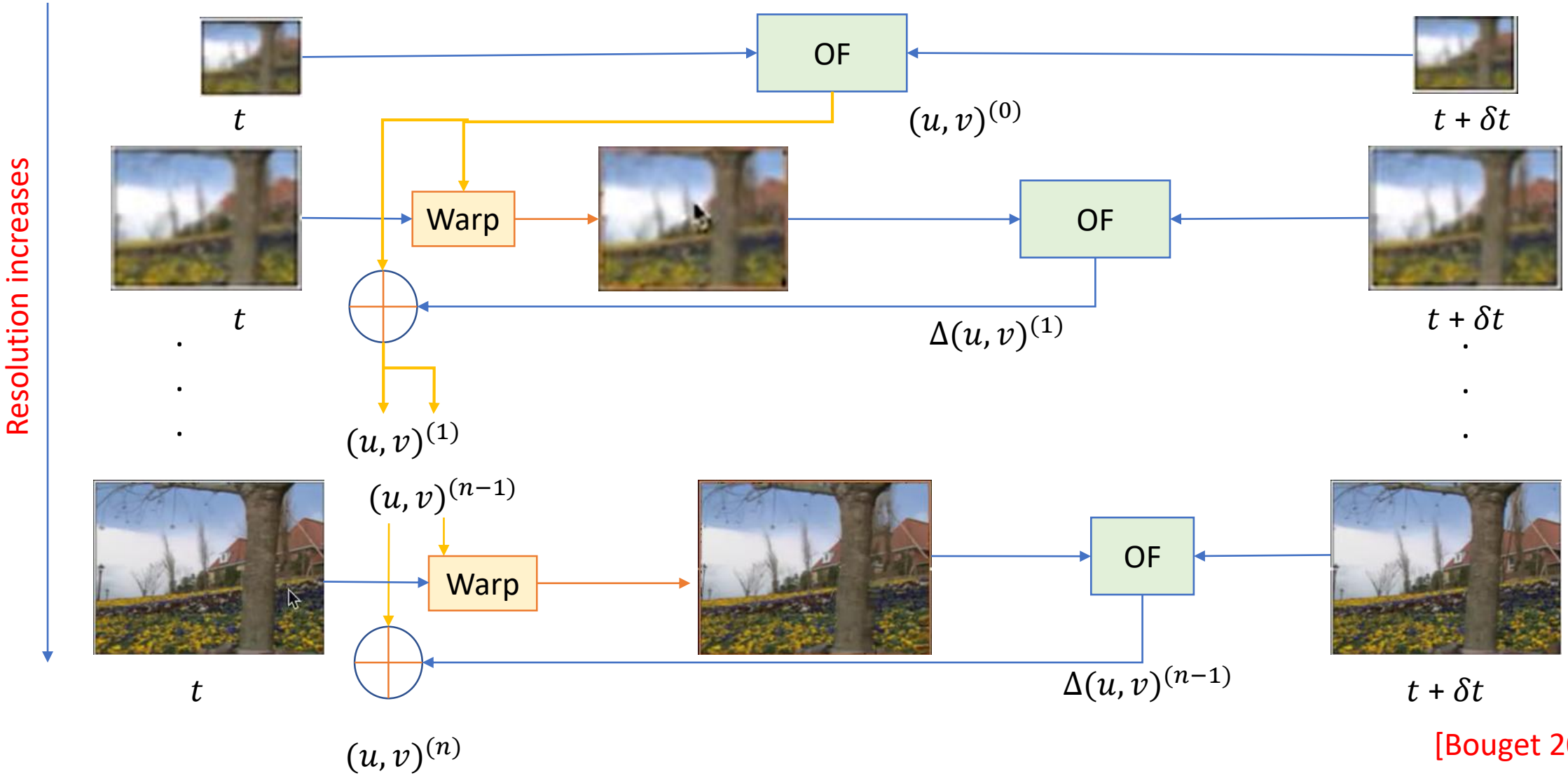
Resolution $\frac{N}{4} \times \frac{N}{4}$

Resolution $\frac{N}{8} \times \frac{N}{8}$

- Resolution pyramid: Simple way to lower the resolution to take each two by two window in one of these images and then find the average of those values and use that value in this new low resolution image.

- At lowest resolution, motion ≤ 1 pixel.
- At the lower resolution, the optical flow constrain equation becomes valid again

Coarse-to-Fine Estimation Algorithm



[Bouget 2000]

Coarse-to-Fine Estimation Algorithm

1) Build a Pyramid (e.g., 4 levels: Level 0 = original, Level 3 = coarsest).
   - Downsample images by blurring + subsampling (typically by a factor of 2 per level).
2) Compute Initial Flow at Coarsest Level
   - Estimate flow (u_3, v_3) between I_1 and I_2 at the lowest resolution (smallest images).
3) Iteratively Refine Flow Up the Pyramid
   - For each level l (from coarsest to finest):
     - Upsample Flow from **level l+1** to level **l**: (Upsampling doubles flow vectors to match the next level's resolution.)

$$u_l^{up} = 2 \cdot \text{resize}(u_{l+1}), \quad v_l^{up} = 2 \cdot \text{resize}(v_{l+1})$$

     - Warp I_2 using the upsampled flow: $I_2^w(x,y) = I_2(x + u_l^{up}(x,y), y + v_l^{up}(x,y))$

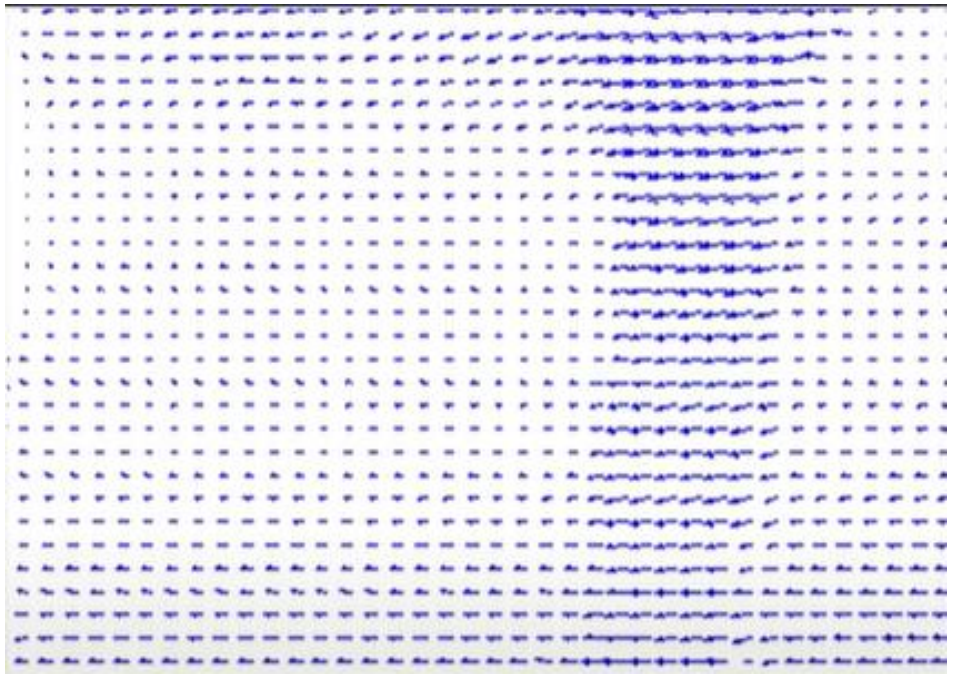
     (This aligns I_2 closer to I_1 at the current resolution.)
     - Compute Residual Flow (du, dv) between I_1 and I_2^w at **level l**:
     - Update Flow Estimate:

$$u_l = u_l^{up} + du, \quad v_l = v_l^{up} + dv$$

     (Combines upsampled flow + residual corrections.)
4) Repeat Until Level 0 (Original Resolution)
   - After refining at the finest level, (u_0, v_0) is the final optical flow for the full-resolution images.

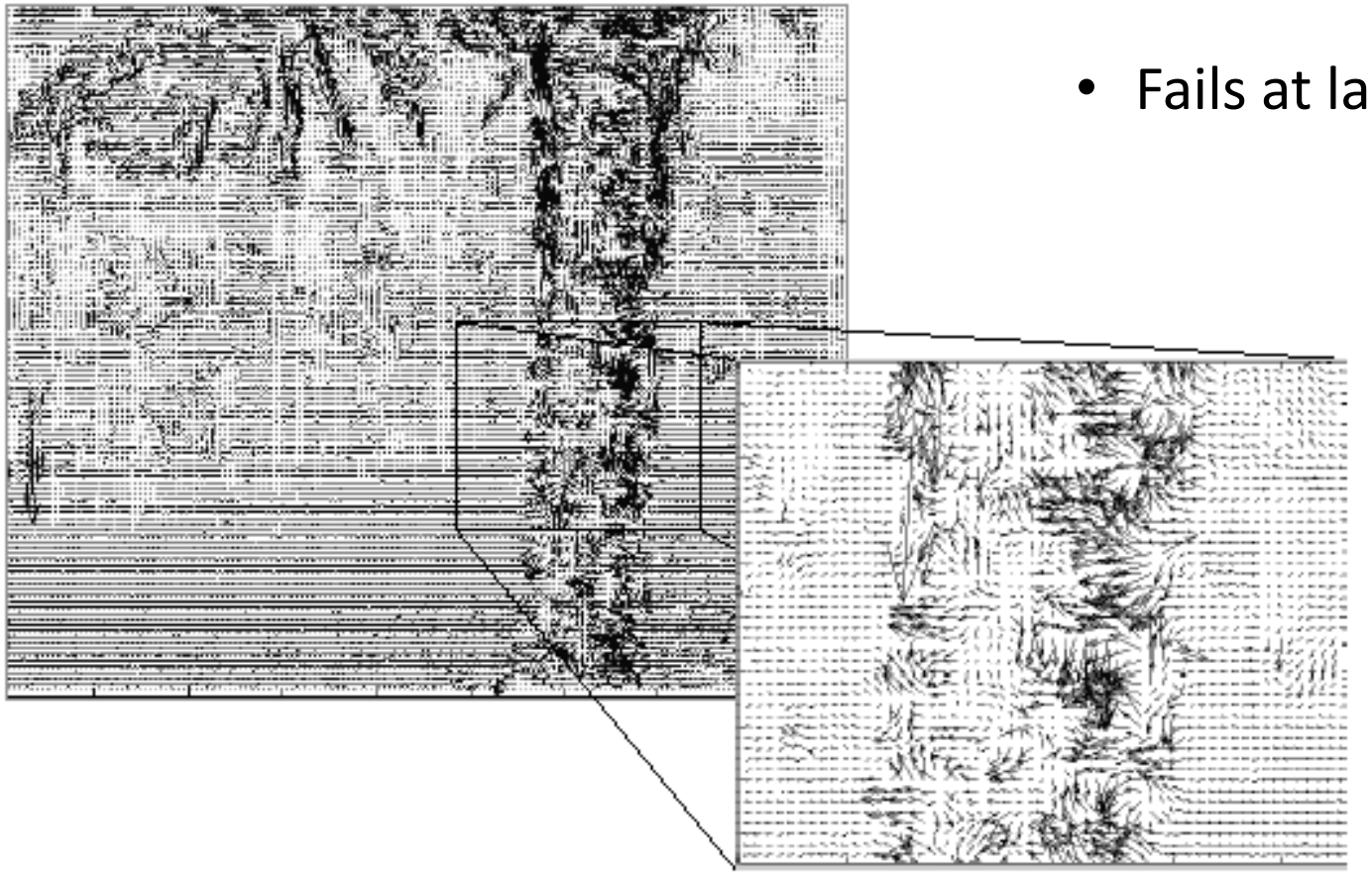
Results: Tree Sequence

Image Sequence



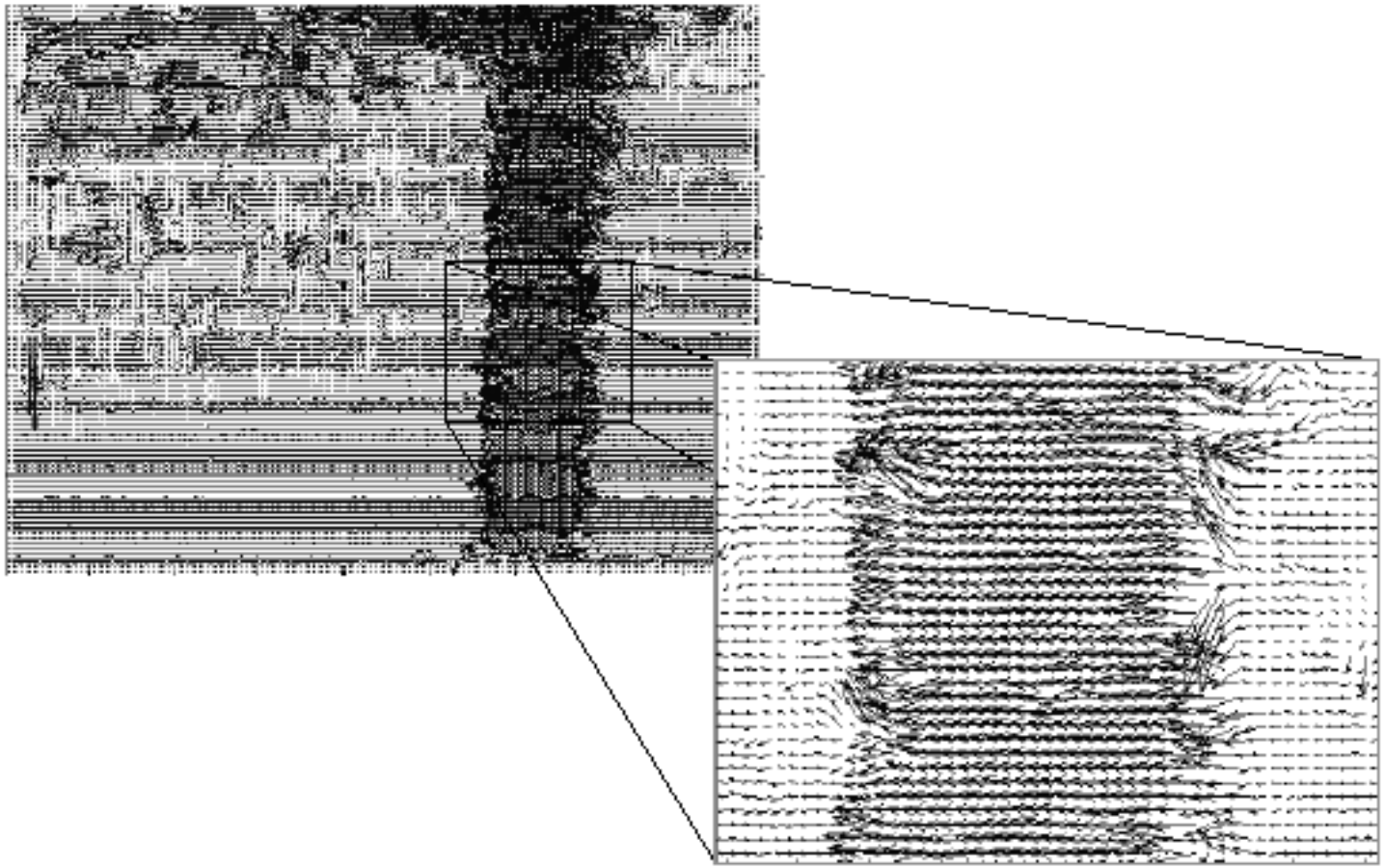
Optical Flow

Lucas-Kanade without Pyramids



- Fails at large motions.

Lucas-Kanade with Pyramids



Alternative Approach: Template Matching

- Brute force approach to compute OF.
- Determine OF using Template Matching

Template Window T

Search Window S

Image I_1 at time t

Image I_2 at time t t + δt

- For each template window T in Image I_1, find the corresponding match in I_2 using search window
- The difference between the location of the corresponding point is optical flow

Alternative Approach: Template Matching

- Brute force approach to compute OF.
- Determine OF using Template Matching

Template Window T

Image I_1 at time t

Search Window S

Image I_2 at time t t + δt

- Template matching is slow when search window S is large.
- Also, mismatches are possible,

Dense and Sparse Optical Flow

Dense and Sparse Optical Flow

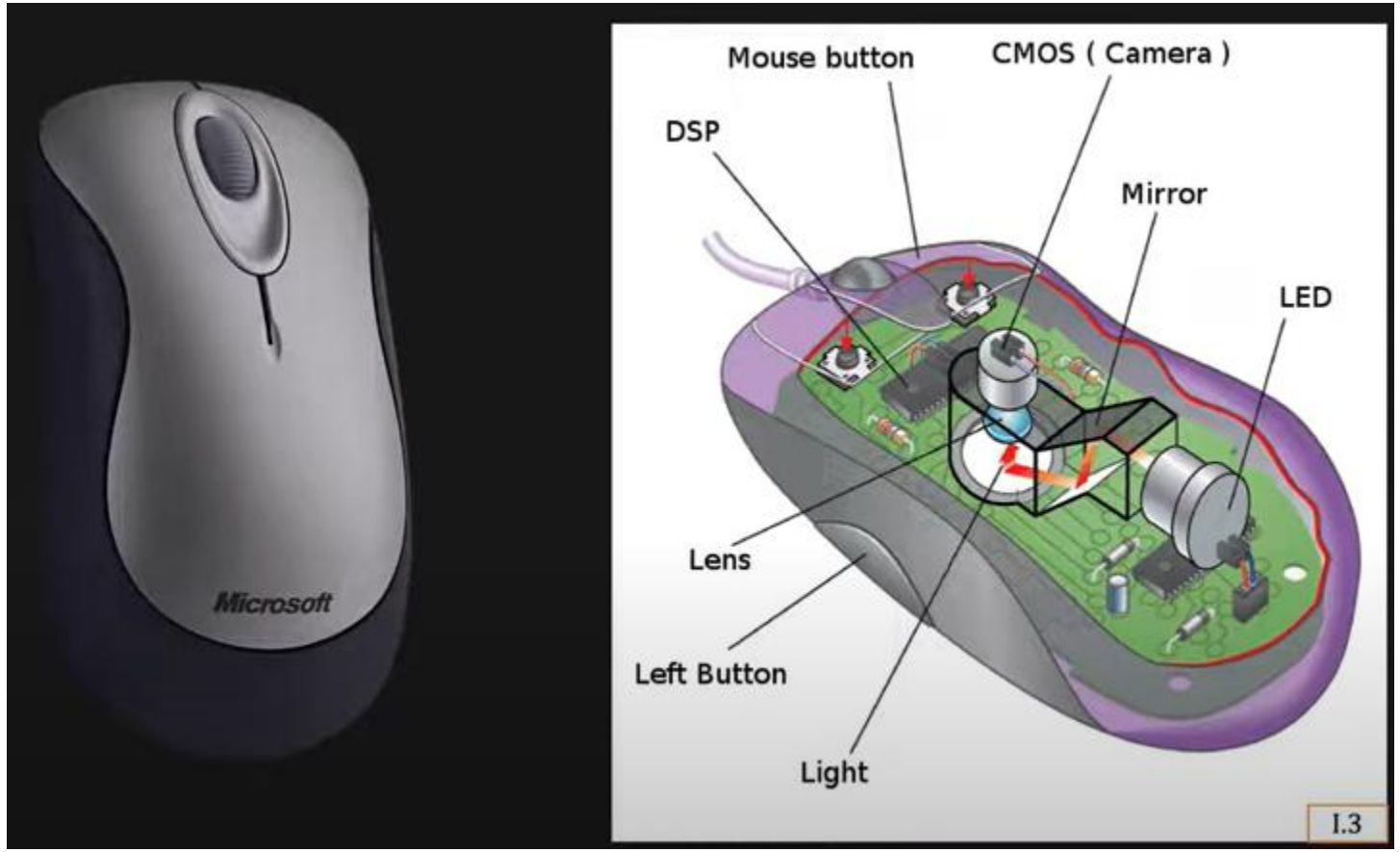
- Dense optical flow
  - Compute estimate for each pixel.
  - Higher accuracy at the cost of slow/expense computation.
- Sparse optical flow
  - Compute estimate for some interesting feature points (given by corners or SIFT).
  - Much less computation cost.

Sparse

Dense

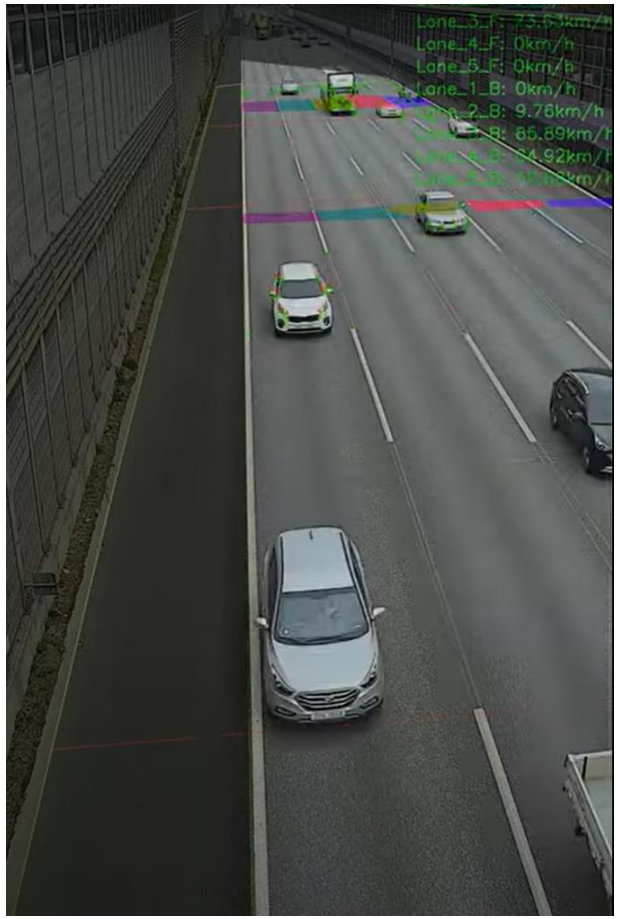
Application of Optical Flow

Optical Mouse



Estimating Mouse Movements

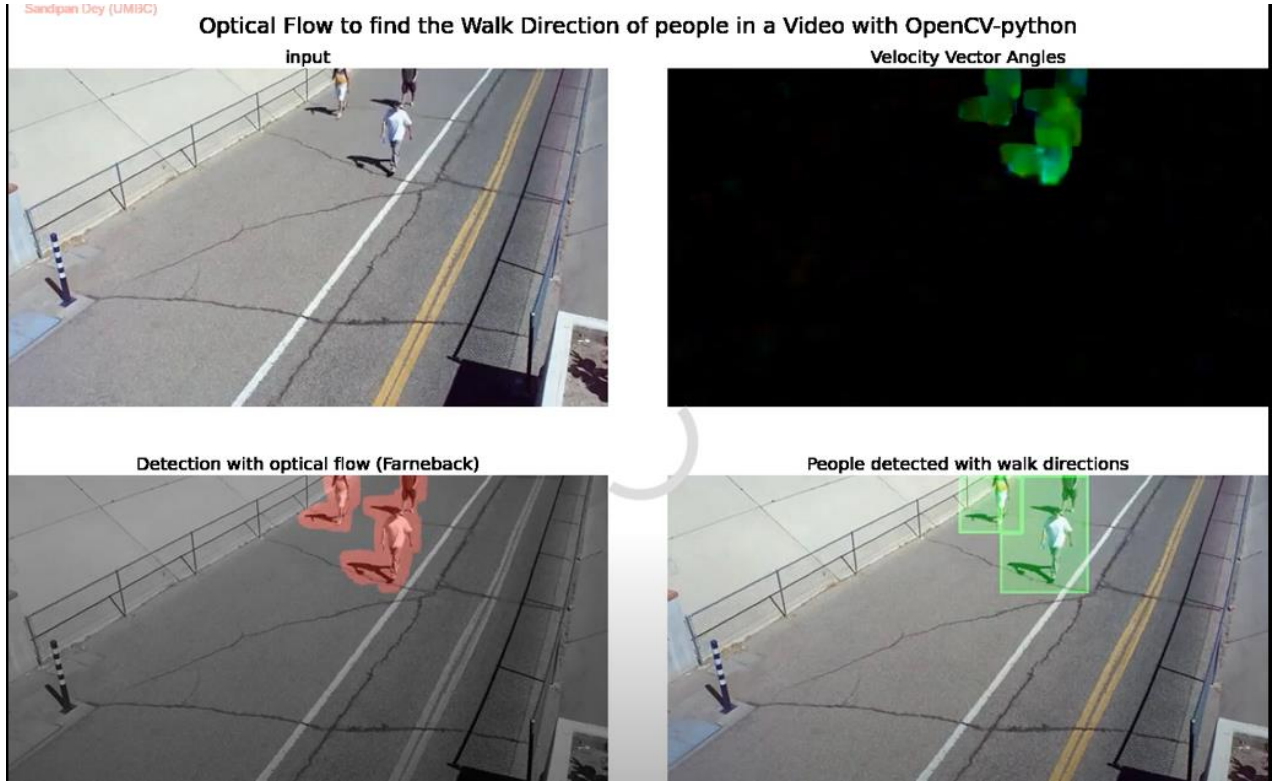
Traffic Monitoring



Finding Velocities of Vehicles from flow vector

https://www.youtube.com/watch?v=1-TnogKwtDM

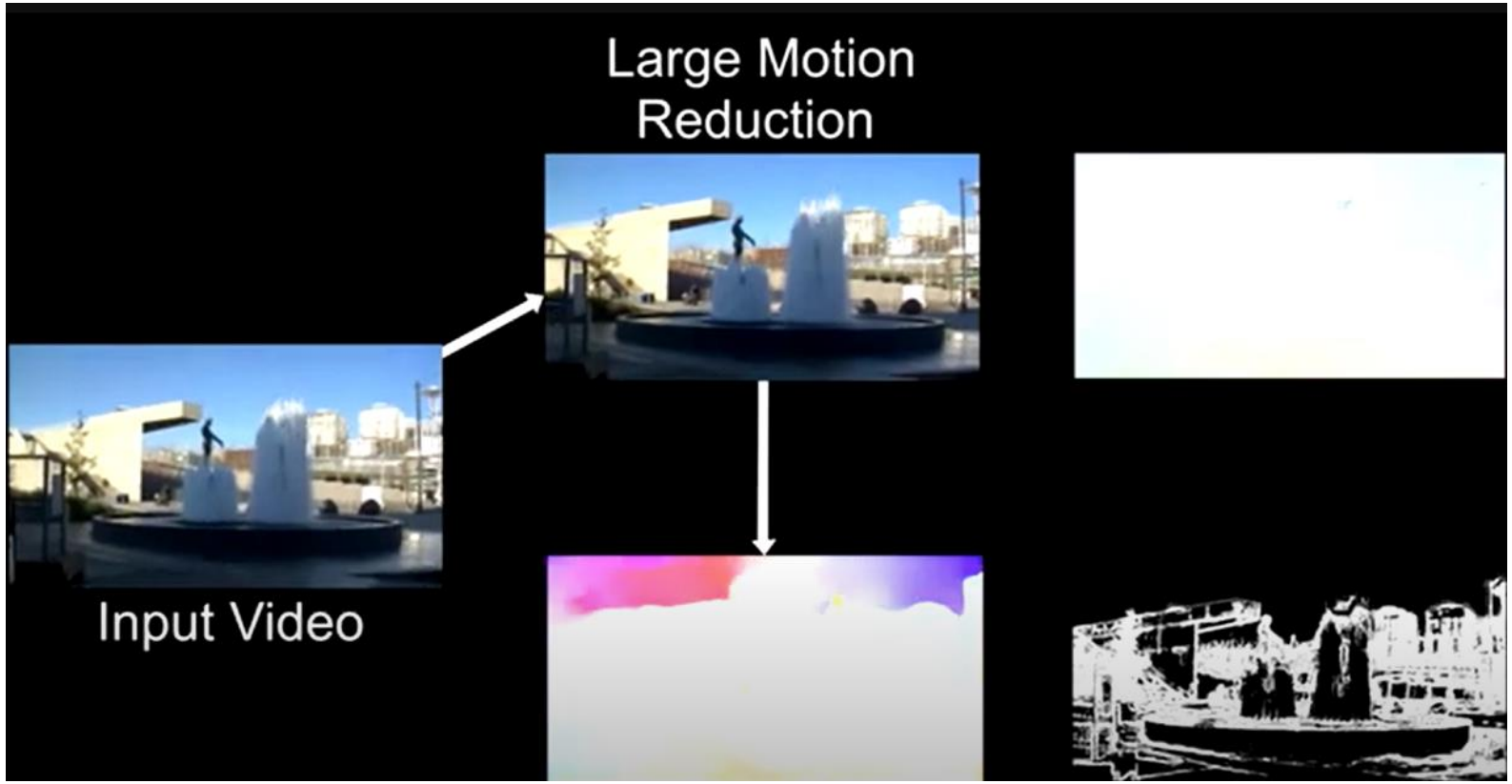
Finding direction of moving object



Finding direction of people walking in video

https://www.youtube.com/watch?v=e_TeY6QRp4c

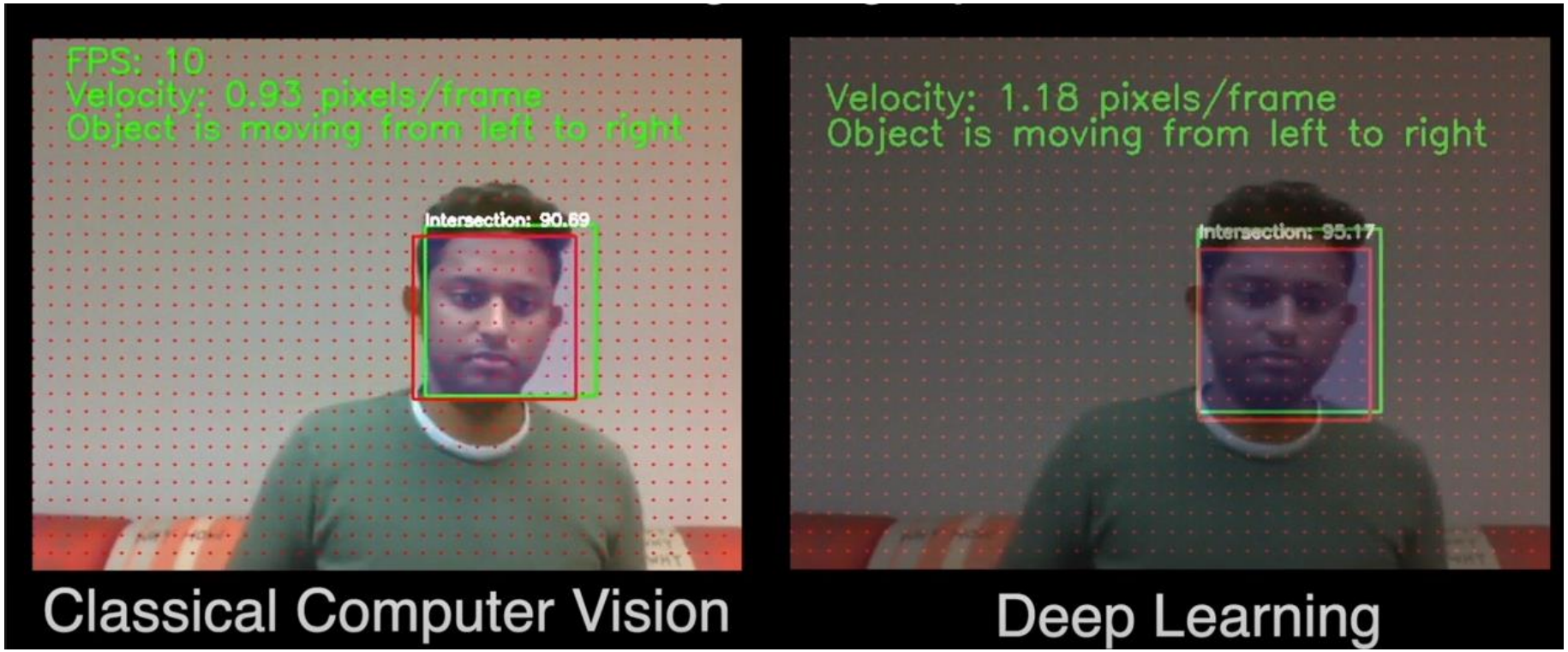
Image/Video Stabilization



Optical Flow used to remove camera shake.

https://www.youtube.com/watch?v=m4SKRRjMxPE

Tracking



Face Tracking

https://www.youtube.com/watch?v=YQDdv9CqYyA

Recap

- Motion Field and Optical Flow:
- Optical Flow Constraint Equation
  - Aperture problem
- Lukas Kanade Method
- What if we have large Motion
  - Coarse to Fine Flow Estimation
- Dense and Sparse Optical Flow
- Application of Optical Flow

References

- Ch 8 (Szeliski)
- Lucas and Kanade, "An iterative image registration technique with an application to stereo vision", 1981

Next: Intro to Object Recognition