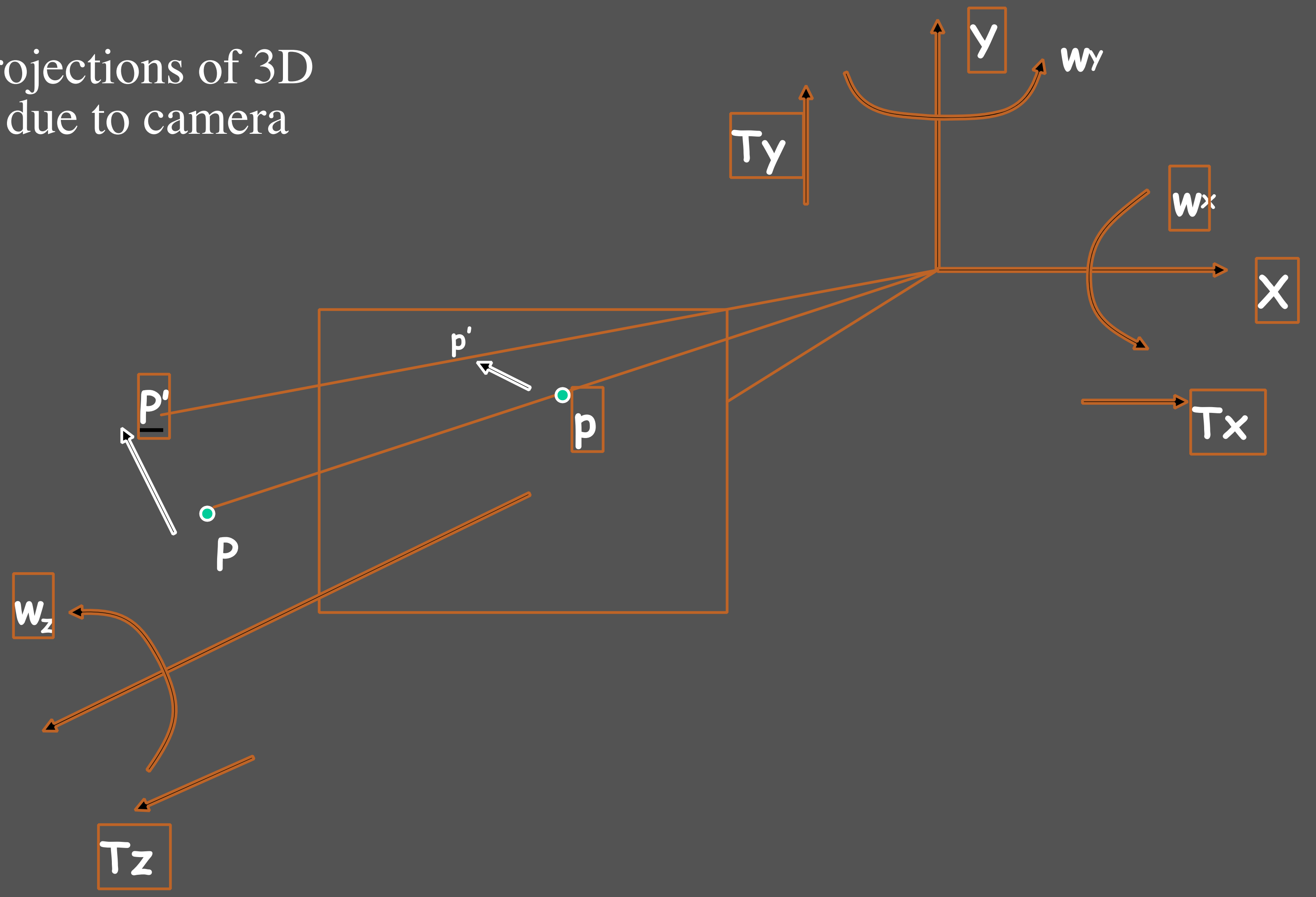
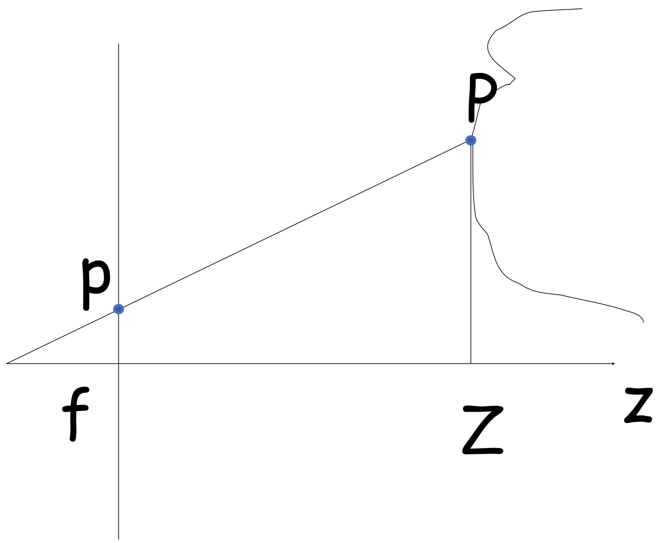
Visual Motion & Motion Field & Optical flow estimation

Motion Field

Motion Field : 2D projections of 3D displacement vectors due to camera and/or object motion



Consider a 3D point P and its image



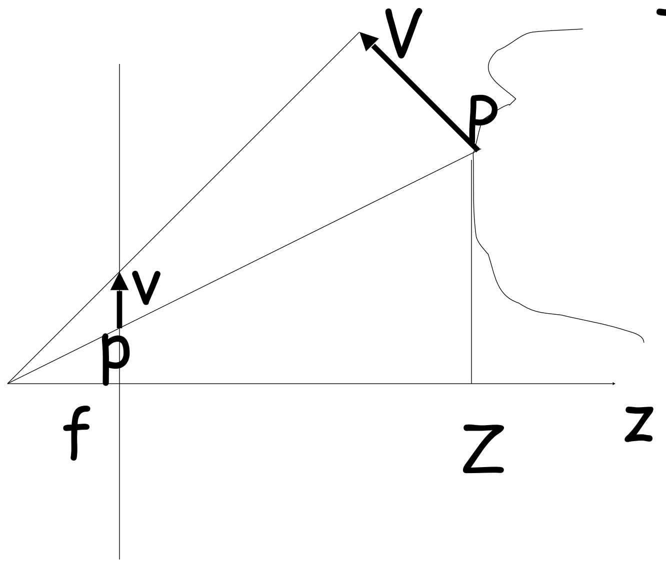
$$P = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$p = \begin{bmatrix} x \\ y \\ f \end{bmatrix}$$

Using pinhole camera equation:

$$p = \frac{fP}{Z}$$

Relative motion



The relative velocity of P wrt camera:

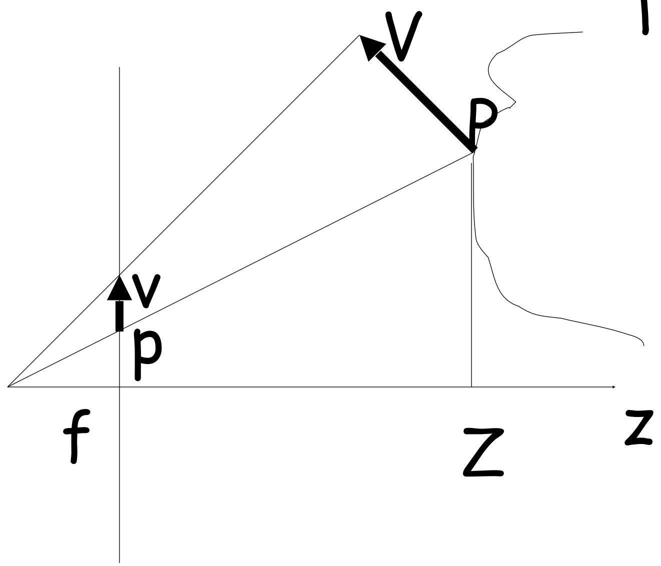
$$V = -t - \omega \times P$$

Translation velocity

Rotation angular velocity

$$t = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \qquad \omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

3D Relative Velocity



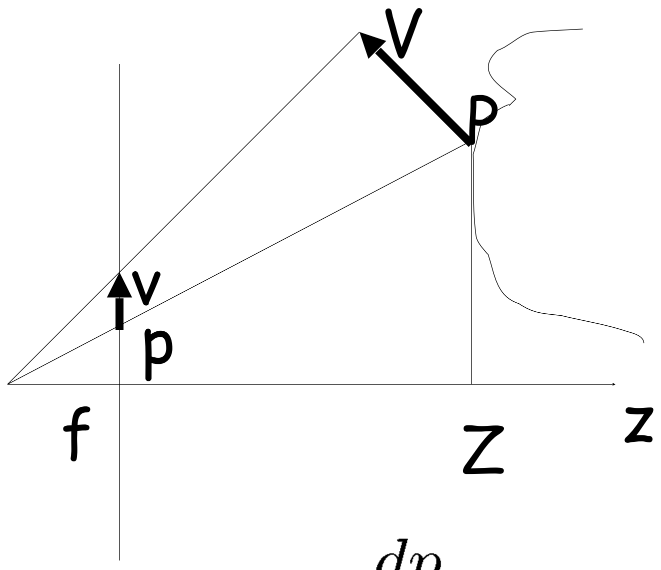
The relative velocity of P wrt camera:

$$V = -t - \omega \times P$$

$$t = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad \omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad P = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$\begin{aligned} V_x &= -t_x - \omega_y Z + \omega_z Y \\ V_y &= -t_y - \omega_z X + \omega_x Z \\ V_z &= -t_z - \omega_x Y + \omega_y X \end{aligned}$$

Motion Field



$$p = \frac{fP}{Z}$$

Taking derivative wrt time:

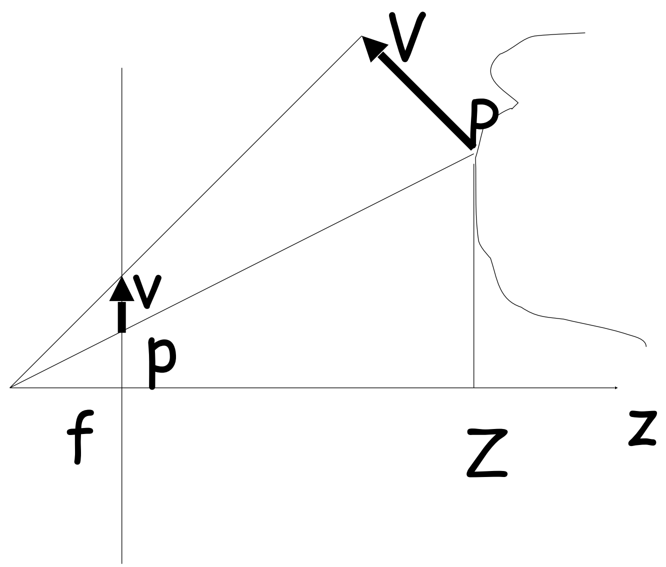
$$\frac{dp}{dt} = \mathbf{v} = \frac{d\frac{fP}{Z}}{dt} \qquad = \frac{f}{Z^2}\left[\frac{dP}{dt}.Z - P.\frac{dZ}{dt}\right]$$

$$= \frac{f}{Z^2}\left[V.Z - P.V_z\right]$$

the velocity of p

$$\boxed{\mathbf{v} = f\frac{V}{Z} - p\frac{V_z}{Z}}$$

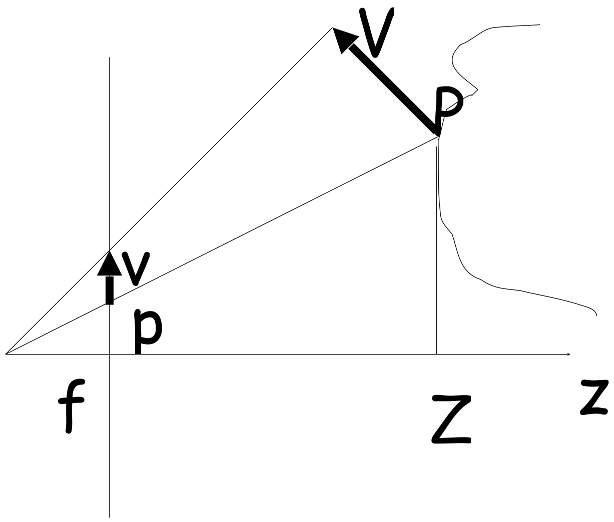
Motion Field



$$\boxed{v = f\frac{V}{Z} - p\frac{V_z}{Z}}$$

$$
\begin{aligned}
v_x = u &= f\frac{V_x}{Z} - x\frac{V_z}{Z} \\
v_y = v &= f\frac{V_y}{Z} - y\frac{V_z}{Z} \\
v_z &= f\frac{V_z}{Z} - f\frac{V_z}{Z} = 0
\end{aligned}
$$

Motion Field

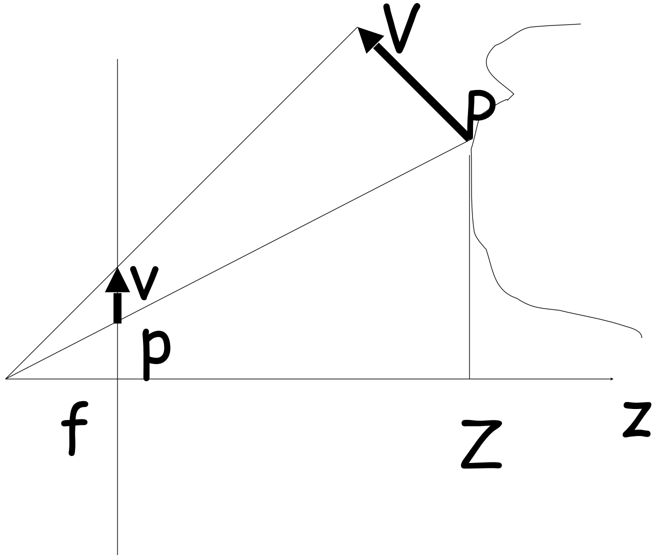


$$
\begin{aligned}
u &= f\frac{V_x}{Z} - x\frac{V_z}{Z} \\
v &= f\frac{V_y}{Z} - y\frac{V_z}{Z}
\end{aligned}
$$

$$
\begin{aligned}
V_x &= -T_x - \omega_y Z + \omega_z Y \\
V_y &= -T_y - \omega_z X + \omega_x Z \\
V_z &= -T_z - \omega_x Y + \omega_y X
\end{aligned}
$$

$$
\begin{aligned}
u &= \frac{t_z x - t_x f}{Z} + \frac{\omega_x x y}{f} - \omega_y\left(f + \frac{x^2}{f}\right) + \omega_z y \\
v &= \frac{t_z y - t_y f}{Z} + \omega_x\left(f + \frac{y^2}{f}\right) - \frac{\omega_y x y}{f} - \omega_z x
\end{aligned}
$$

Motion Field:

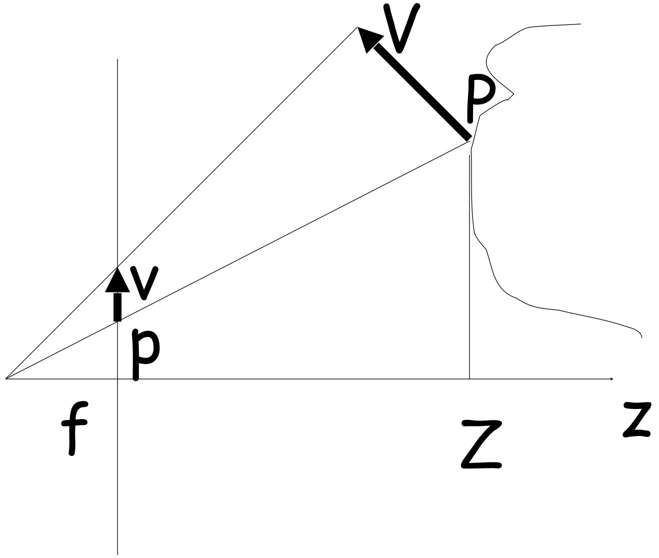


Translational component

Scaling ambiguity
(t and Z can only be derived up to a scale Factor)

$$
\begin{aligned}
u &= \frac{t_z x - t_x f}{Z} + \frac{\omega_x x y}{f} - \omega_y \left(f + \frac{x^2}{f}\right) + \omega_z y \\
v &= \frac{t_z y - t_y f}{Z} + \omega_x \left(f + \frac{y^2}{f}\right) - \frac{\omega_y x y}{f} - \omega_z x
\end{aligned}
$$

Motion Field:

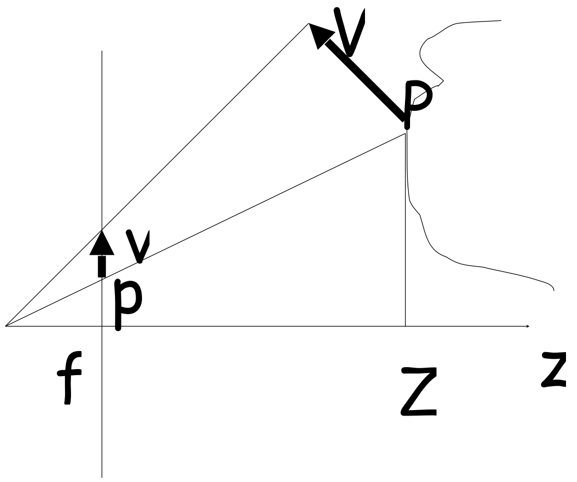


Rotational component

$$
\begin{aligned}
u &= \frac{t_z x - t_x f}{Z} + \frac{\omega_x x y}{f} - \omega_y\left(f + \frac{x^2}{f}\right) + \omega_z y \\
v &= \frac{t_z y - t_y f}{Z} + \omega_x\left(f + \frac{y^2}{f}\right) - \frac{\omega_y x y}{f} - \omega_z x
\end{aligned}
$$

NOTE: The rotational component is independent of depth Z !

Motion Field



$$\boxed{\mathbf{v}(x,y) = \frac{1}{Z(x,y)}\mathbf{A}(x,y)\mathbf{V} + \mathbf{B}(x,y)\omega}$$

$$\mathbf{v}(x,y) = \begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} \qquad \mathbf{A}(x,y) = \begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix} \qquad \mathbf{B}(x,y) = \begin{bmatrix} \frac{xy}{f} & -f - \frac{x^2}{f} & y \\ f + \frac{y^2}{f} & -\frac{xy}{f} & x \end{bmatrix}$$

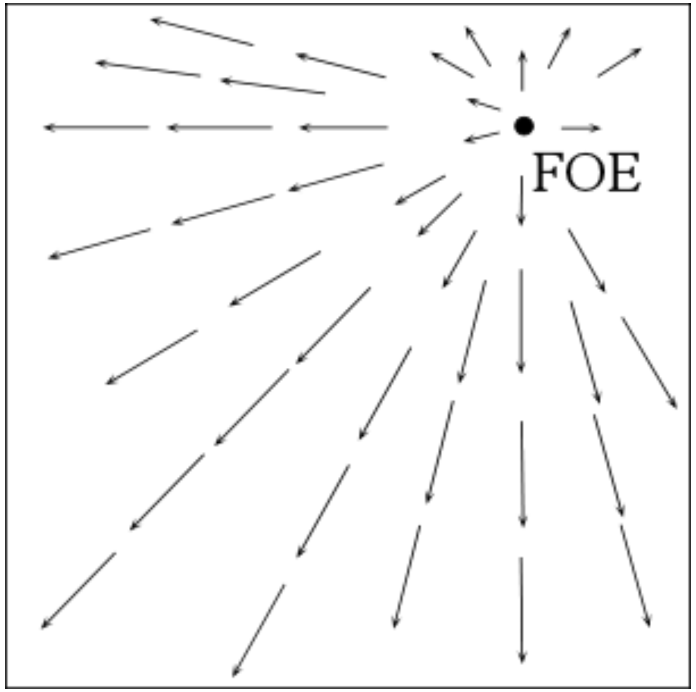
$$V = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \qquad \omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

Translational field

$$t_z \neq 0$$

$$u_{tr}(x,y) = (x - x_o)\frac{t_z}{Z}$$

$$v_{tr}(x,y) = (y - y_o)\frac{t_z}{Z}$$



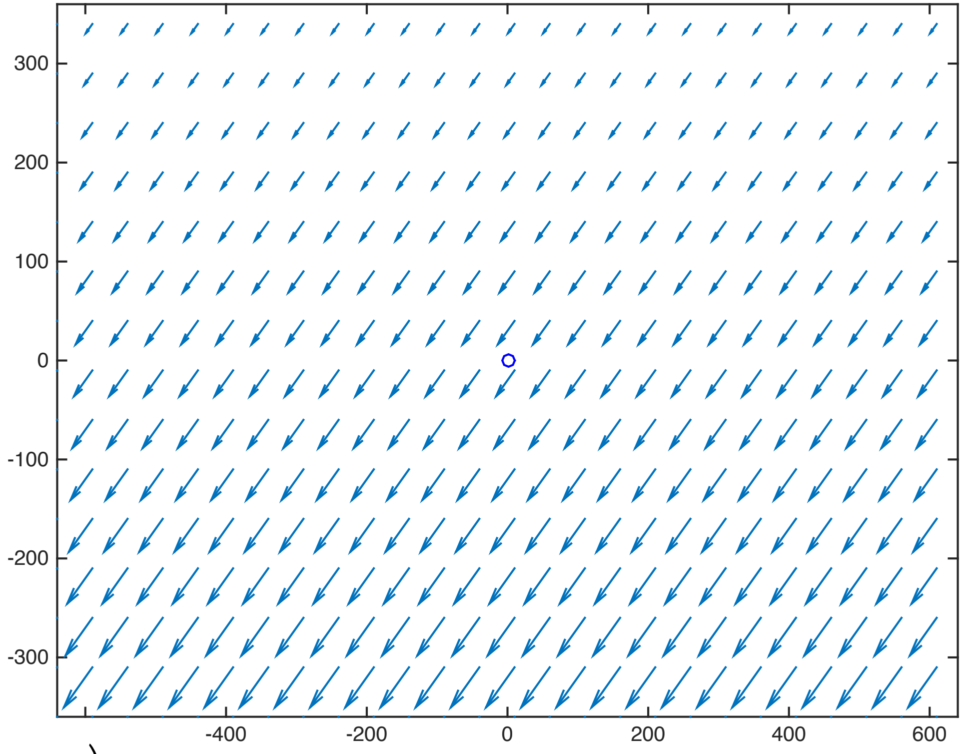
where $p_o = (x_0, y_0) = \left(\frac{t_x}{t_z} \cdot f, \frac{t_y}{t_z} \cdot f\right)$ is the focus of expansion (FOE) or focus of contraction (FOC).

Translational field

$$t_z = 0$$

$$u_{tr}(x,y) = -\frac{t_x \cdot f}{Z}$$

$$v_{tr}(x,y) = -\frac{t_y \cdot f}{Z}$$



where $p_o = (x_0, y_0) = (\infty, \infty)$ is the focus of expansion (FOE)

or focus of contraction (FOC).

All motion field vectors are parallel to each other and inversely proportional to depth !

Pure Translation: Properties of the MF

- *If $t_z \neq 0$ the MF is RADIAL with all vectors pointing towards (or away from) a single point p_o. If $t_z = 0$ the MF is PARALLEL.*

- *The length of the MF vectors is inversely proportional to depth Z. If $t_z \neq 0$ it is also directly proportional to the distance between p and p_o.*

$$u_{tr}(x,y) = (x - x_o)\frac{t_z}{Z}$$

$$v_{tr}(x,y) = (y - y_o)\frac{t_z}{Z}$$

Motion Analysis

$$\boxed{\mathbf{v}(x,y)=\frac{1}{Z(x,y)}\mathbf{A}(x,y)\mathbf{V}+\mathbf{B}(x,y)\omega}$$

V => {Z, V, w}

Foreground
Background

Temporal
Persistence

Scene
Geometry

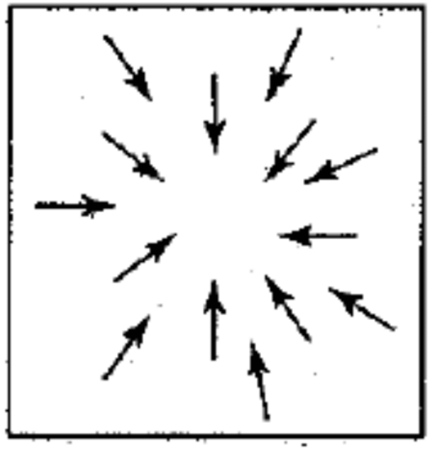
Layers & Mosaics

Segment, **Track**, Fingerprint
Moving Objects

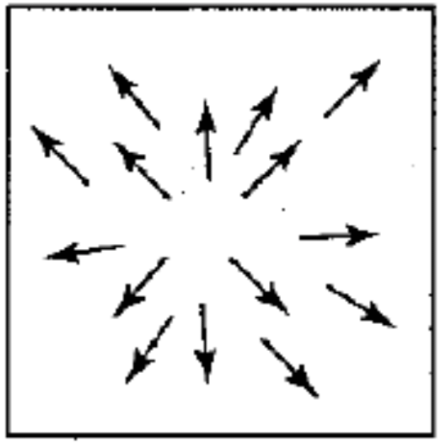
Layers with 2D/3D
Scene Models

Motion Analysis provides
Compact Representation for Manipulation & Recognition of Scene Content

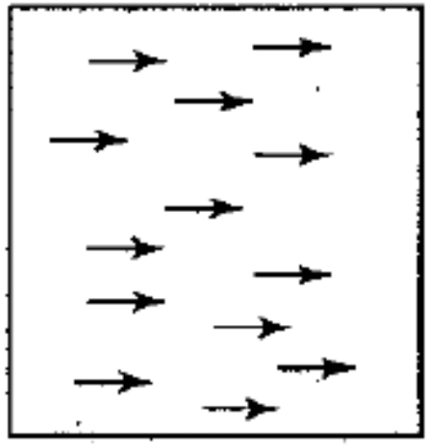
Typical Motion Fields



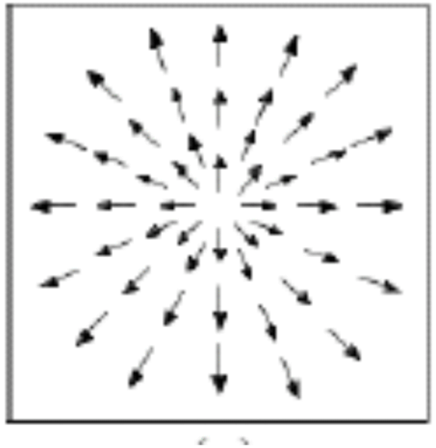
Zoom out



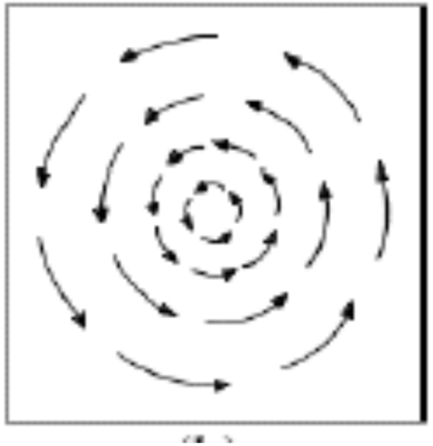
Zoom in



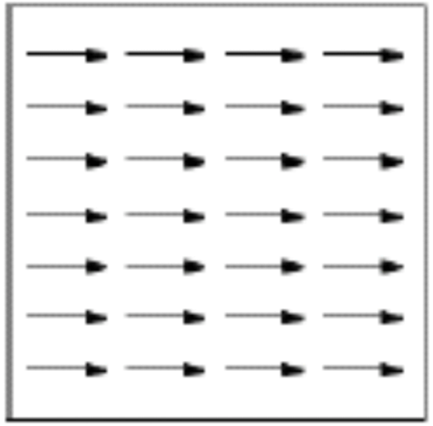
Pan right to left



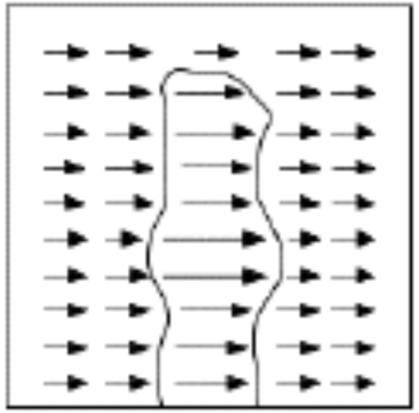
Forward motion



Rotation



Horizontal translation



Closer objects appear to move faster!!

Optical Flow Estimation

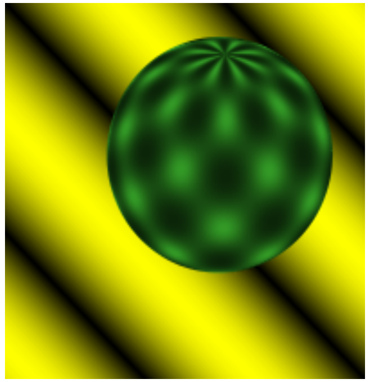
- Optical Flow
  - *Brightness constancy constraint*
  - *Aperture problem*
  - *Lucas-Kanade flow*
  - *Iterative refinement*
  - *Coarse-to-fine estimation*
- Global parametric motion
- Global Optical flow Constraint
- GFeature Tracking (sparse optical flow)

- **Definition-1**: **optical flow is the apparent motion of brightness patterns in the image**.
- **Ideally**, optical flow would be the same as the motion field.
- Have to be careful: apparent motion can be caused by lighting changes without any actual motion.
    - Think of a uniform rotating sphere under fixed lighting vs. a stationary sphere under moving illumination.

Optical Flow & Motion Field

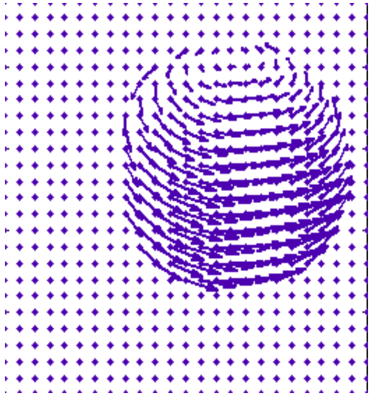
- **Definition-2**

The **optical flow** is a velocity field in the image which transforms one image into the next image in a sequence [Horn&Schunck]



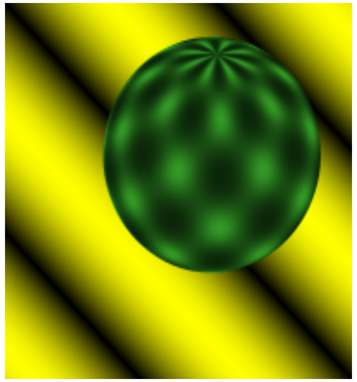
frame #1

+



flow field

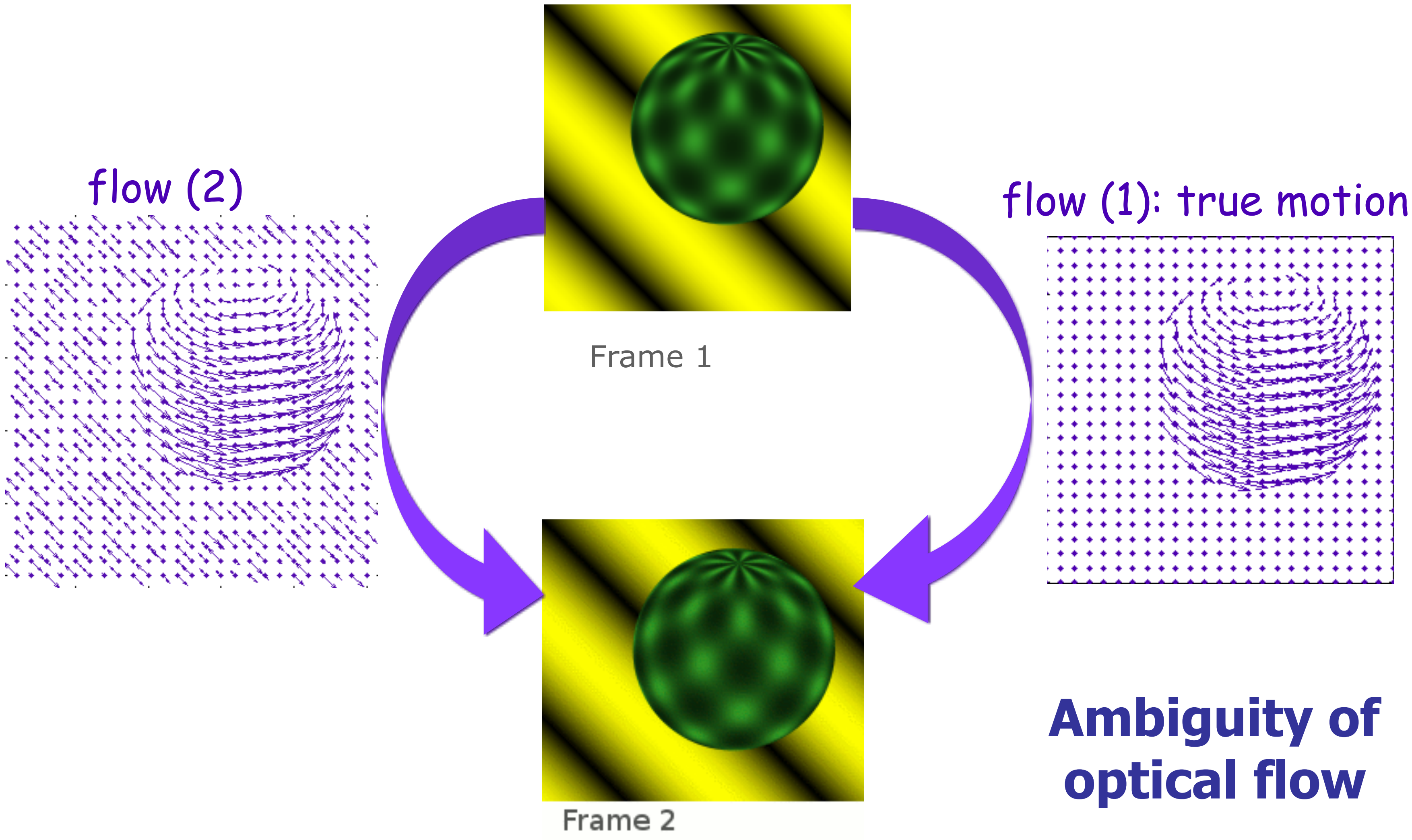
=



frame #2

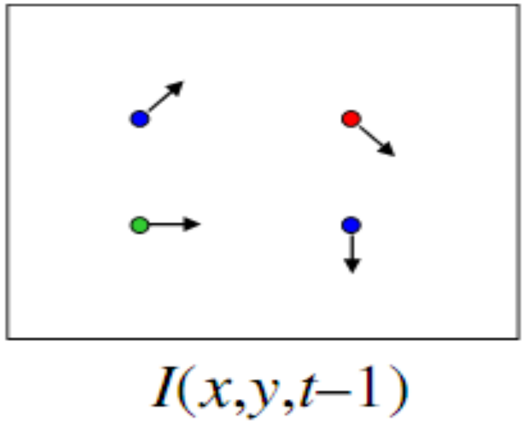
The **motion field** … is the projection into the image of three-dimensional motion vectors [Horn&Schunck]

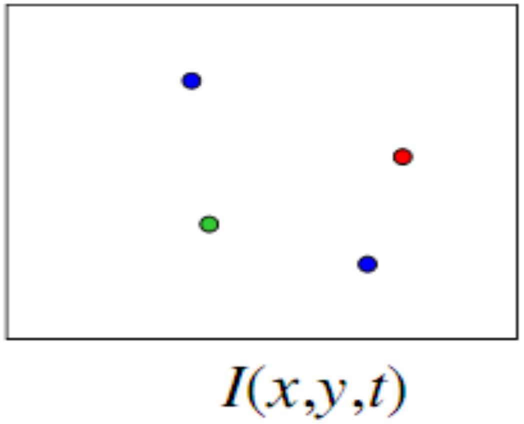
Optical Flow & Motion Field



Ambiguity of optical flow

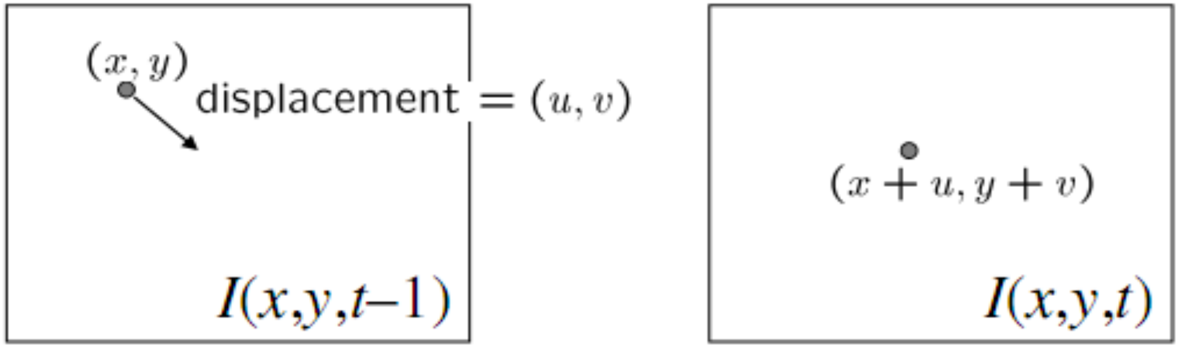
Estimating Optical Flow





- Given two subsequent frames, estimate the apparent motion field u(x,y) and v(x,y) between them.
- **Key assumptions**
  - Brightness constancy: projection of the same point looks the same in every frame. (**Local image constraints**)
  - Small motion: points do not move very far.
  - Spatial coherence: points move like their neighbors.

Local image constraints



- **Brightness Constancy Equation:**

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

- **Linearizing the right hand side using Taylor expansion:**

$$I(x, y, t-1) \approx I(x, y, t) + I_x \cdot u(x, y) + I_y \cdot v(x, y)$$

- **Hence,** $I_x \cdot u + I_y \cdot v + I_t \approx 0$

Spatial derivatives

Temporal derivative

Local image constraints

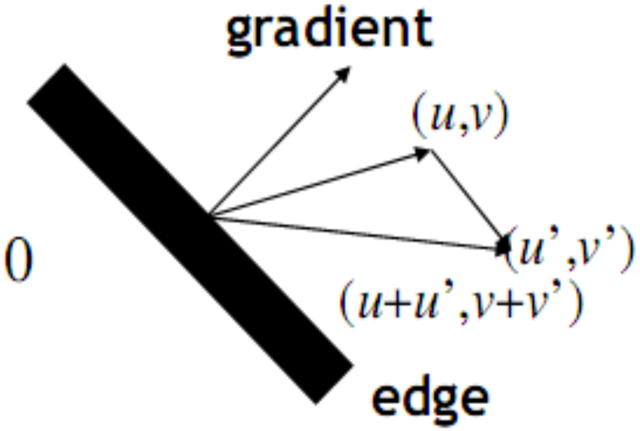
$$I_x \cdot u + I_y \cdot v + I_t = 0$$

- How many equations and unknowns per pixel?
  - One equation, two unknowns
- Intuitively, what does this constraint mean?

$$\nabla I \cdot (u, v) + I_t = 0$$

- The component of the flow perpendicular to the gradient (i.e., parallel to the edge) is unknown

If (u,v) satisfies the equation,
so does $(u+u', v+v')$ if $\nabla I \cdot (u', v') = 0$



Solving the Aperture Problem

- How to get more equations for a pixel?
- **Spatial coherence constraint**: pretend the pixel's neighbors have the same (u,v)
  - If we use a 5x5 window, that gives us 25 equations per pixel

$$0 = I_t(\mathbf{p_i}) + \nabla I(\mathbf{p_i}) \cdot [u \;\; v]$$

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

B. Lucas and T. Kanade. **An iterative image registration technique with an application to stereo vision**. In Proceedings of the International Joint Conference on Artificial Intelligence, pp. 674–679, 1981

- **Least squares problem:**

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

- **Minimum least squares solution given by solution of**

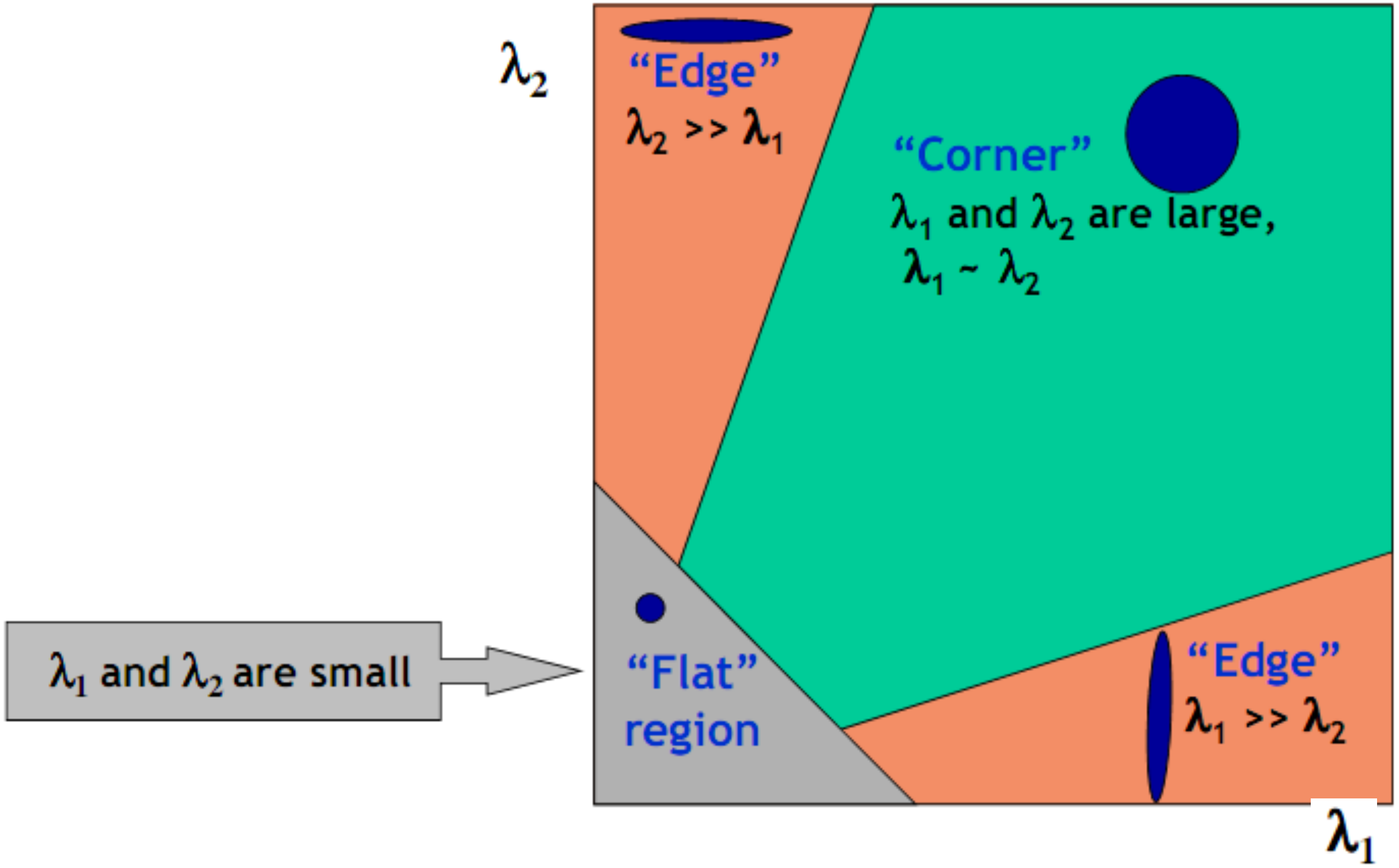
$$\underset{2\times 2}{(A^T A)} \; \underset{2\times 1}{d} = \underset{2\times 1}{A^T b}$$

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

$\boxed{\text{(The summations are over all pixels in the K x K window)}}$

Interpreting the Eigenvalues

- Classification of image points using eigenvalues of the second moment matrix:



Iterative Refinement

1. Estimate velocity at each pixel using one iteration of Lucas and Kanade estimation.

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

2. Warp one image toward the other using the estimated flow field.

3. Refine estimate by repeating the process.

B. Lucas and T. Kanade. **An iterative image registration technique with an application to stereo vision**. In Proceedings of the International Joint Conference on Artificial Intelligence, pp. 674–679, 1981

Lucas-Kanade flow

- Brightness constant equation (Optical equation)

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

- Spatial coherence constraint (local): pretend the pixel's neighbors (Ω) have the same (u,v)

$$E(u,v) = \sum_{(x,y)\in\Omega} w(x,y)\left(I_x(x,y)\cdot u + I_y(x,y)\cdot v + I(x,y,t) - I(x,y,t-1)\right)^2$$

$$= \sum_{(x,y)\in\Omega} w(x,y)\left(\begin{bmatrix} I_x & I_y & I_t \end{bmatrix}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}\right)^2 = \begin{bmatrix} u & v & 1 \end{bmatrix} M \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

$$M = \sum_{(x,y)\in\Omega} w(x,y)\begin{bmatrix} I_x^2 & I_xI_y & I_xI_t \\ I_yI_x & I_y^2 & I_yI_t \\ I_tI_x & I_tI_y & I_t^2 \end{bmatrix}$$

Lucas-Kanade flow

➢ Solve independently for each point [Lucas & Kanade 1981]

$$\frac{\partial E(u,v)}{\partial(u,v)} = 0 \Rightarrow \begin{bmatrix} \sum_{(x,y)\in\Omega} w(x,y) I_x^2 & \sum_{(x,y)\in\Omega} w(x,y) I_x I_y \\ \sum_{(x,y)\in\Omega} w(x,y) I_y I_x & \sum_{(x,y)\in\Omega} w(x,y) I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} \sum_{(x,y)\in\Omega} w(x,y) I_x I_t \\ \sum_{(x,y)\in\Omega} w(x,y) I_y I_t \end{bmatrix}$$

$$G_\sigma * \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -G_\sigma * \begin{bmatrix} I_x I_t \\ I_y I_t \end{bmatrix}$$

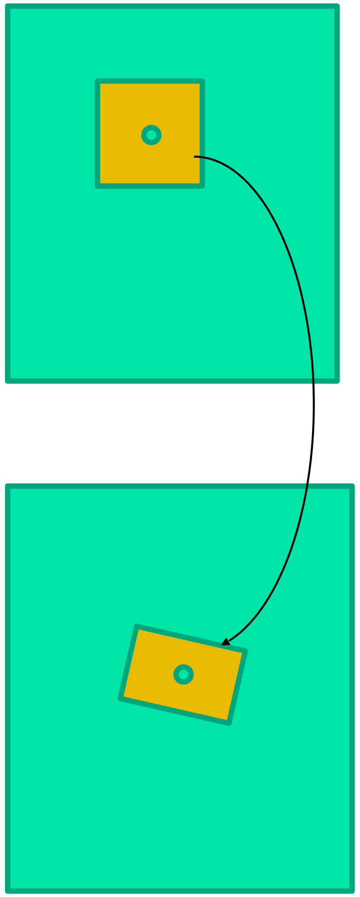
Affine Motion

- Affine Motion

$$u(x, y) = a_1 + a_2 x + a_3 y$$
$$v(x, y) = a_4 + a_5 x + a_6 y$$

- Substituting into the brightness constancy equation:

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



Affine Motion

- Affine Motion

$$u(x, y) = a_1 + a_2 x + a_3 y$$

$$v(x, y) = a_4 + a_5 x + a_6 y$$

- Substituting into the brightness constancy equation:

$$\boxed{I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t \approx 0}$$

- Each pixel provides 1 linear constraint in 6 unknowns.
- Spatial coherence constrains, Least squares minimization:

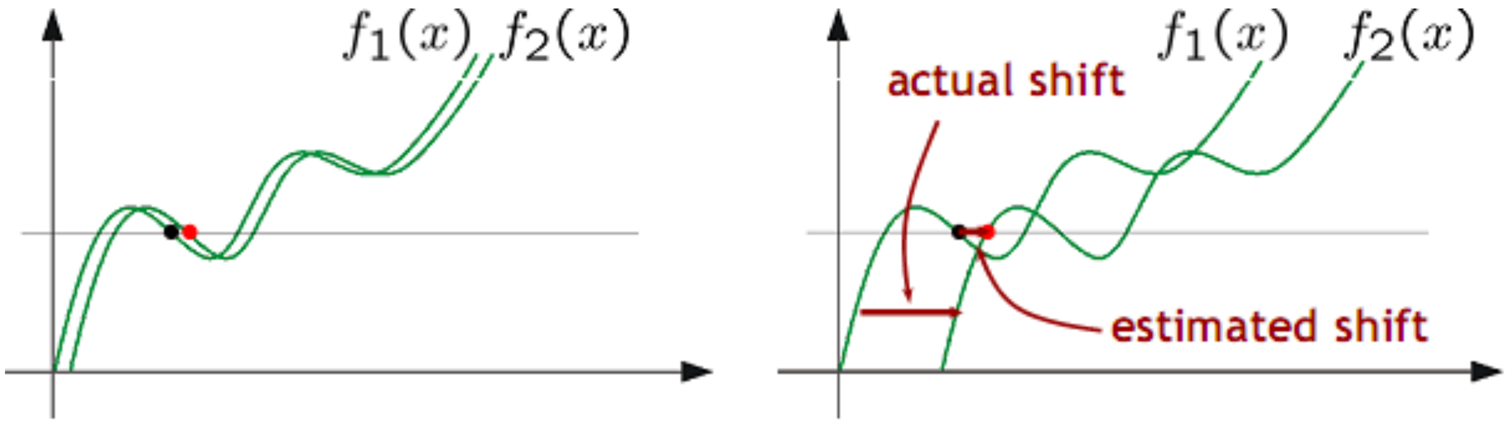
$$\boxed{Err(\vec{a}) = \sum \left[I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t \right]^2}$$

Problem Cases in Lucas-Kanade

- **The motion is large** (larger than a pixel)
  - ***Iterative refinement, coarse-to-fine estimation***
- **A point does not move like its neighbors**
  - ***Motion segmentation***
- **Brightness constancy does not hold**
  - *Do exhaustive neighborhood search with **normalized correlation**.*

Dealing with Large Motions/ Temporal Aliasing

- Temporal aliasing causes ambiguities in optical flow because images can have **many pixels with the same intensity.**
  - I.e., how do we know which 'correspondence' is correct?

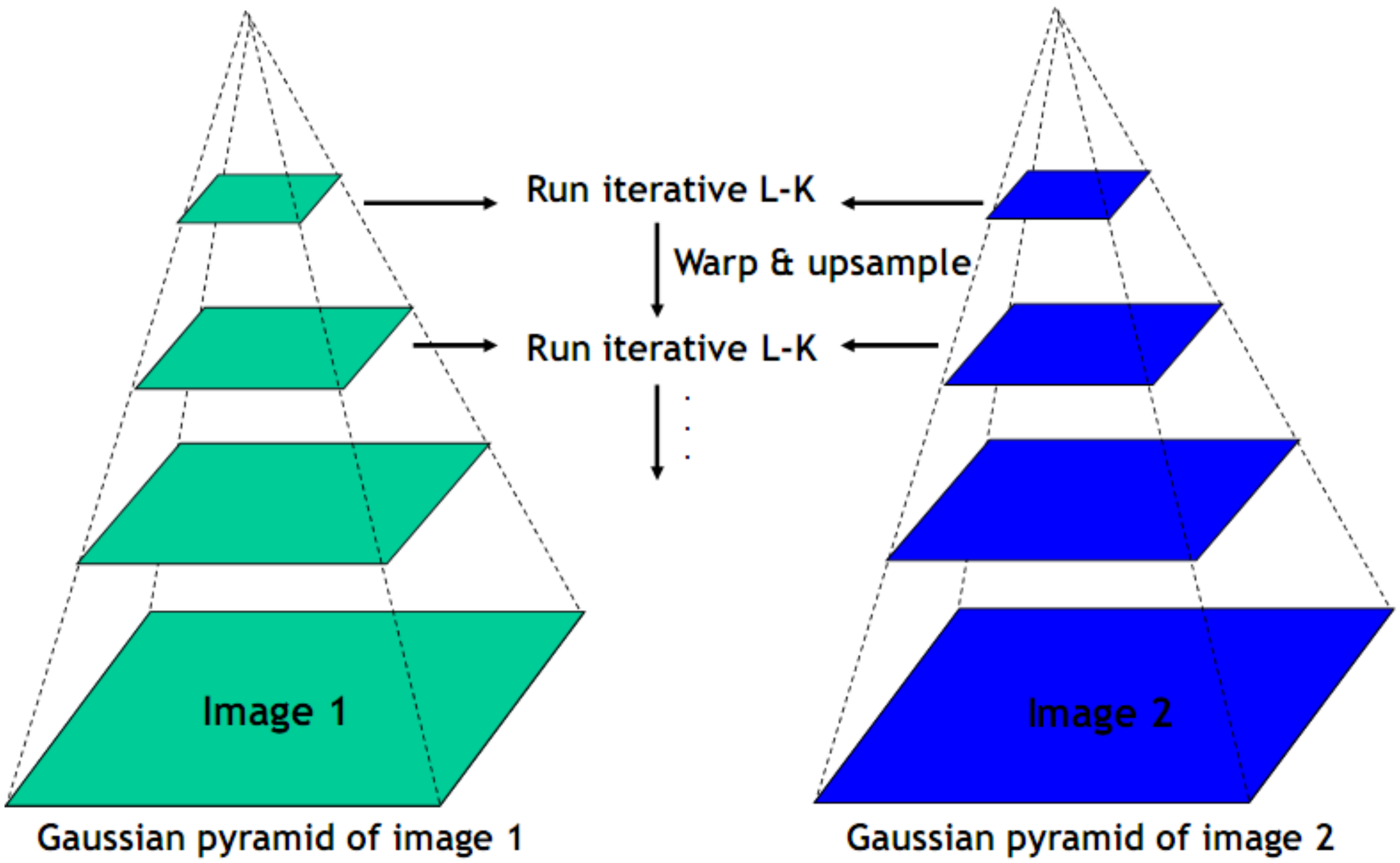


Nearest match is correct (no aliasing)

Nearest match is incorrect (aliasing)

To overcome aliasing: coarse-to-fine estimation.

Coarse-to-fine Optical Flow Estimation



Jean-Yves Bouguet, *Pyramidal Implementation of the Lucas Kanade Feature Tracker*, TR, Intel, , 1997

Extension: Gradient constancy

Brightness is not always constant

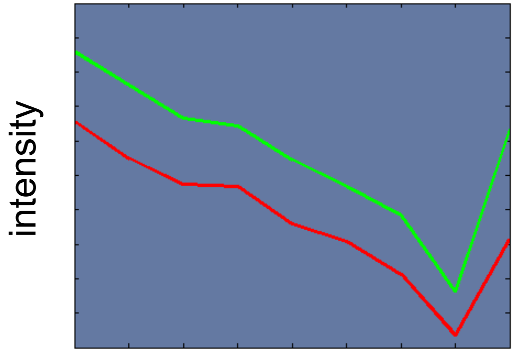
Rotating cylinder

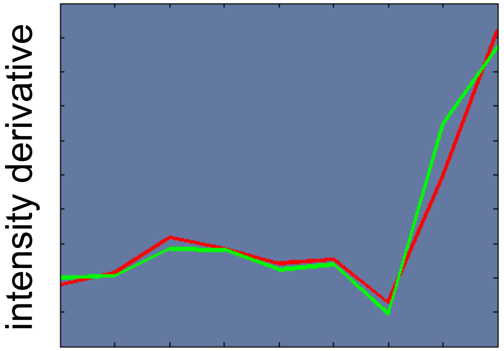
Brightness constancy does not always hold

Gradient constancy holds

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

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





Local constraints (data) + Local spatial coherence

- **Brightness constancy**

$$I(x+u, y+v, t+1) - I(x, y, t) = 0$$

linearized

$$[u \;\; v \;\; 1] \;\; J \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \cong 0$$

averaged linearized

Local spatial coherence

$$\delta^2_{LIN+GAUSS} = [u \;\; v \;\; 1] \; (G_\rho * J) \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \cong 0$$

- **Gradient constancy**

$$\nabla I(x+u, y+v, t+1) - \nabla I(x, y, t) = 0$$

Slide credit: D. Frolova and D. Simakov

Feature Tracking

(sparse optical flow)

Tracking Challenges

- **Ambiguity of optical flow**
  - *Find good features to track*
- **Large motions**
  - *Discrete search instead of Lucas-Kanade*
- **Changes in shape, orientation, color**
  - *Allow some matching flexibility*
- **Occlusions, disocclusions**
  - *Need mechanism for deleting, adding new features*
- **Drift – errors may accumulate over time**
  - *Need to know when to terminate a track*

Handling Large Displacements

- Define a small area around a pixel as the **template**.
- Match the template against each pixel **within a search area** in next image – *just like stereo matching!*
- Use a match measure such as SSD or correlation.
- After finding the best discrete location, can use Lucas-Kanade to get **sub-pixel estimate**.

离散粗定位+基于梯度精细搜索

Tracking Over Many Frames

- Select features in first frame
- For each frame:
  - Update positions of tracked features
    - **Discrete search** or **Lucas-Kanade** (Image gradient)
  - Terminate inconsistent tracks
    - **Compute similarity with corresponding feature** in the previous frame or in the first frame where it's visible
- Start new tracks if needed
  - Typically every ~10 frames, new features are added to "refill the ranks"

Shi-Tomasi Feature Tracker

- Find good features using eigenvalues of second-moment matrix
  - Key idea: “good” features to track are the ones that can be tracked reliably.
- From frame to frame, track with Lucas-Kanade and a pure translation model.
  - More robust for small displacements, can be estimated from smaller neighborhoods.
- Check consistency of tracks by affine registration to the first observed instance of the feature.
  - Affine model is more accurate for larger displacements.
  - Comparing to the first frame helps to minimize drift.

J. Shi and C. Tomasi. **Good Features to Track**. CVPR 1994

KLT—Pyramidal tracking algorithm (Coarse-to-fine)

Goal: Let $\mathbf{u}$ be a point on image I. Find its corresponding location $\mathbf{v}$ on image J

Build pyramid representations of I and J: $\{I^L\}_{L=0,\dots,L_m}$ and $\{J^L\}_{L=0,\dots,L_m}$

Initialization of pyramidal guess: $\mathbf{g}^{L_m} = [g_x^{L_m} \;\; g_x^{L_m}]^T = [0 \;\; 0]^T$

for $L = L_m$ **down to** 0 **with step of** -1

Location of point $\mathbf{u}$ on image I^L: $\mathbf{u}^L = [p_x \;\; p_y]^T = \mathbf{u}/2^L$

Derivative of I^L with respect to x: $$I_x(x,y) = \frac{I^L(x+1,y) - I^L(x-1,y)}{2}$$

Derivative of I^L with respect to y: $$I_y(x,y) = \frac{I^L(x,y+1) - I^L(x,y-1)}{2}$$

Spatial gradient matrix: $$G = \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} \begin{bmatrix} I_x^2(x,y) & I_x(x,y)\, I_y(x,y) \\ I_x(x,y)\, I_y(x,y) & I_y^2(x,y) \end{bmatrix}$$

Initialization of iterative L-K: $\overline{\nu}^0 = [0 \;\; 0]^T$

for $k = 1$ **to** K **with step of** 1 (or until $\|\overline{\eta}^k\| <$ accuracy threshold)

Image warp

Image difference: $\delta I_k(x,y) = I^L(x,y) - \boxed{J^L(x + \underline{g_x^L + \nu_x^{k-1}}, y + \underline{g_y^L + \nu_y^{k-1}})}$

Image mismatch vector: $\bar{b}_k = \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} \begin{bmatrix} \delta I_k(x,y)\, I_x(x,y) \\ \delta I_k(x,y)\, I_y(x,y) \end{bmatrix}$

Optical flow (Lucas-Kanade): $\overline{\eta}^k = G^{-1}\, \bar{b}_k$

Guess for next iteration: $\overline{\nu}^k = \overline{\nu}^{k-1} + \overline{\eta}^k$

end of for-loop on k

Final optical flow at level L*:* $\mathbf{d}^L = \overline{\nu}^K$

Guess for next level $L-1$*:* $\mathbf{g}^{L-1} = [g_x^{L-1} \;\; g_y^{L-1}]^T = 2\,(\mathbf{g^L} + \mathbf{d}^L)$

end of for-loop on L

Final optical flow vector: $\mathbf{d} = \mathbf{g}^0 + \mathbf{d}^0$

Location of point on J*:* $\mathbf{v} = \mathbf{u} + \mathbf{d}$

Solution: The corresponding point is at location $\mathbf{v}$ on image J

Real-Time GPU Implementations

- This basic feature tracking framework (Lucas-Kanade + Shi-Tomasi) is commonly referred to as “KLT tracking”.
  - Used as preprocessing step for many applications (recall the Structure-from-Motion pipeline)
  - Lends itself to easy parallelization
- Very fast GPU implementations available
  - C. Zach, D. Gallup, J.-M. Frahm, Fast Gain-Adaptive KLT tracking on the GPU. In CVGPU’08 Workshop, Anchorage, USA, 2008
  - 216 fps with automatic gain adaptation
  - 260 fps without gain adaptation

http://www.cs.unc.edu/~ssinha/Research/GPU_KLT/
http://cs.unc.edu/~cmzach/opensource.html

Summary

✦ Motion field & Optical flow field

✦ Optical flow equation & aperture problem

✦ LK's Method and Horn' Method

✦ Feature Tracking & Sparse optical flow