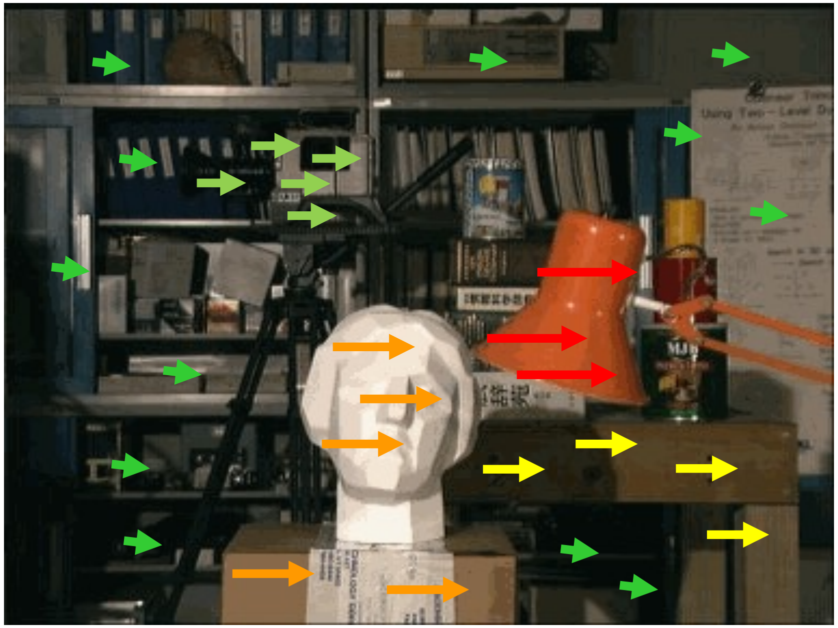
From 1D correspondence (stereo) to 2D correspondence problems (motion)

1D shifts along **epipolar lines**.

Assumption for stereo:

only camera moves,
<u>3D scene is stationary</u>



vector field (motion) with a priori known direction

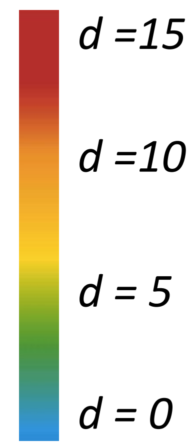
Slide credit: Yuri Boykov, Boqing Gong, Ce Liu, Steve Seitz, Larry Zitnick, Ali Farhadi

From 1D correspondence (stereo) to 2D correspondence problems (motion)

1D shifts along **epipolar lines**.

Assumption for stereo:

only camera moves,
<u>3D scene is stationary</u>



vector field (motion) with a priori *known direction*

⇨ We estimate only *magnitude* represented by a **scalar field** (disparity map)

From 1D correspondence (stereo) to 2D correspondence problems (motion)

In general, correspondences between two images
may not be described by global models (like *homography*) or
by shifts along known **epipolar lines**.

if 3D scene
is <u>NOT stationary</u>
motion is
vector field
with **arbitrary directions**
(no epipolar line constraints)

Society Of Robots

From 1D correspondence (stereo) to 2D correspondence problems (motion)

In general, correspondences between two images may not be described by global models (like *homography*) or by shifts along known **epipolar lines**.

For (non-rigid) motion the correspondences between two video frames are described by a general ***optical flow***

if 3D scene is NOT stationary motion is **vector field** with **arbitrary directions**
(no epipolar line constraints)

Society Of Robots

The cause of motion

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

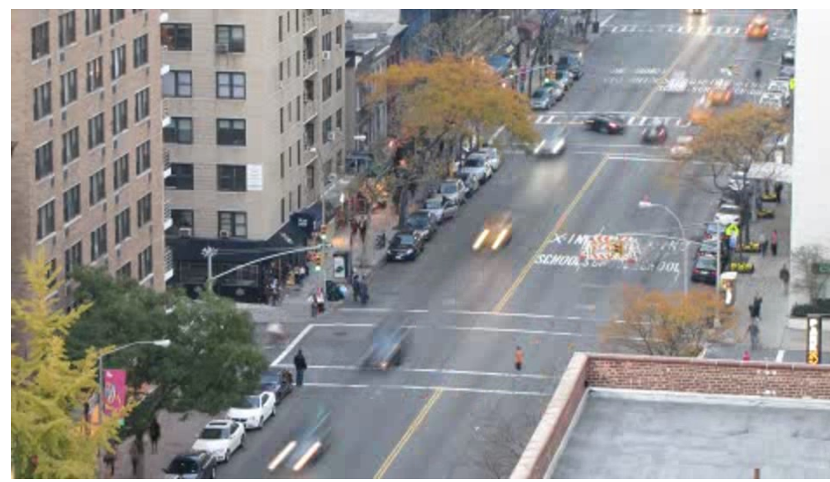
- Varying either of them causes motion
  - Static camera, moving objects (surveillance)
  - Moving camera, static scene (3D capture)
  - Moving camera, moving scene (sports, movie)
  - Static camera, moving objects, moving light (time lapse)

Motion scenarios (priors)

Static camera, moving scene

Moving camera, static scene

Moving camera, moving scene



Static camera, moving scene, moving light

We still don't touch these areas

How can we recover motion?

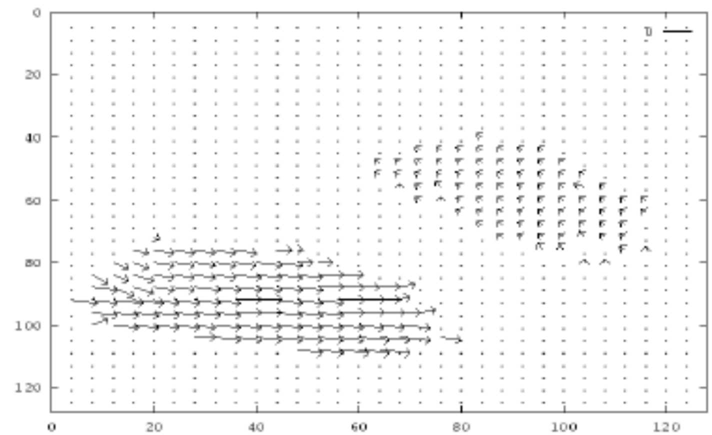
Recovering motion

- Feature-tracking
  - Extract visual features (corners, textured areas) and "track" them over multiple frames
- Optical flow
  - Recover image motion at each pixel from spatio-temporal image brightness variations (optical flow)

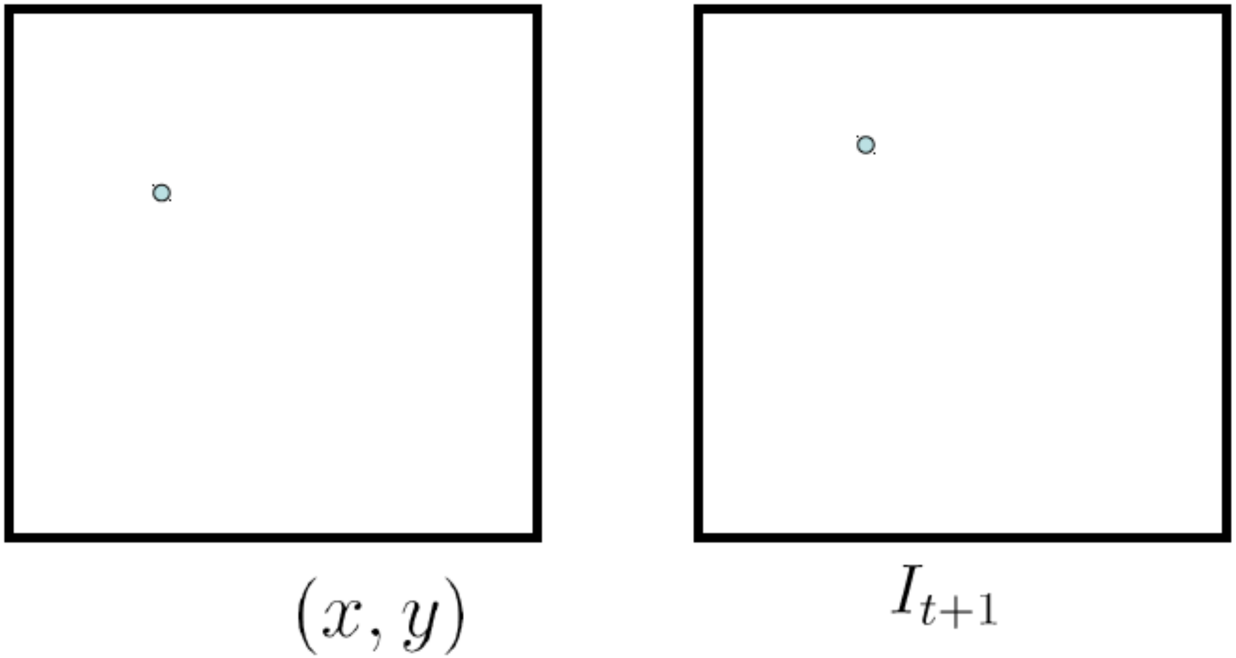
Two problems, one registration method

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.

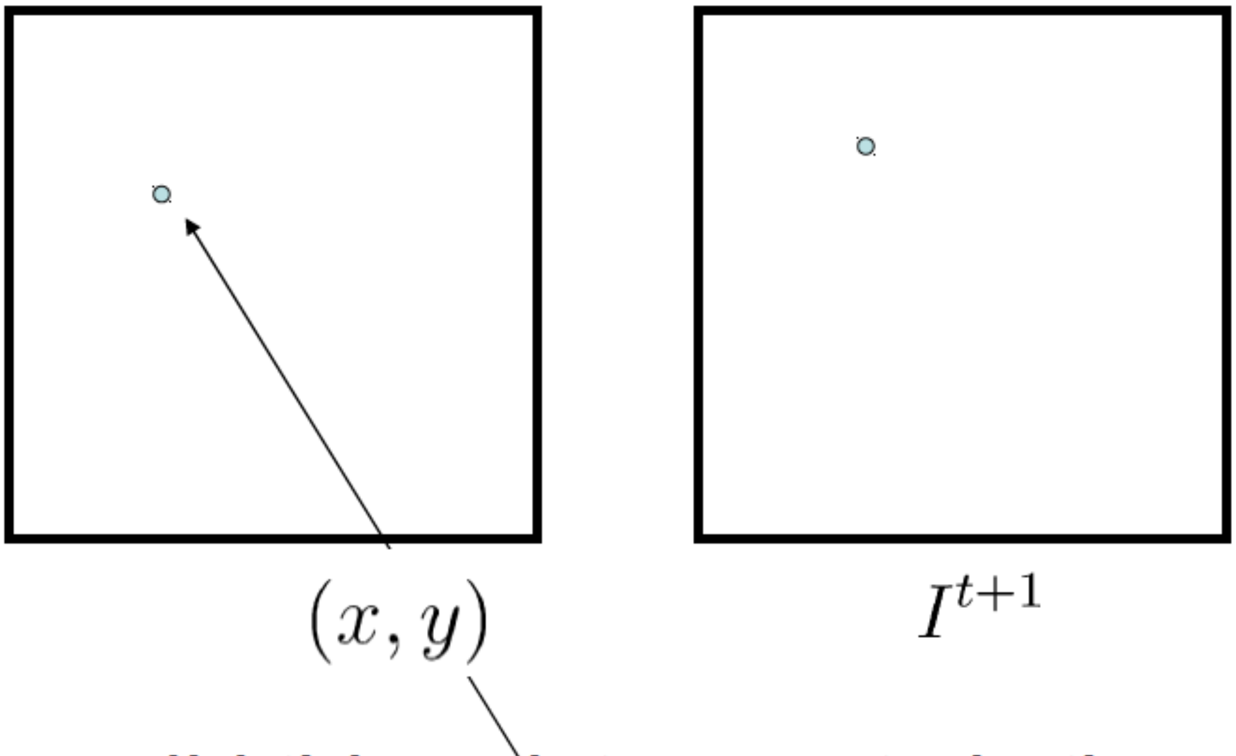
Hamburg Taxi seq



Basic Setup

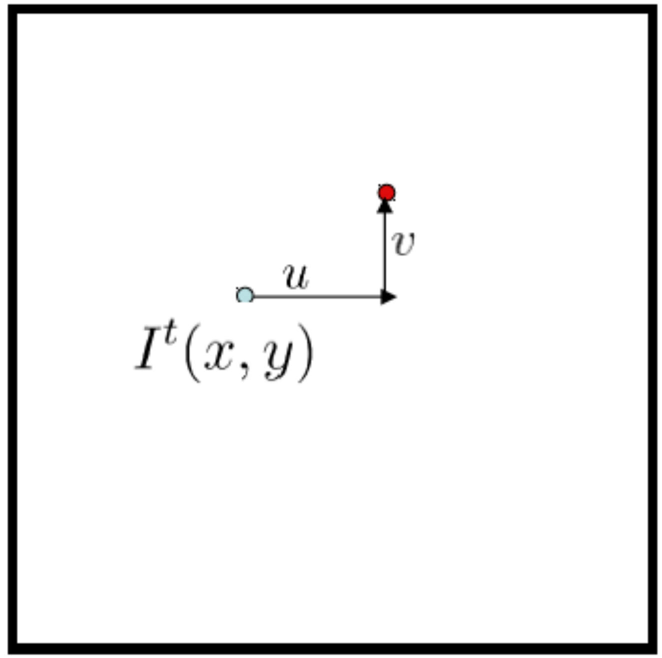


Basic Question



- Where did this point move to in the next image?

Basic Assumption



- Image Brightness Constancy Equation:

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

- Assumes that the scene doesn't change intensity

Understanding the assumption

$$I(x,y,t) \approx I(x_{t_0}, y_{t_0}, t_{t_0}) + \frac{\partial I}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial I}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial I}{\partial t}\frac{\partial t}{\partial t}$$

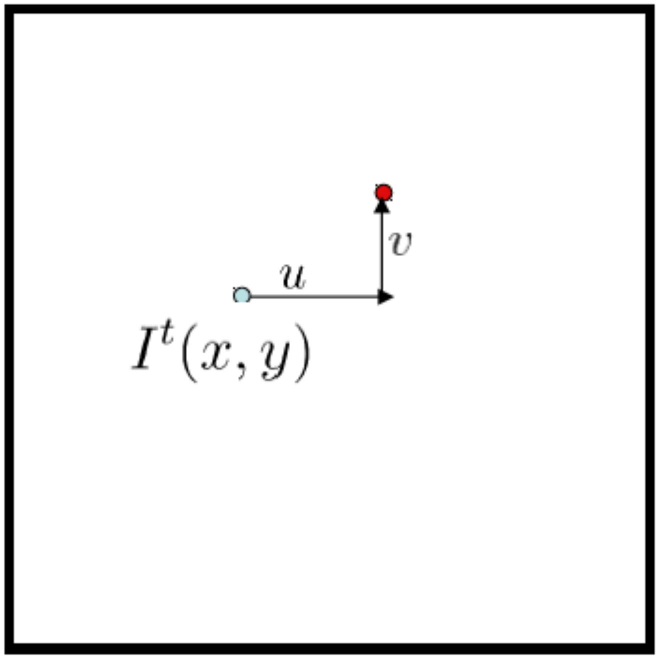
Again, if we assume that the intensity of the scene doesn't change, then

$$I(x,y,t) = I(x_{t_0}, y_{t_0}, t_{t_0})$$

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

Extended reading: Taylor expansion

Understanding the assumption

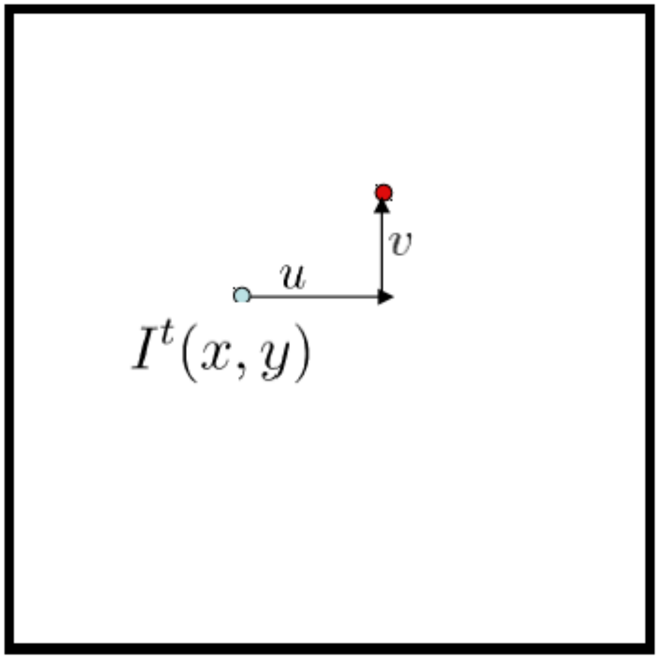


$$\frac{\partial I}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial I}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial I}{\partial t}\frac{\partial t}{\partial t} = 0$$

⇕

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

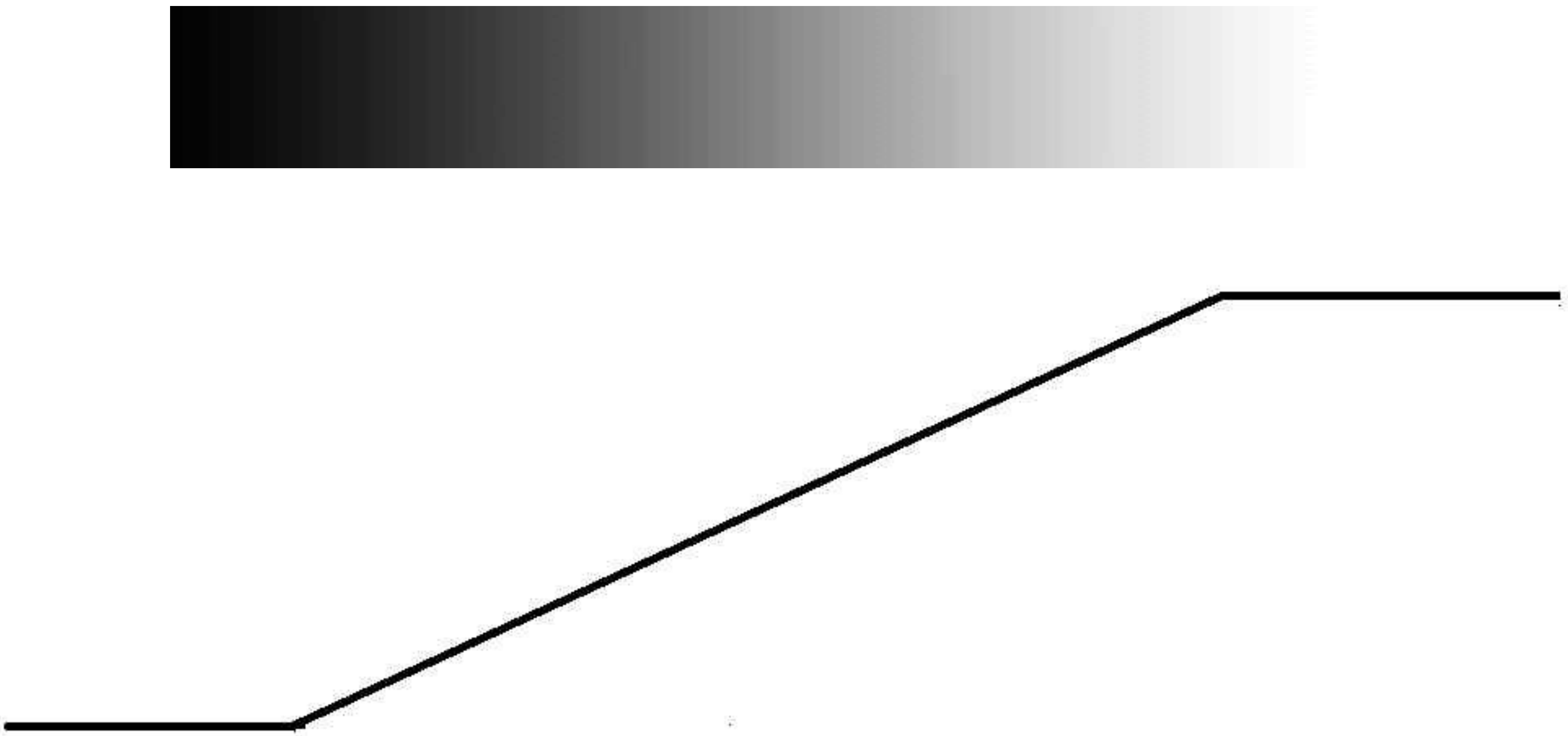
What's Next:
Another way of deriving the equation



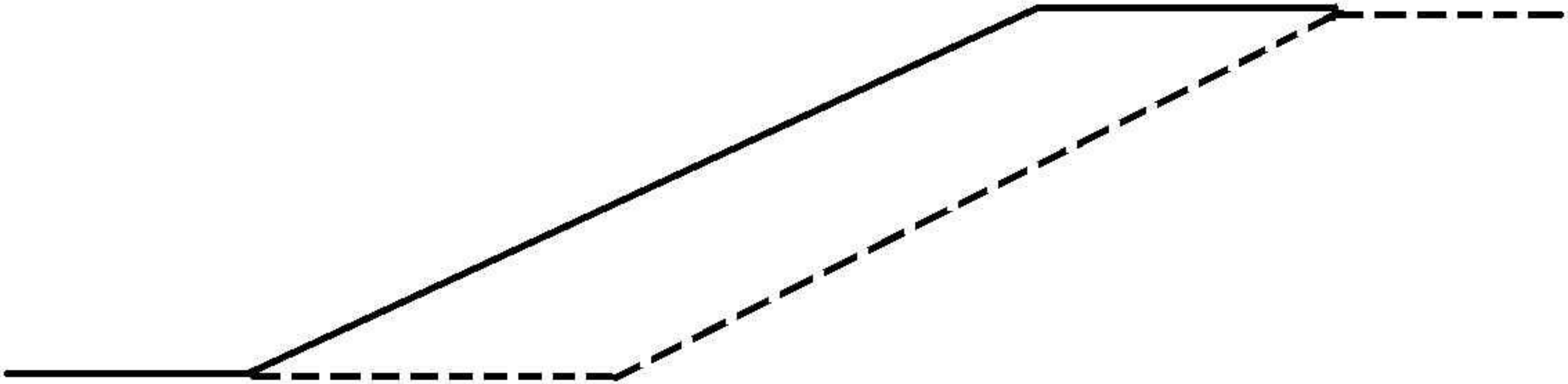
$$I_x u + I_y v + I_t = 0$$

Imagine there is this (ramp) pattern of Intensity (image brightness) being viewed from above.

The ramp’s pattern of brightness (Intensity Profile) can be viewed as a plot.

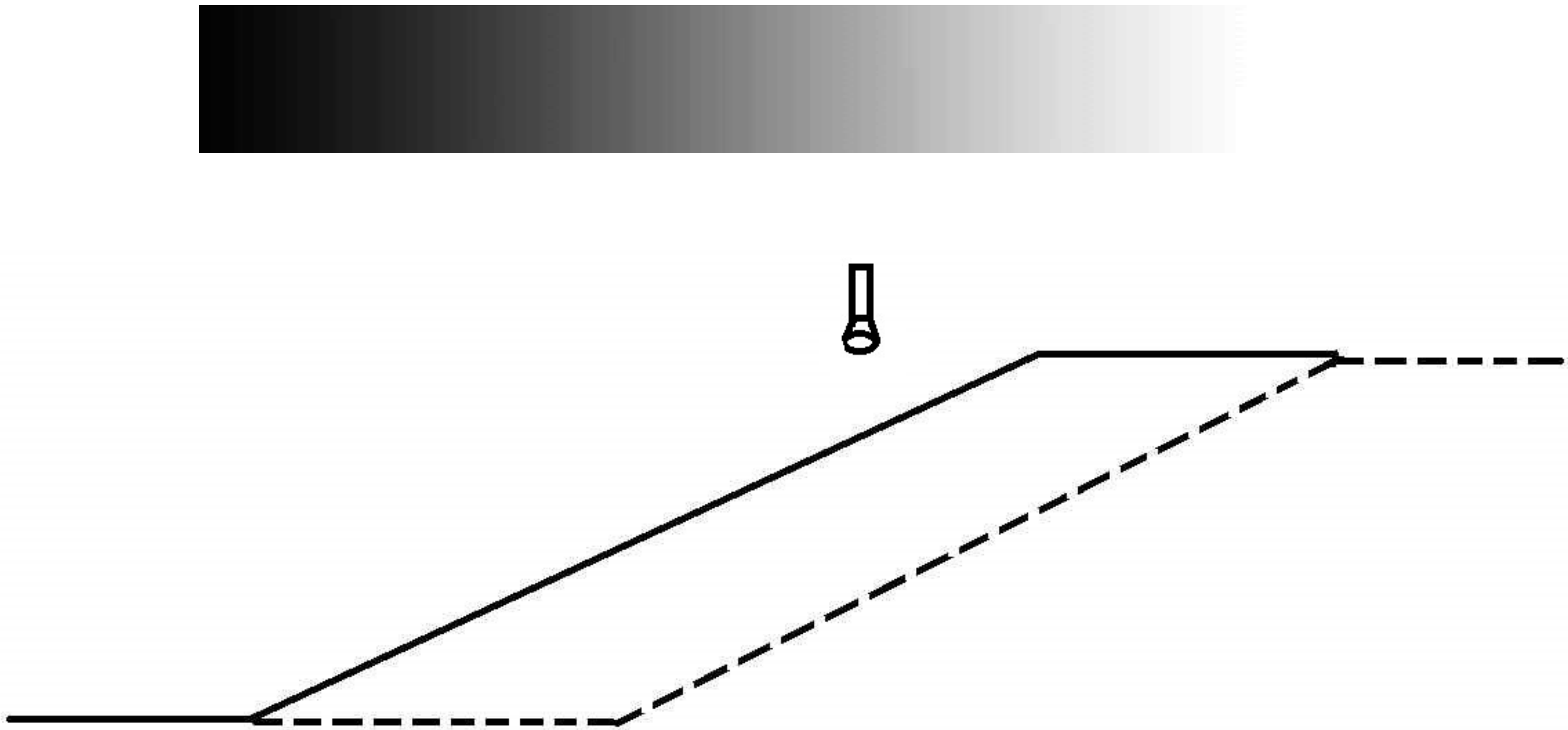


Now, Suppose the pattern moves to the right.

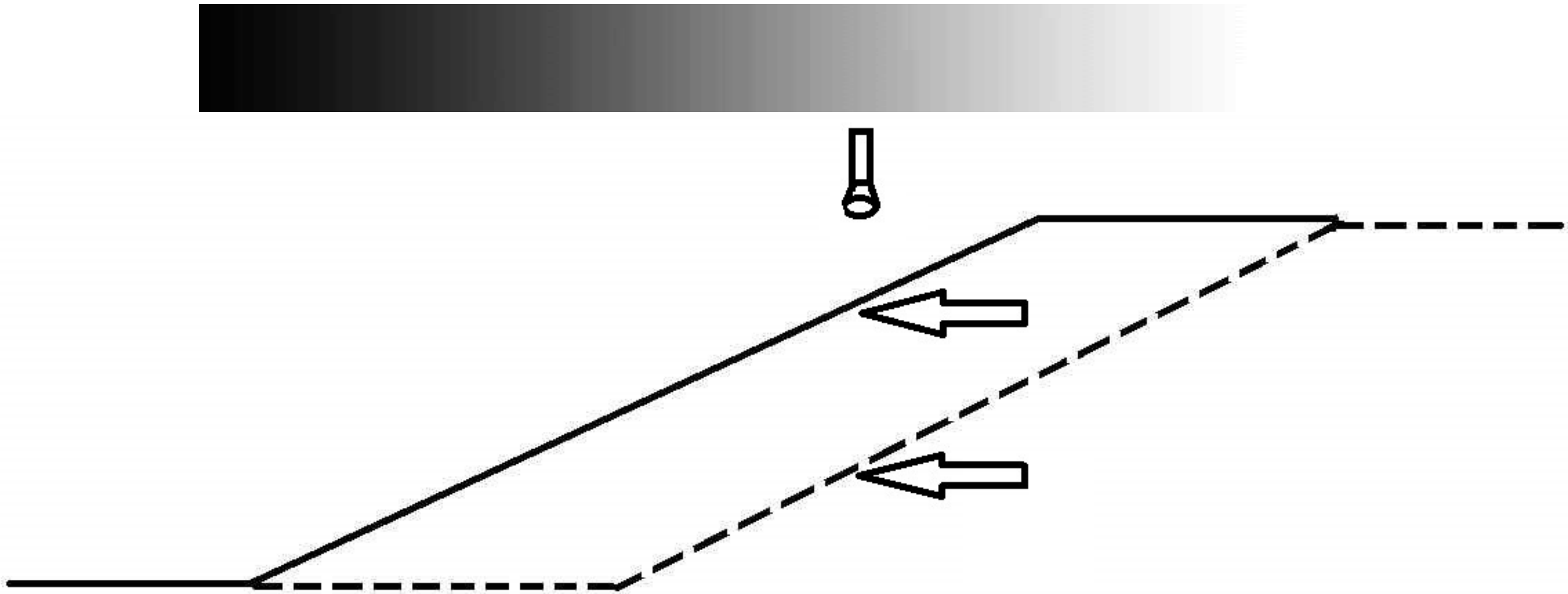


Dotted line shows where pattern has moved to, within unit time.

Now, Suppose a Single Pixel Camera saw this

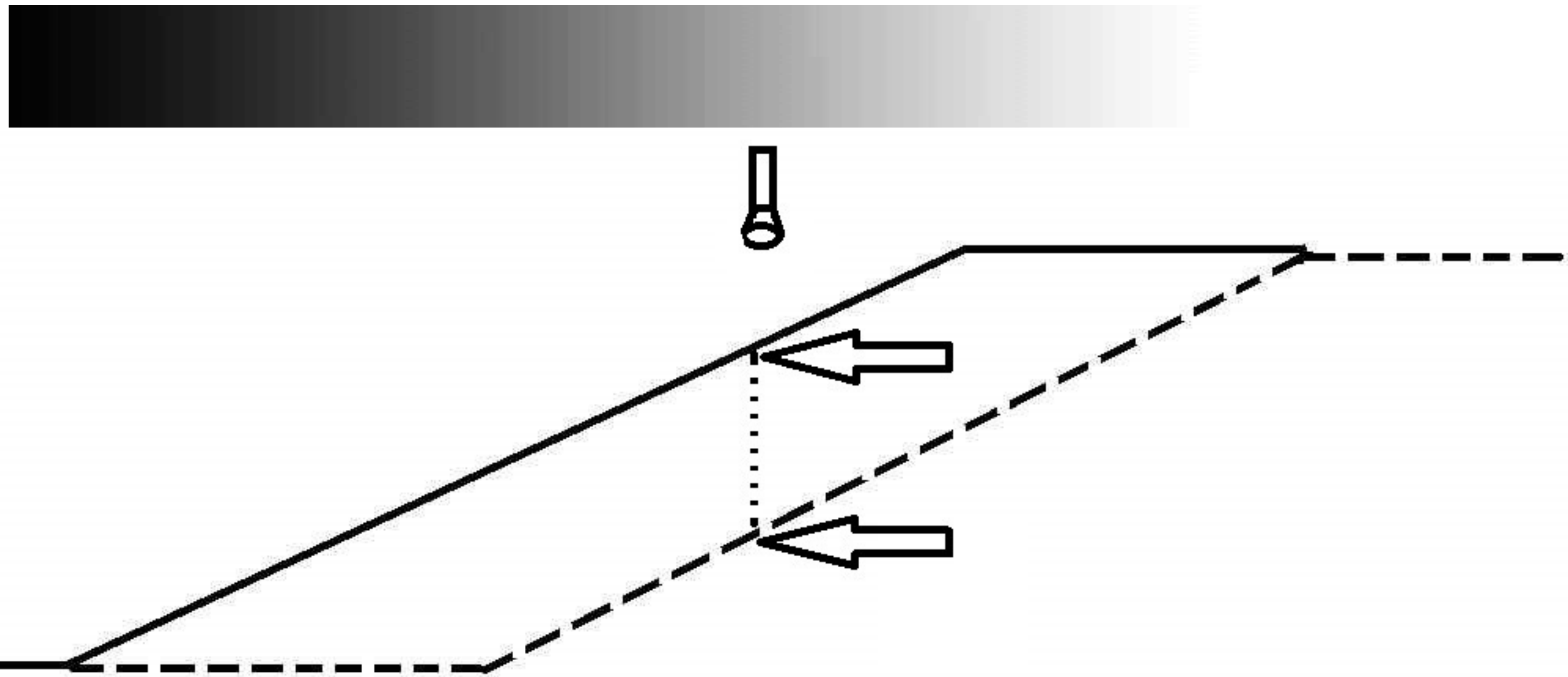


The Sensor would see the brightness change



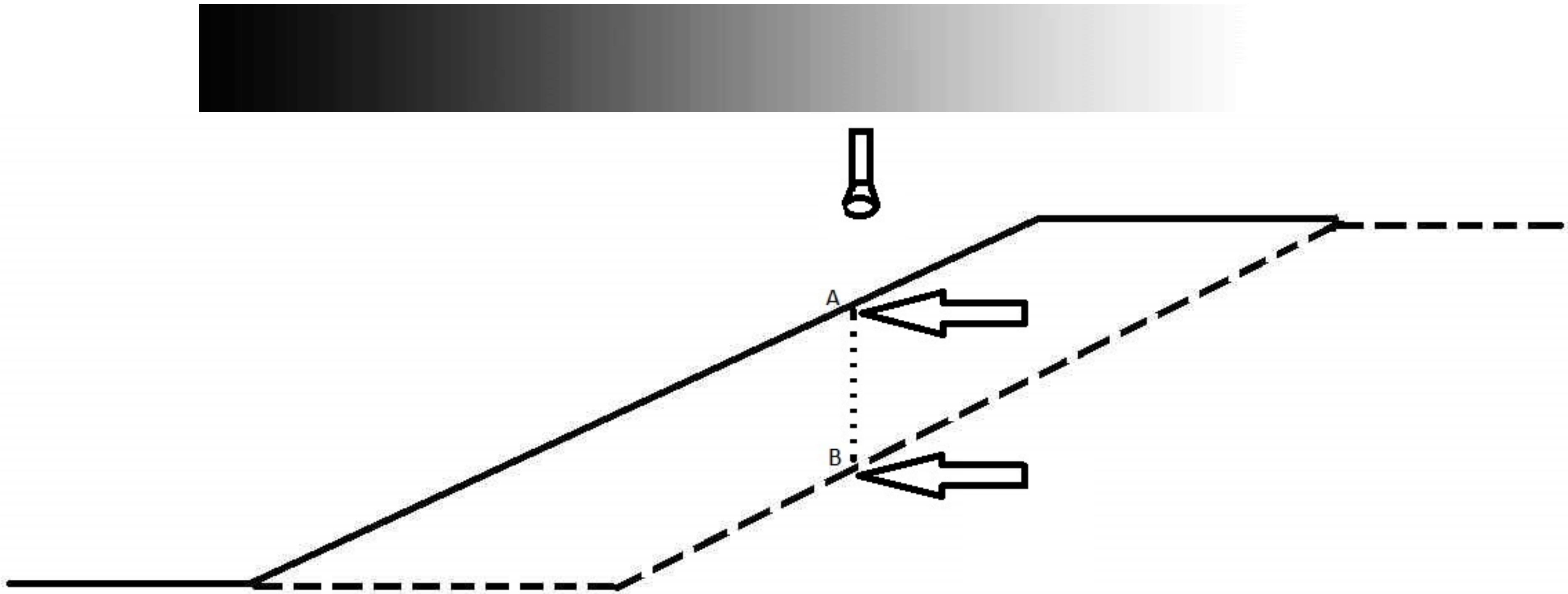
The 2 arrows show what two brightnesses the detector sees

The Sensor would see the brightness change



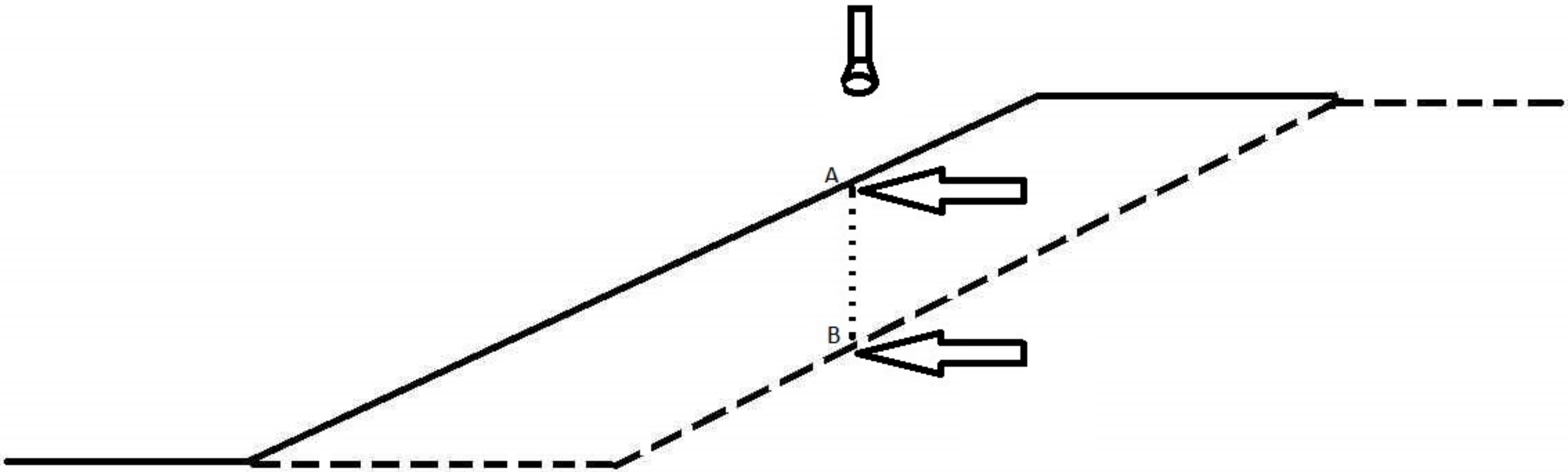
The Dotted segment shows the amount of brightness drop.

The Sensor would see the brightness change

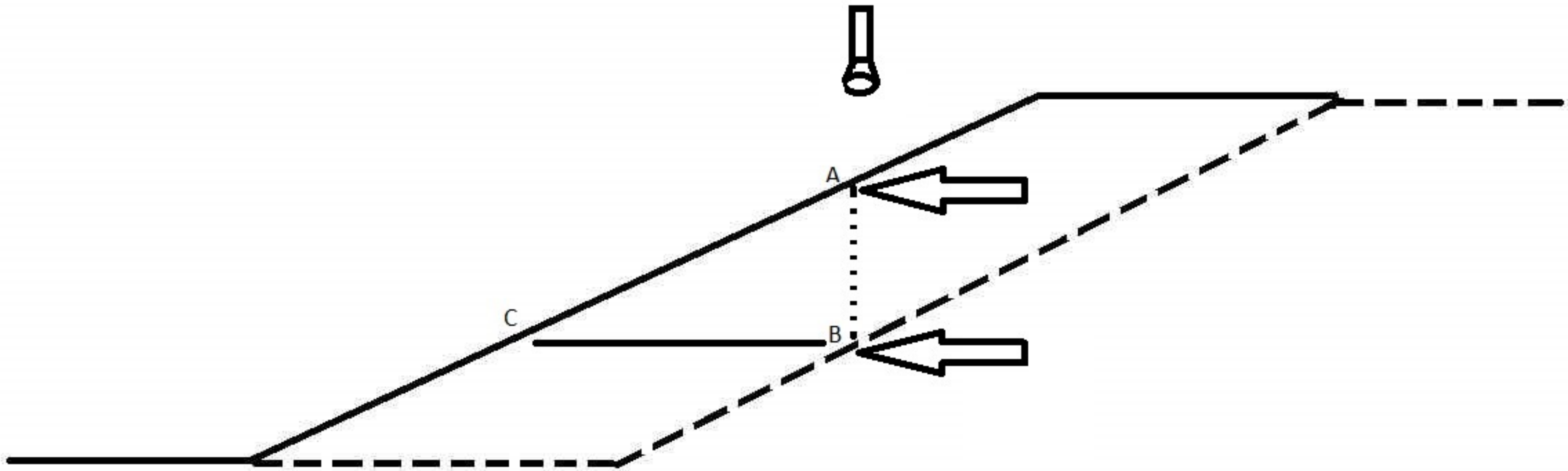


The brightness drop is given by the distance between A and B.

What physical properties of this situation will determine how much brightness drop happens?

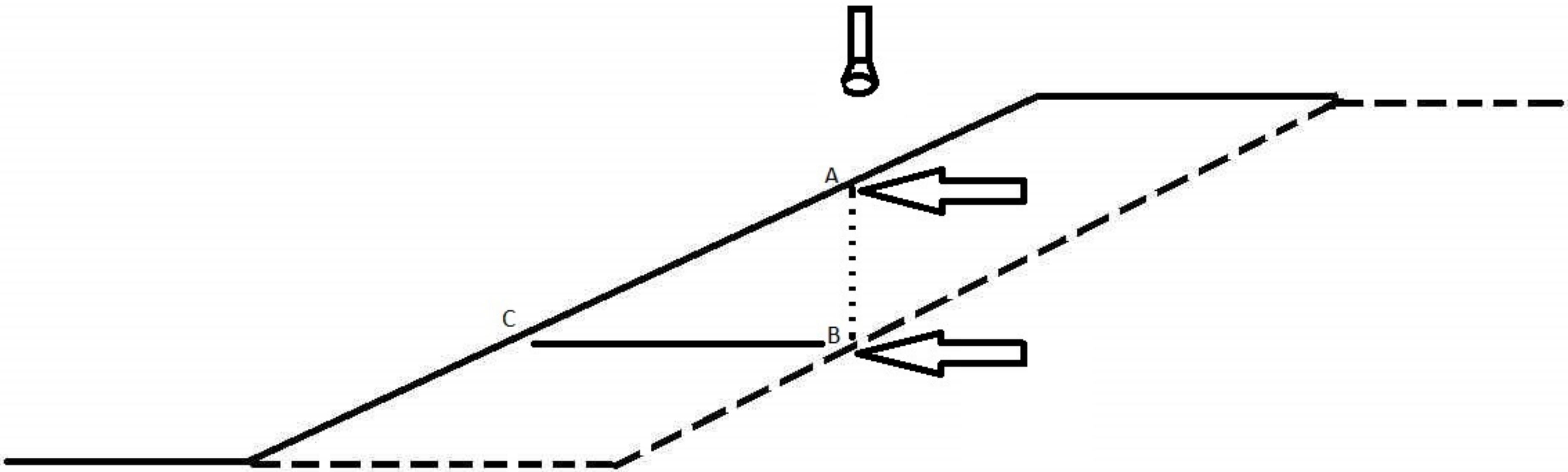


One factor contributing to the quantity of the brightness drop, is the speed.



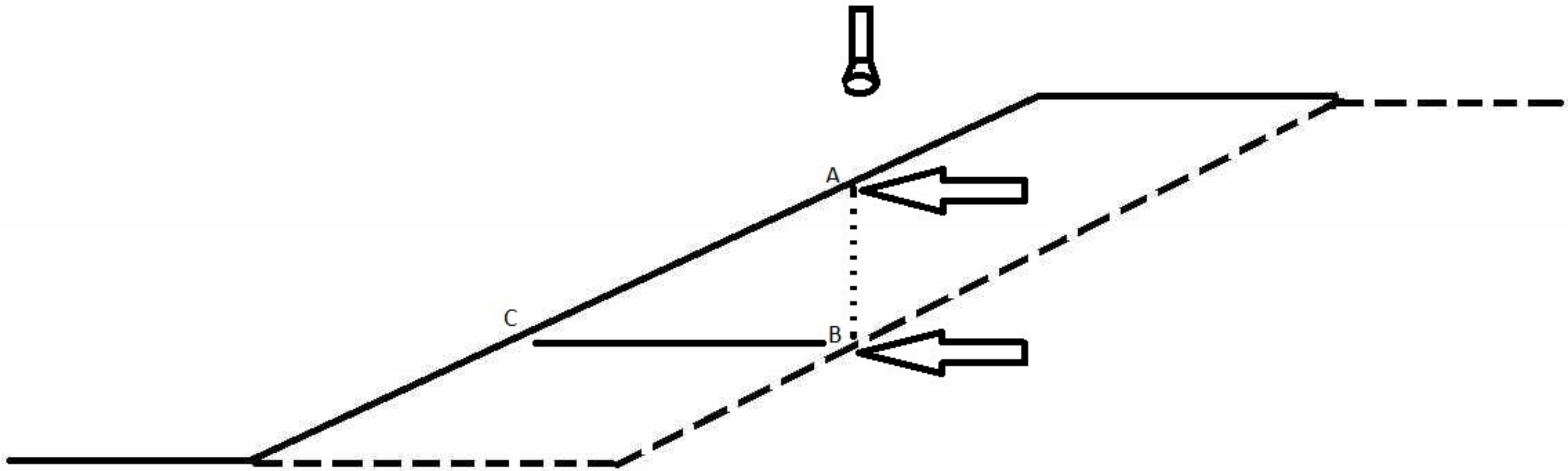
Speed is given by the lowermost dashed segment, which is the same as the length of the segment CB.

One contributing factor is the speed.



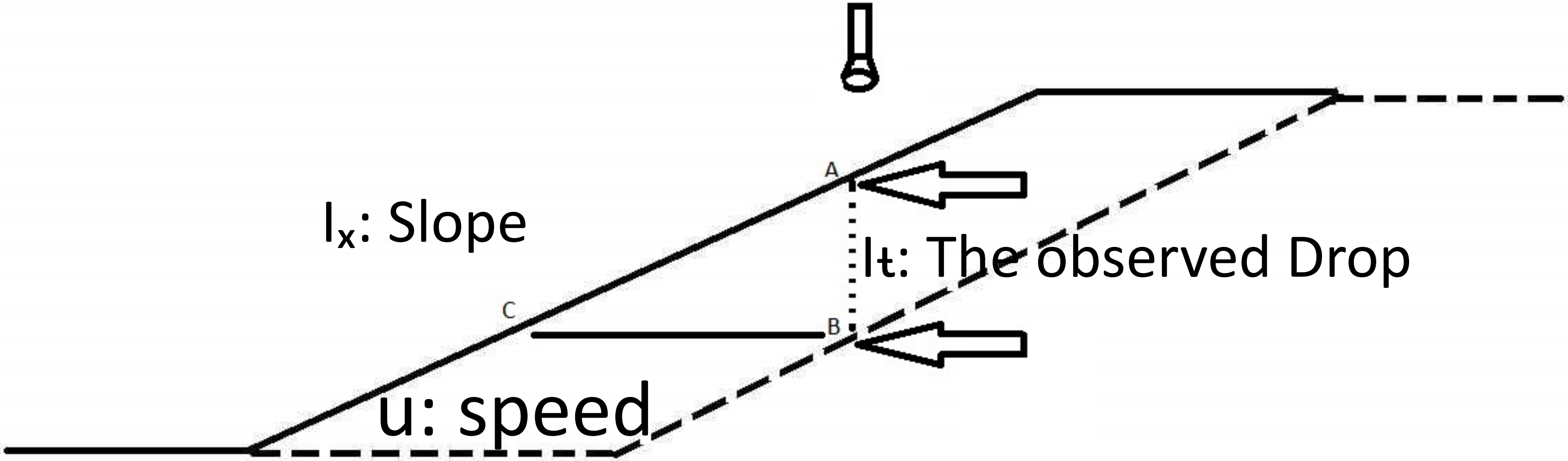
Make sure you understand that the faster speed will give a bigger drop, while slower speed will give smaller drop.

Second contributing factor is the ramp's slope.



Make sure you understand that the steeper slope will give a bigger drop, while shallower slope will give a smaller drop.

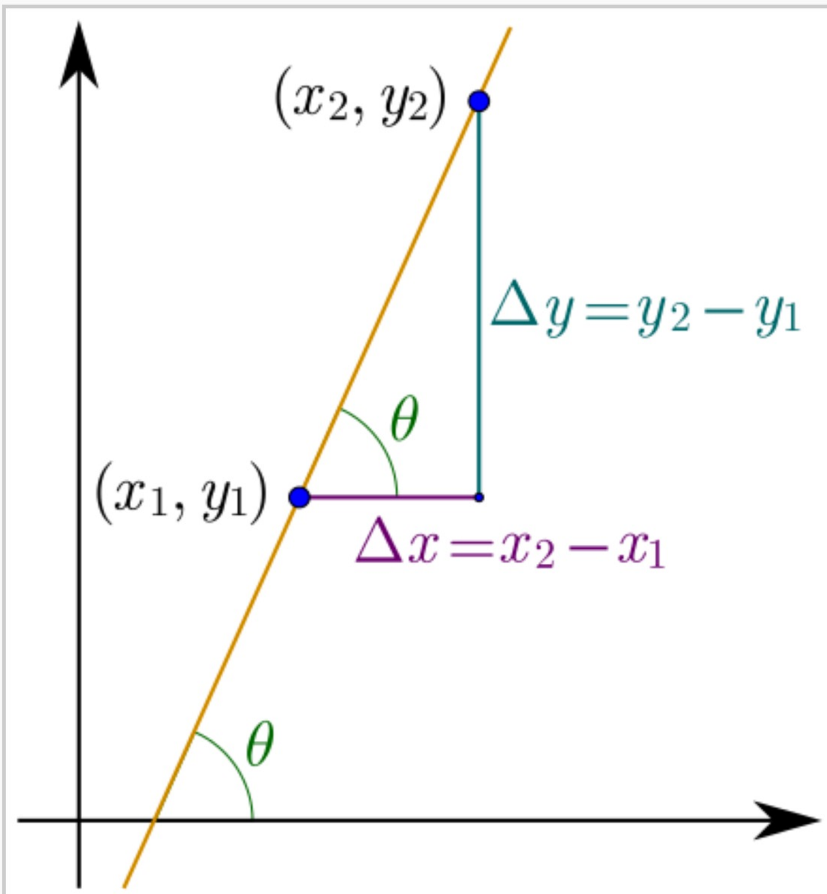
So, drop proportional to ramp's slope & speed



Let us give symbols to these quantities, so we can work them.

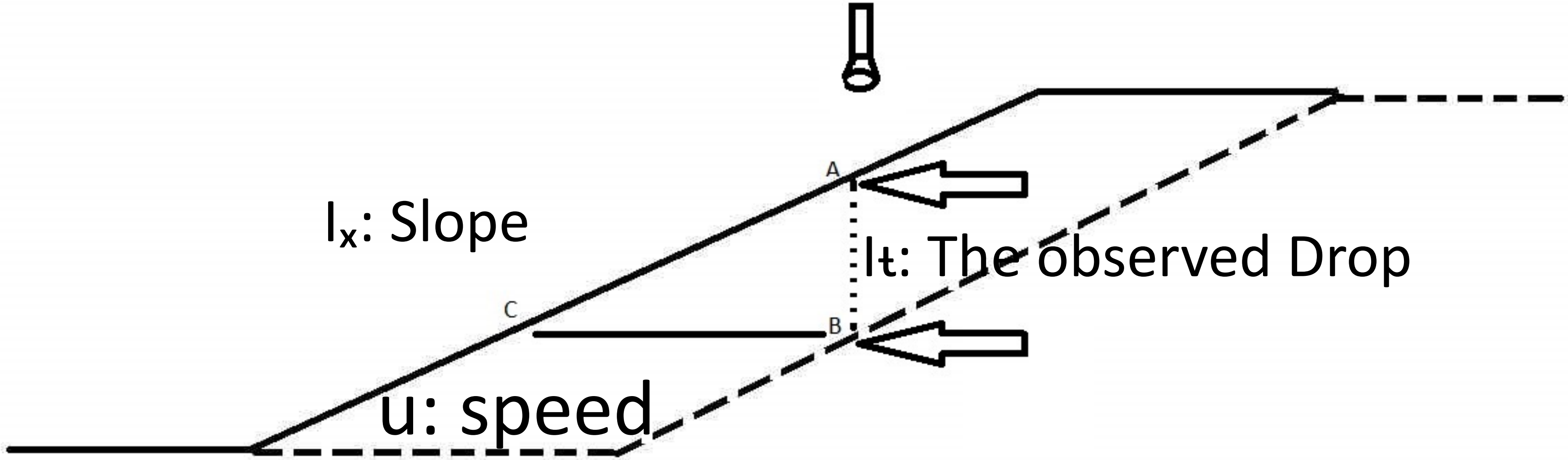
The ramp itself is labeled I(x), for image or intensity function, varying along x. Speed is labeled u. Drop is labeled I_t. Slope is labeled I_x.

Describe slope in math



Slope: $m = \frac{\Delta y}{\Delta x} = \tan(\theta)$

So, drop proportional to ramp's slope & speed



So, $-I_t = u \cdot I_x$

Now, we need to derive a similar equation for the vertical direction.

So, similar reasoning: Suppose the region has variation only in the y-direction (not shown here, the original pattern is shown, you must imagine the new pattern); suppose that the motion is in the vertical direction (called v, now), suppose there is a single pixel sensor (camera) placed over the center of the pattern.

Then, by similar reasoning as before, we get that: $-I_{t_y} = v \cdot I_y$

We had written I_t earlier, when we only had one dimension to play in. Now, to keep things separate, we say I_{t_y}, by which we mean the drop seen by the sensor, but only that portion of the drop that is due to vertical aspects of this problem (in the original equation, to describe the horizontal behavior, we will now be using I_{t_x}.) In the new equation here, the meaning of I_y should be obvious, it is the vertical component of the image gradient.

In practice, the motion could be along both x and y.

So, sum the “drops”,

$$(-I_{t_x}) + (-I_{t_y}) = u \cdot I_x + v \cdot I_y$$

The terms on the Left are to be combined into one term I_t.

$$\mathbf{-I_t = u \cdot I_x + v \cdot I_y}$$

This is a famous equation in the field of Computer Vision, and it has several names:

1) 2d motion Equation
2) Image motion Equation
3) Optical Flow Equation (this term is from perceptual psychology)

Comparing with the first way of deriving the optical flow equation

$$I(x,y,t) \approx I(x_{t_0}, y_{t_0}, t_{t_0}) + \frac{\partial I}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial I}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial I}{\partial t}\frac{\partial t}{\partial t}$$

Again, if we assume that the intensity of the scene doesn't change, then

$$I(x,y,t) = I(x_{t_0}, y_{t_0}, t_{t_0})$$

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

$$\Updownarrow$$

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

The brightness constancy constraint

Can we use this equation to recover image motion (u,v) at each pixel?

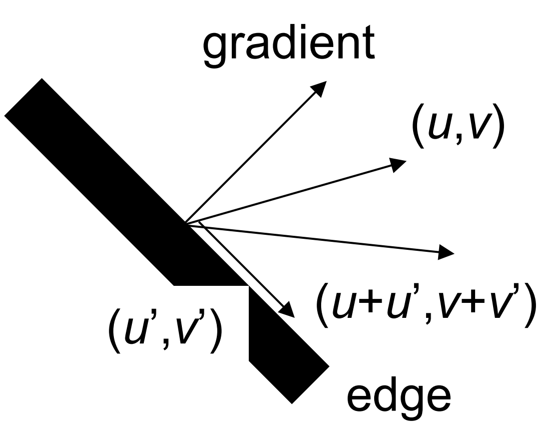
$$\nabla I \cdot [u \;\; v]^T + I_t = 0$$

- How many equations and unknowns per pixel?
  - One equation (this is a scalar equation!), two unknowns (u,v)

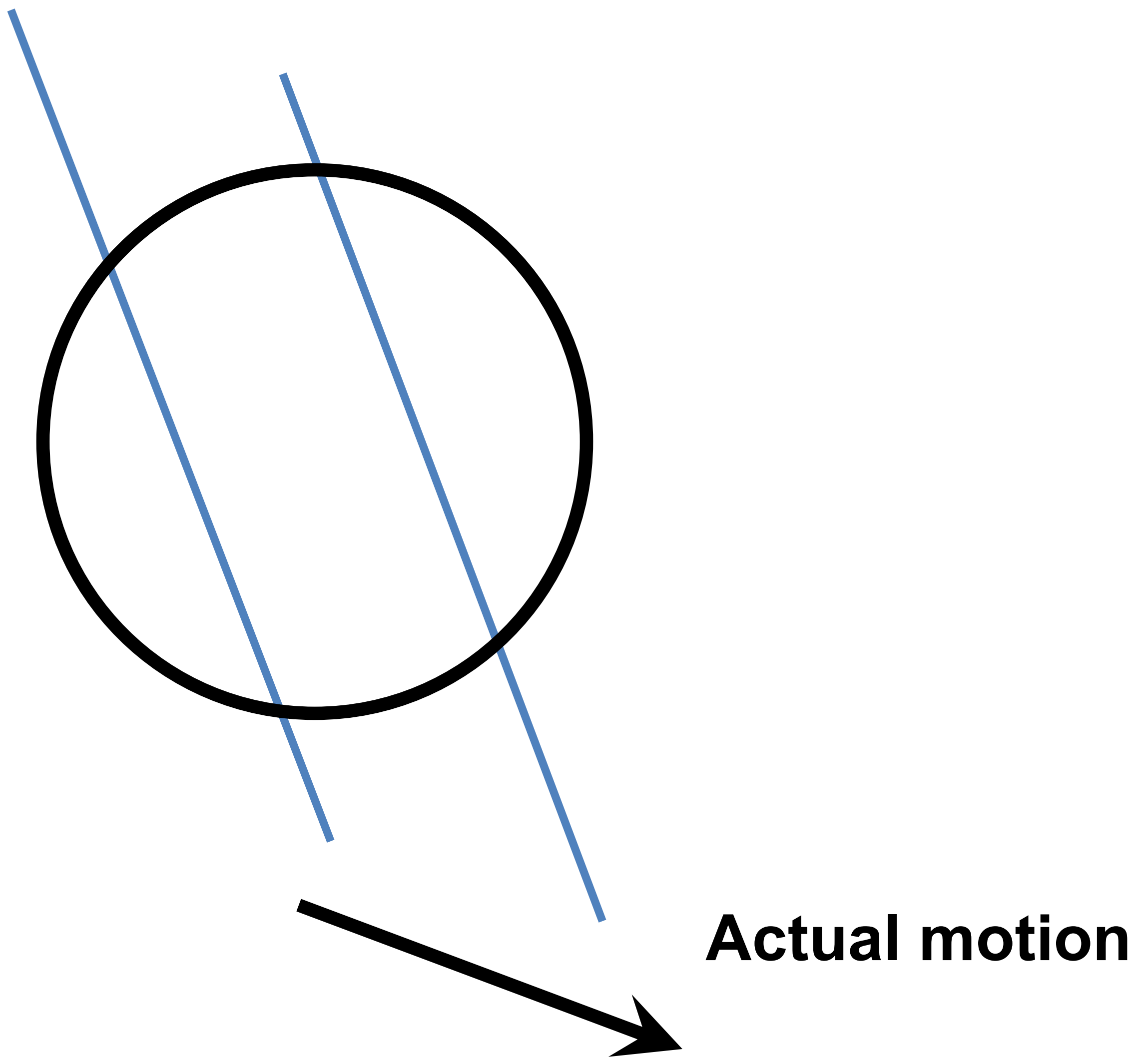
The component of the motion perpendicular to the gradient (i.e., parallel to the edge) cannot be measured

If (u, v) satisfies the equation, so does ($u+u'$, $v+v'$) if

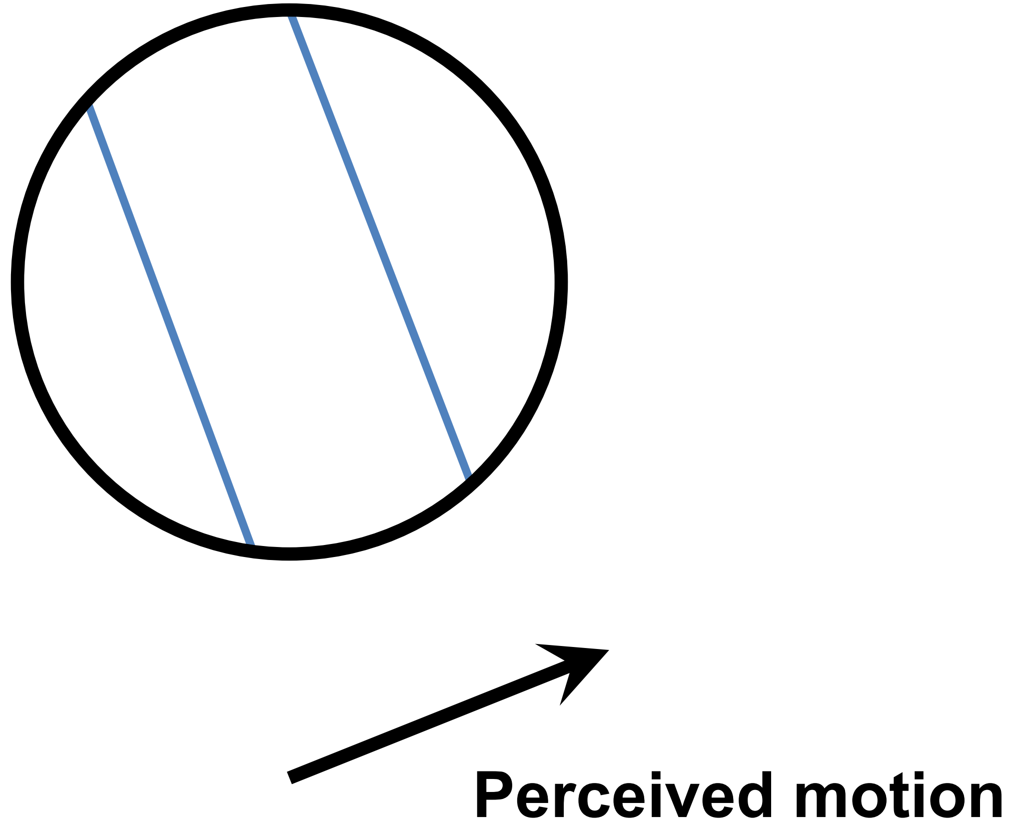
$$\nabla I \cdot [u' \;\; v']^T = 0$$



The aperture problem



The aperture problem



The barber pole illusion

http://en.wikipedia.org/wiki/Barberpole_illusion

The barber pole illusion

http://en.wikipedia.org/wiki/Barberpole_illusion

Solving the ambiguity…

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

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

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

$$\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix}$$

Solving the ambiguity…

- Least squares problem:

$$\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix} \qquad \underset{25\text{x}2}{A} \; \underset{2\text{x}1}{d} = \underset{25\text{x}1}{b}$$

Matching patches across images

- Overconstrained linear system

$$\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix}$$

$$\underset{25\text{x}2}{A} \quad \underset{2\text{x}1}{d} = \underset{25\text{x}1}{b}$$

Least squares solution for *d* given by $(A^T A)\ d = A^T b$

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

The summations are over all pixels in the K x K window

Conditions for solvability

Optimal (u, v) satisfies Lucas-Kanade equation

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

$$A^T A \qquad\qquad\qquad\qquad A^T b$$

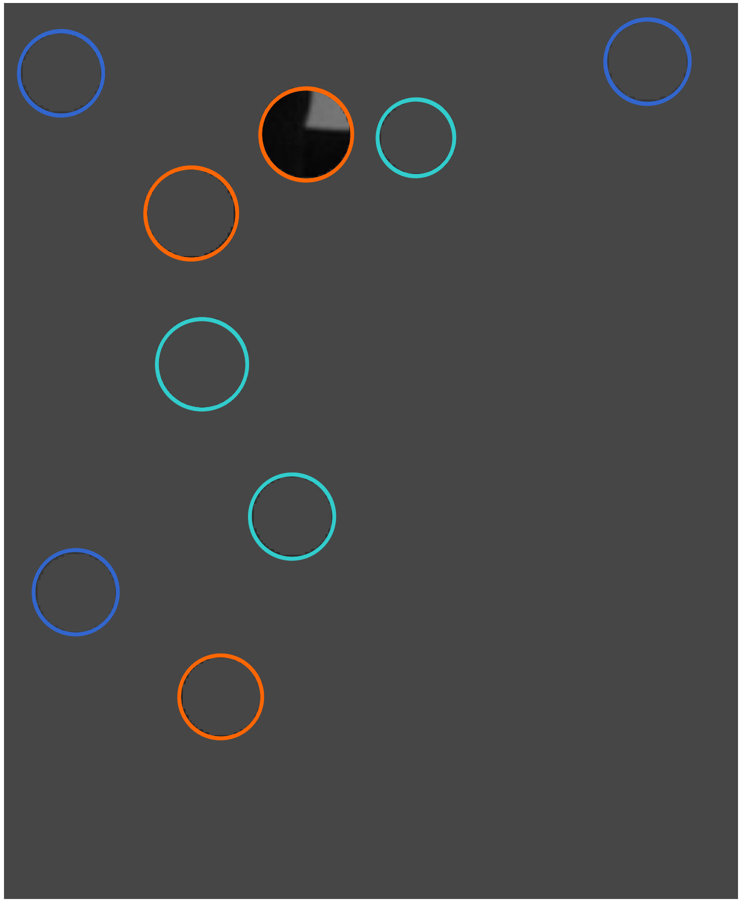
When is this solvable? I.e., what are good points to track?

- **$A^T A$** should be invertible
- **$A^T A$** should not be too small due to noise
  - eigenvalues λ_1 and λ_2 of **$A^T A$** should not be too small
- **$A^T A$** should be well-conditioned
  - λ_1 / λ_2 should not be too large (λ_1 = larger eigenvalue)

Does this remind you of anything?

Criteria for Harris corner detector

Aperture problem

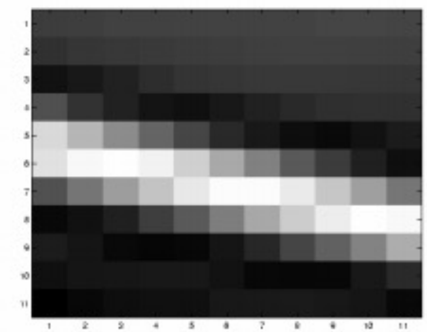


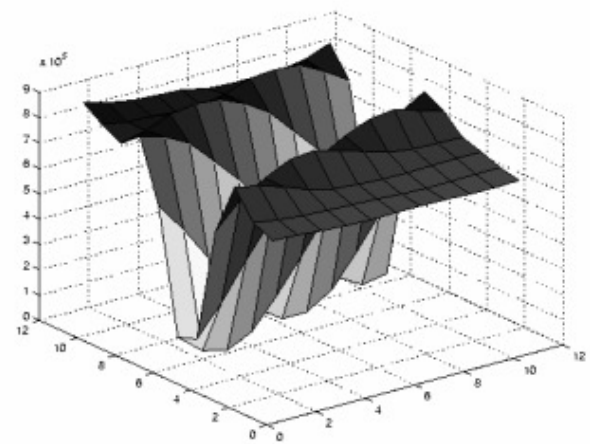
Corners

Lines

Flat regions

Edge



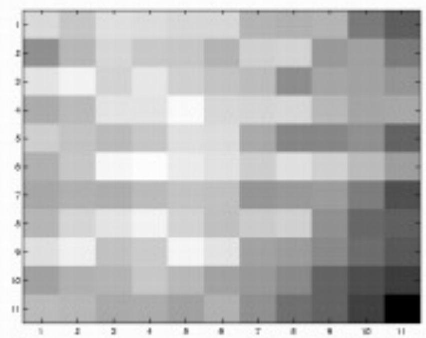


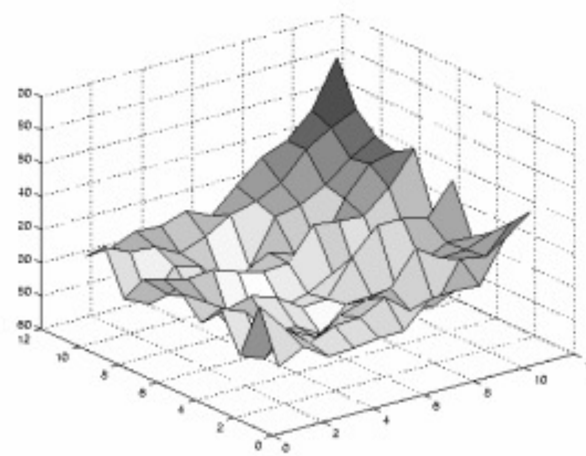
$$\sum \nabla I (\nabla I)^T$$

– large gradients, all the same

– large λ_1, small λ_2

Low Texture Region



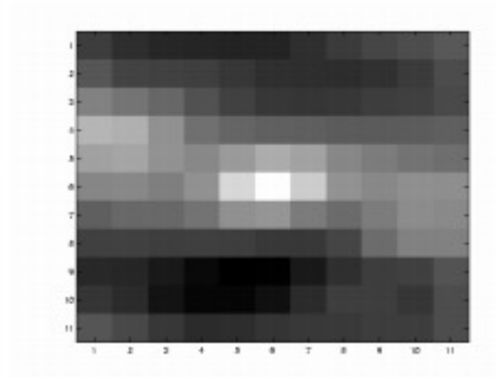


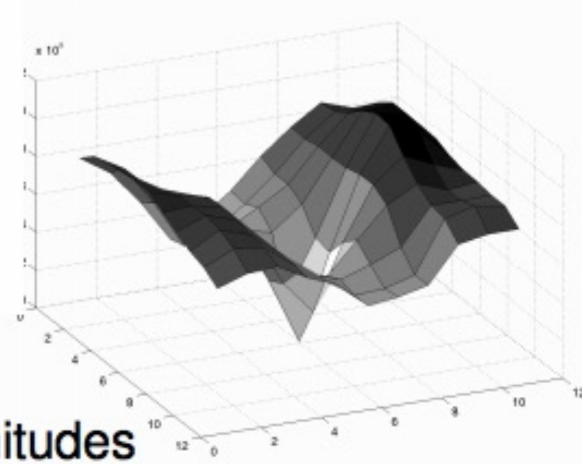
$$\sum \nabla I (\nabla I)^T$$

– gradients have small magnitude

– small λ_1, small λ_2

High Texture Region





$$\sum \nabla I (\nabla I)^T$$

– gradients are different, large magnitudes

– large λ_1, large λ_2

Errors in Lukas-Kanade

- What are the potential causes of errors in this procedure?
  - Suppose A^TA is easily invertible
  - Suppose there is not much noise in the image

When our assumptions are violated

- Brightness constancy is **not** satisfied
- The motion is **not** small
- A point does **not** move like its neighbors
  - window size is too large
  - what is the ideal window size?

Dealing with larger movements: Iterative refinement

Original (x,y) position

$I_t = I(x', y', t+1) - I(x, y, t)$

1. Initialize (x',y') = (x,y)
2. Compute (u,v) by

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

2nd moment matrix for feature patch in first image

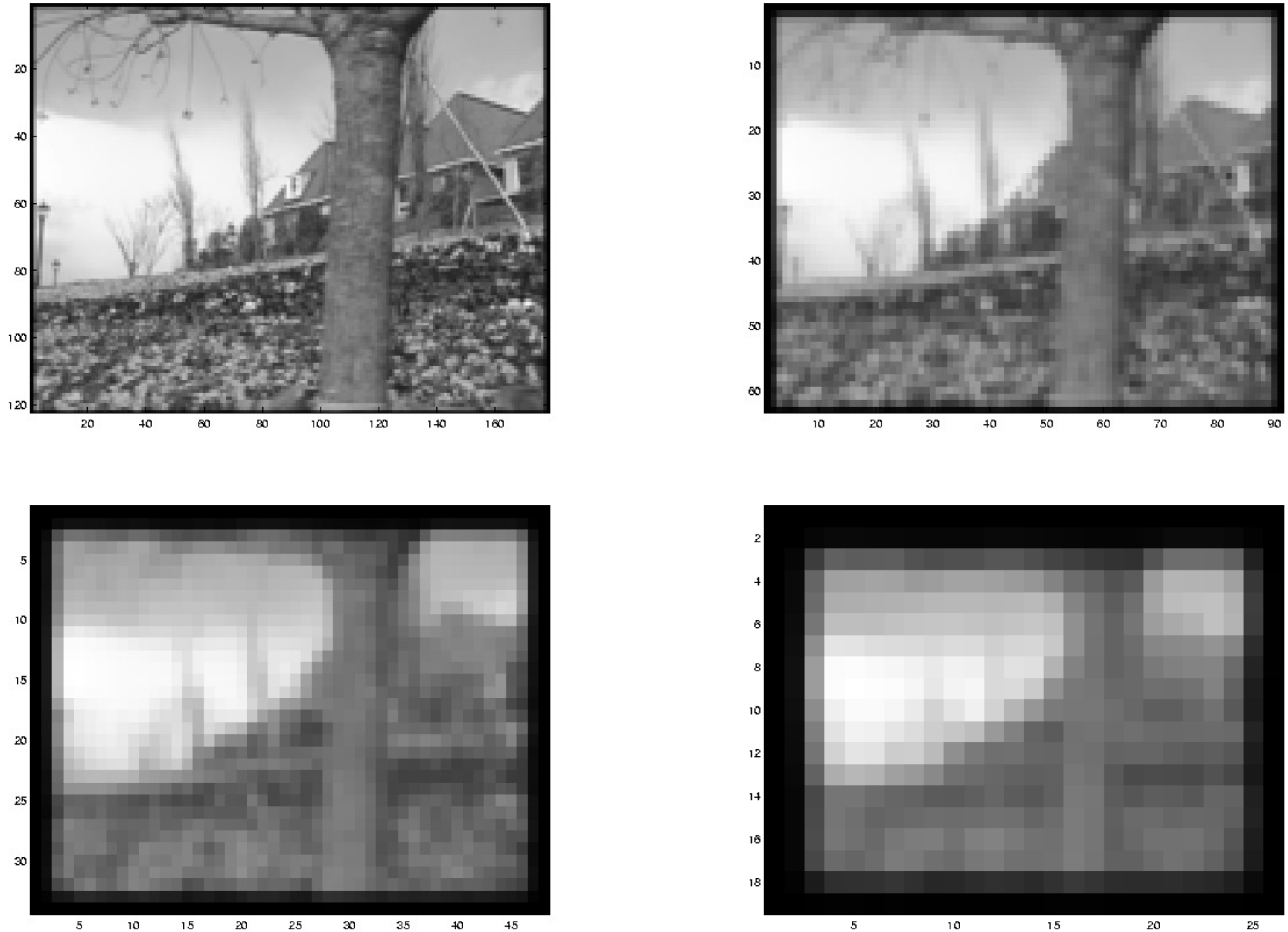
displacement

3. Shift window by (u, v): `x'=x'+u;  y'=y'+v;`
4. Recalculate I_t
5. Repeat steps 2-4 until small change
   - Use interpolation for subpixel values

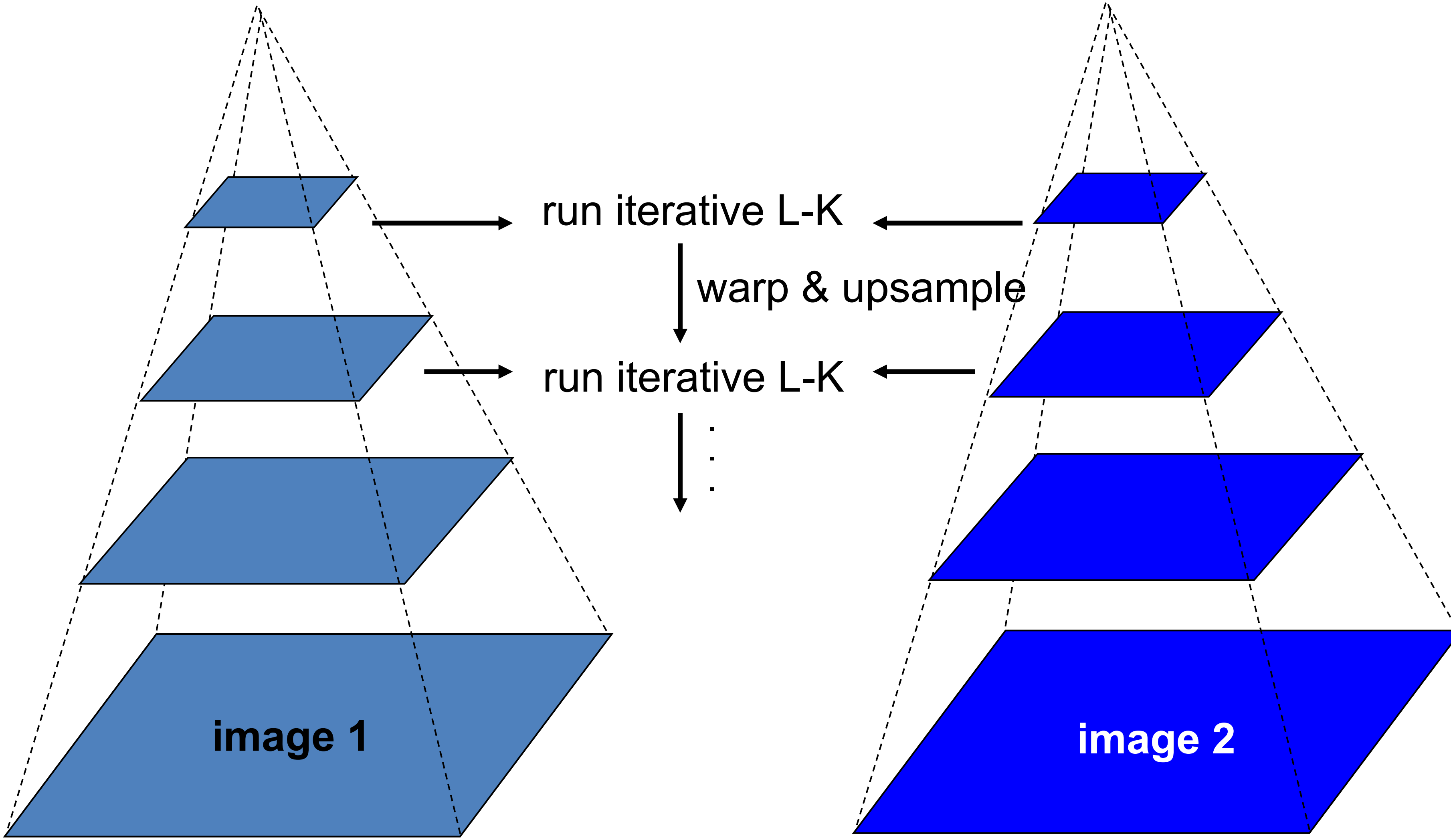
Revisiting the small motion assumption

- Is this motion small enough?
  - Probably not—it's much larger than one pixel (2^{nd} order terms dominate)
  - How might we solve this problem?

Reduce the resolution!



Coarse-to-fine optical flow estimation



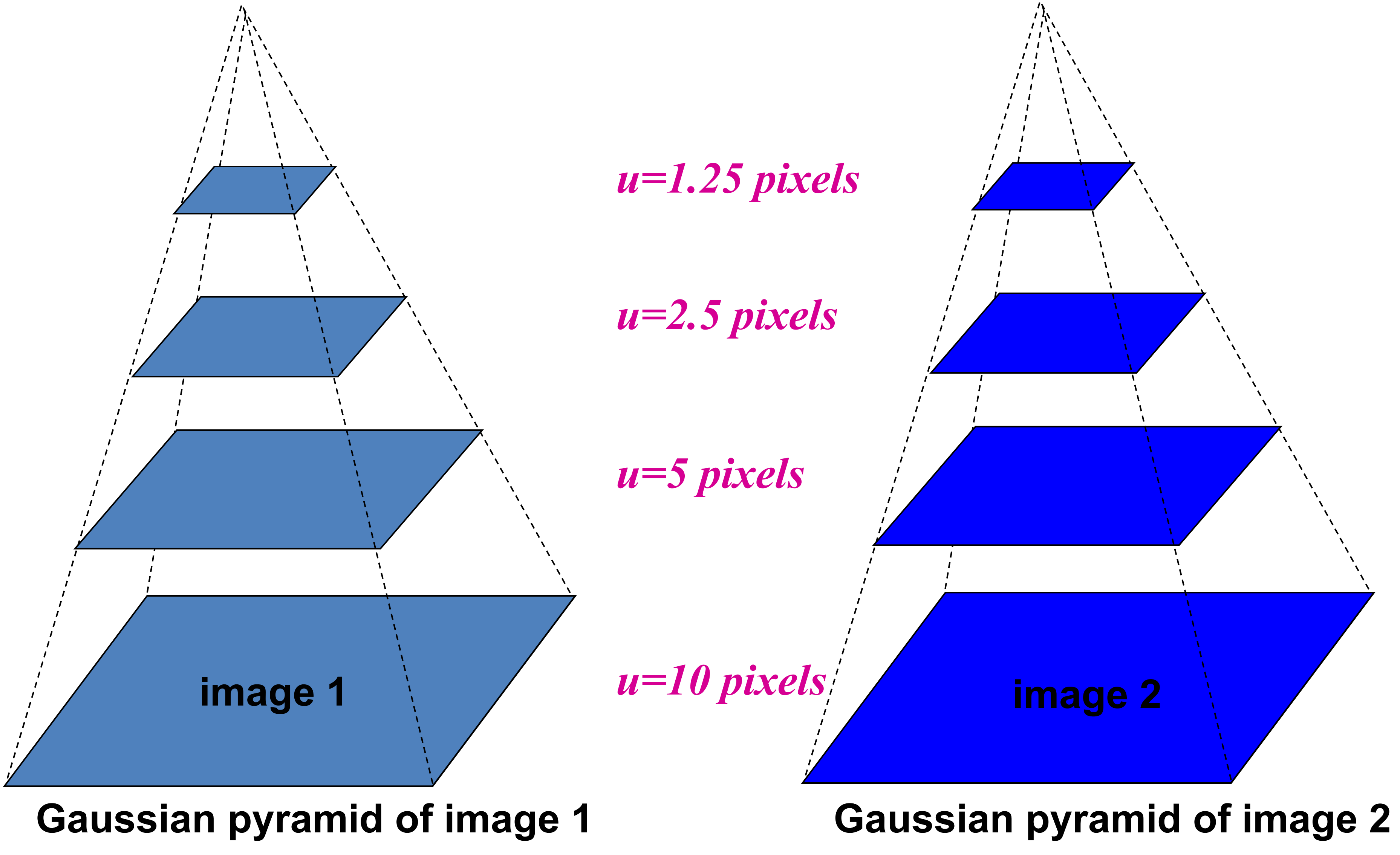
Gaussian pyramid of image 1 (t)

Gaussian pyramid of image 2 (t+1)

A Few Details

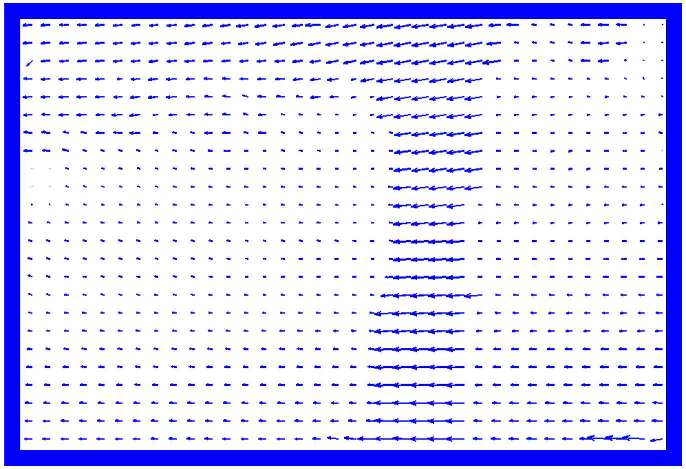
- Top Level
  - Apply L-K to get a flow field representing the flow from the first frame to the second frame.
  - Apply this flow field to warp the first frame toward the second frame.
  - Rerun L-K on the new warped image to get a flow field from it to the second frame.
  - Repeat till convergence.
- Next Level
  - Upsample the flow field to the next level as the first guess of the flow at that level.
  - Apply this flow field to warp the first frame toward the second frame.
  - Rerun L-K and warping till convergence as above.
- Etc.

Coarse-to-fine optical flow estimation

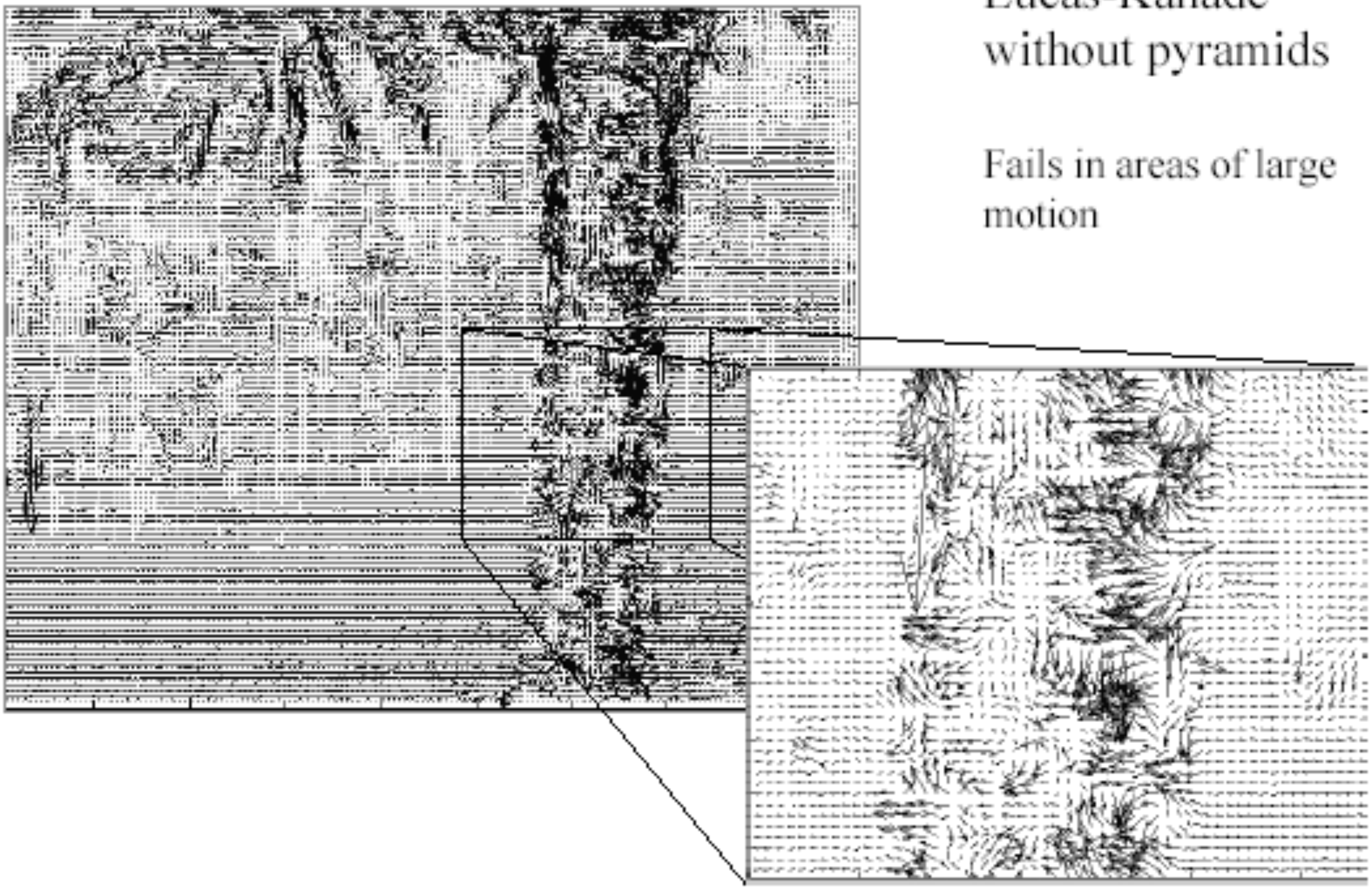


The Flower Garden Video

What should the optical flow be?



Optical Flow Results

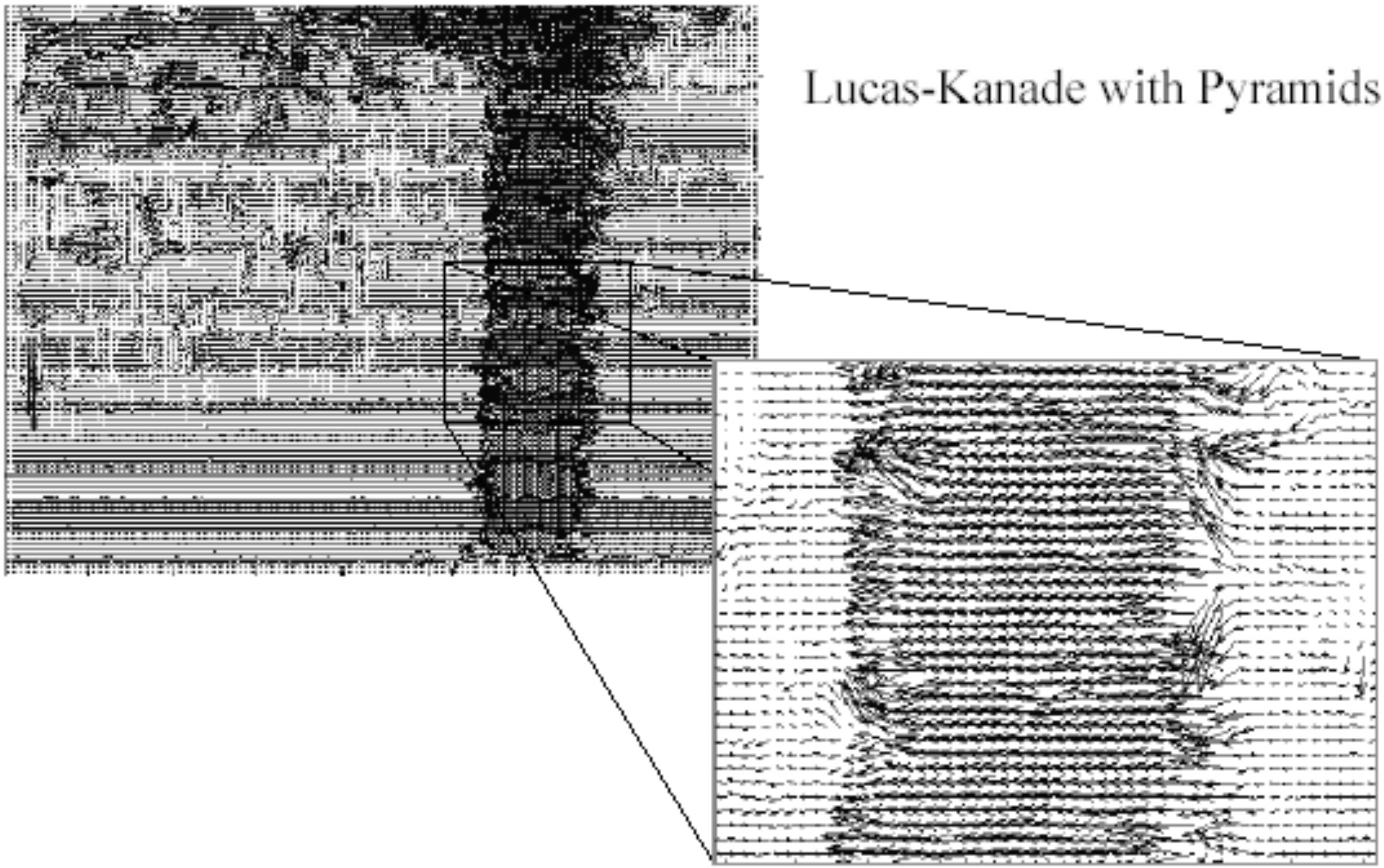


Lucas-Kanade without pyramids

Fails in areas of large motion

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Optical Flow Results



Lucas-Kanade with Pyramids

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

From 1D correspondence (stereo) to 2D correspondence problems (motion)

$$E(\mathbf{v}) = \sum_{p \in G} \underbrace{D_p(v_p)}_{\text{color-consistency}} + \sum_{\{p,q\} \in N} \underbrace{V(v_p, v_q)}_{\text{regularity}}$$

$$D_p(v_p) = (I_p^t - I_{p+v_p}^{t+1})^2$$

$$V(v_p, v_q) = w \cdot \|v_p - v_q\|^2$$

Horn-Schunck 1981
optical flow regularization
- 2nd order optimization (pseudo Newton)
- Rox/Cox/Ishikawa's method only works for scalar-valued variables

optical flow
$\mathbf{v} = \{v_p\}$

if 3D scene is NOT stationary
motion is **vector field** with **arbitrary directions**
(no epipolar line constraints)

SOCIETY OF ROBOTS

more difficult problem

need 2D shift vectors v_p

(no epipolar line constraint)

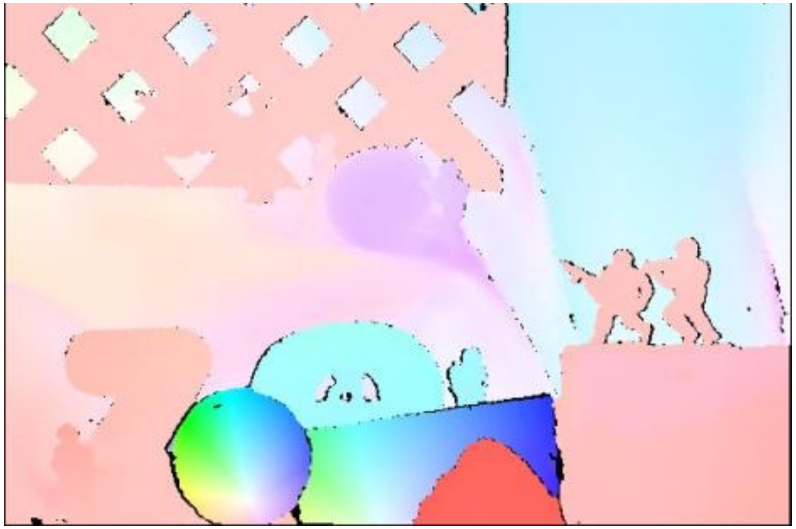
Horn-Schunck Optical Flow (1981)	**Lucas-Kanade Optical Flow (1981)**
brightness constancy	method of differences
small motion	
'smooth' flow (flow can vary from pixel to pixel)	**'constant' flow** (flow is constant for all pixels)
global method	local method

Flow quality evaluation

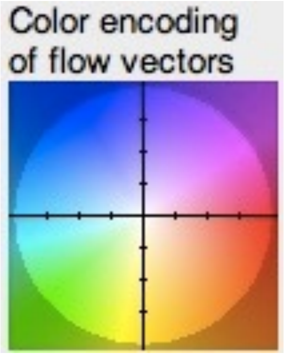
Flow quality evaluation

Flow quality evaluation

- Middlebury flow page
  - http://vision.middlebury.edu/flow/

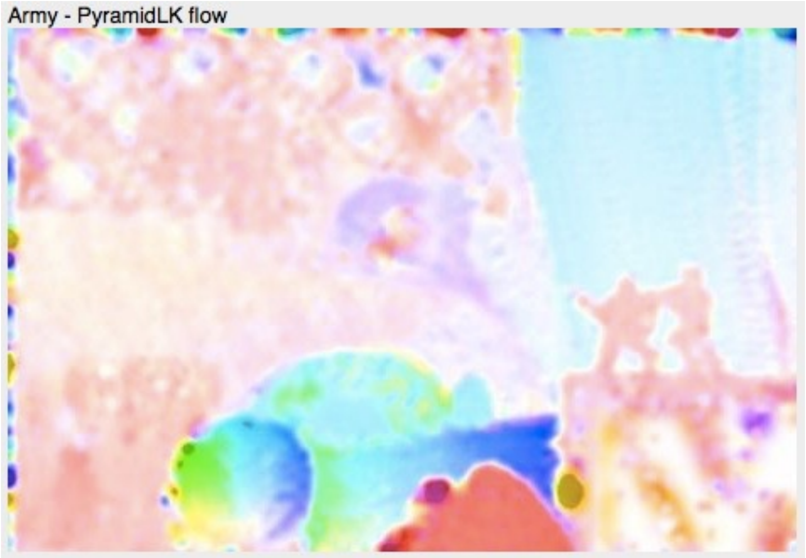


Ground Truth

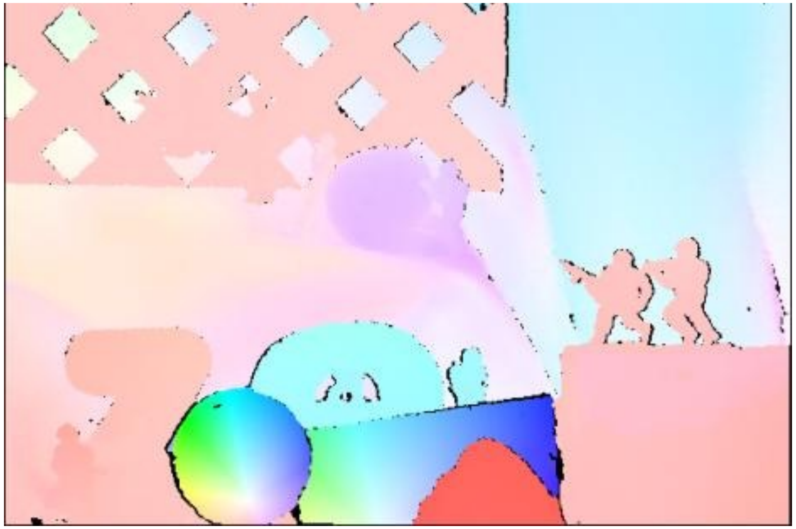


Flow quality evaluation

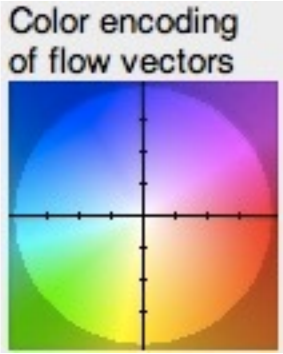
- Middlebury flow page
  - http://vision.middlebury.edu/flow/



Lucas-Kanade flow



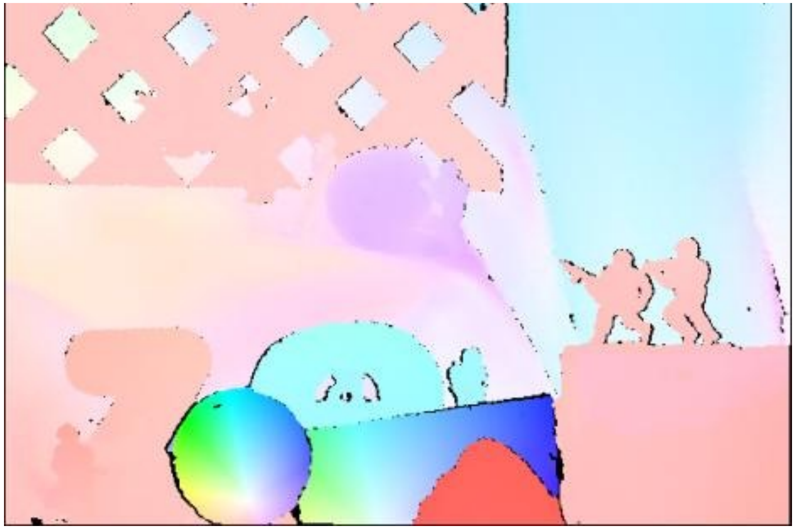
Ground Truth



Flow quality evaluation

- Middlebury flow page
  - http://vision.middlebury.edu/flow/

Best-in-class alg



Ground Truth