lecture 8

image registration

Sept. 30, 2020

Reminder: lecture recording

Example of Stereo Image Pair (the Pentagon)

$I(x, y)$

$J(x, y)$

Two frames of video

Frames 70 and 71 of Matlab's   'traffic.mj2' video   (mj2 is 'motion jpeg 2000')

1D version of registration problem

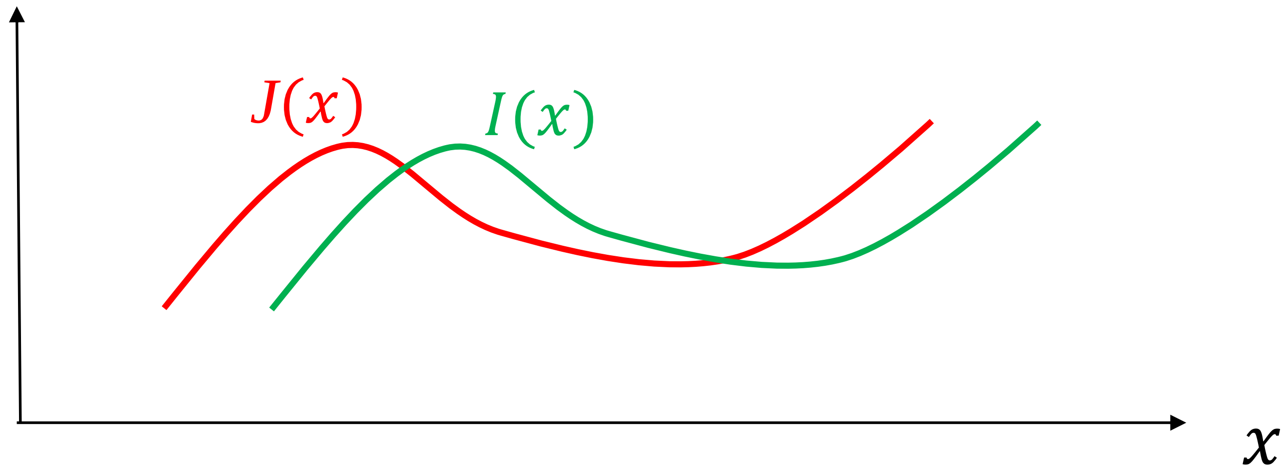
Given two 1D images, $I(x)$ and $J(x)$ and a point x_0, find the shift h that minimizes the following.

$$\sum_{x \in Ngd(x_0)} W(x - x_0)\{I(x+h) - J(x)\}^2$$

We solve this problem independently at each x_0.

$$\sum_{x \in Ngd(x_0)} W(x - x_0)\{I(x+h) - J(x)\}^2$$

If $h > 0$, then $I(x+h)$ shifts $I(x)$ to the left by h

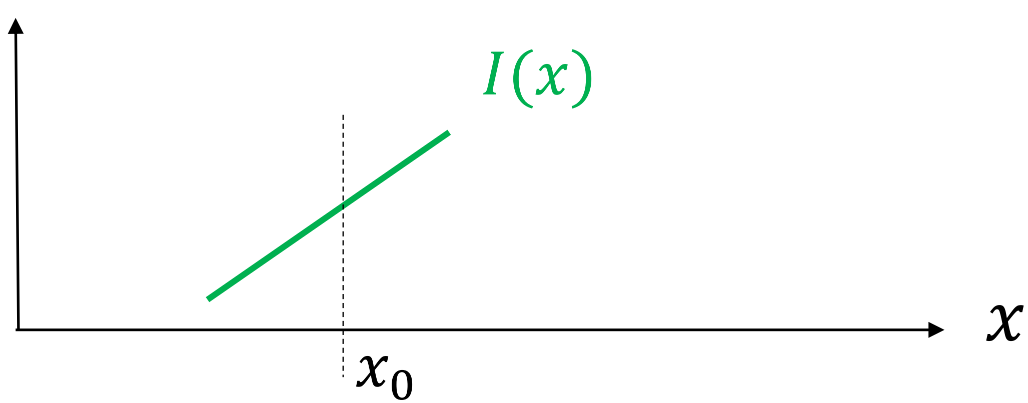


Suppose we have smoothed the two image frames by convolving with a small Gaussian (say $\sigma \approx 1$).

$$\sum_{x \in Ngd(x_0)} W(x - x_0)\{I(x+h) - J(x)\}^2$$

$$I(x+h) \approx I(x) + h\frac{dI}{dx}(x)$$

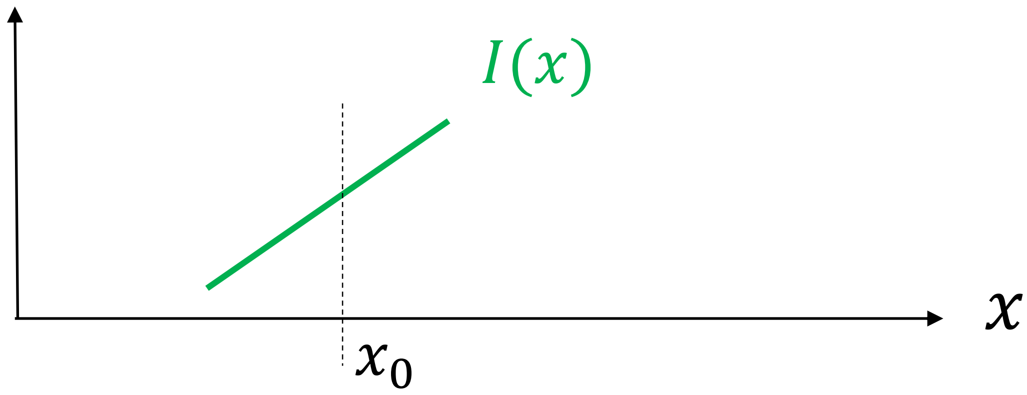
Then we can approximate $I(x+h)$ using a Taylor series expansion to first order.



Q: Are we assuming that $I(x)$ is linear over the whole neighborhood $Ngd(x_0)$?

$$\sum_{x \in Ngd(x_0)} W(x - x_0)\{I(x+h) - J(x)\}^2$$

$$I(x+h) \approx I(x) + h\frac{dI}{dx}(x)$$

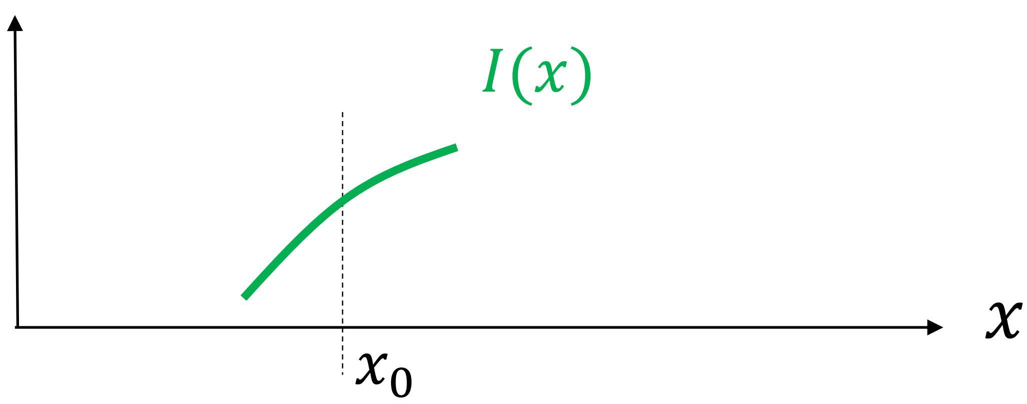


Q: Are we assuming that $I(x)$ is linear over the whole neighborhood $Ngd(x_0)$?

A: No. We might have a different linear approximation for different values of x in the neighborhood.

$$\sum_{x \in Ngd(x_0)} W(x - x_0)\{I(x+h) - J(x)\}^2$$

$$I(x+h) \approx I(x) + h\frac{dI}{dx}(x)$$



$$\sum_{x \in Ngd(x_0)} W(x - x_0)\{I(x+h) - J(x)\}^2$$

$$I(x+h) \approx I(x) + h\frac{dI}{dx}(x)$$

Now, find the shift h that minimizes :

$$\sum_{x \in Ngd(x_0)} W(x - x_0)(I(x) - J(x) + h\frac{dI(x)}{dx})^2$$

Find the shift h that minimizes :

$$\sum_{x \in Ngd(x_0)} W(x - x_0)(I(x) - J(x) + h\frac{dI(x)}{dx}\ )^2$$

This is another least squares problem. Take derivative with respect to h and set to 0:

$$h = \frac{\sum_{x \in Ngd(x_0)} W(x - x_0)(J(x) - I(x))\frac{dI(x)}{dx}}{\sum_{x \in Ngd(x_0)} W(x - x_0)(\frac{dI(x)}{dx}\ )^2}$$

What are assumptions?

- Smoothness of *both* images. Why ?
- Shift h is small. Why ?
- Shift h varies slowly. Why ?

What are assumptions?

- Underlying image has been smoothed enough so we can reliably estimate $\frac{dI}{dx}$ at each x. We smooth $J(x)$ also because we are comparing it with $I(x)$.

- The shift h is small, so that a first order Taylor approximation is likely to be valid.

- The model assumes $I(x+h)$ and $J(x)$ correspond for a single value of h in the whole neighborhood of x_0.

Iterative method

- The assumptions on previous slide will typically only hold approximately. Therefore, the estimate we obtain is only an approximation.

- We can iterate this method to try to get a better solution. How ?

Iterative Method

Given an initial estimate h_0 of the shift, we shift the image $I(x) \rightarrow I(x+h_0)$. We then solve the problem again by comparing $I(x+h_0)$ and $J(x)$, and so on.

We don't actually shift the image. We just need to interpolate to know $I(x+h_0)$ and $\frac{dI(x+h_0)}{dx}$.

[Details on interpolation omitted.]

Iterative Method

$$h = -\frac{\sum_{x \in Ngd(x_0)} W(x - x_0)(I(x + h_k) - J(x))\frac{dI(x+h_k)}{dx}}{\sum_{x \in Ngd(x_0)} W(x - x_0)(\frac{dI(x+h_k)}{dx})^2}$$

Given the k-th estimate of the shift, we can update to get the (k +1)-th estimate.

$$h_{k+1} \leftarrow h_k + h$$

Is this guaranteed to converge to the correct solution ?

2D image registration

$$\sum_{(x,y)\in Ngd(x_0,y_0)} \{I(x+h_x, y+h_y) - J(x,y)\}^2$$

$$I(x+h_x, y+h_y) \approx I(x,\ y) + \frac{\partial I}{\partial x}h_x + \frac{\partial I}{\partial y}h_y$$

We would have a weighting function too.

$$\underbrace{\sum_{(x,y)\in Ngd(x_0,y_0)} W(x-x_0, y-y_0)\{I(x,y) - J(x,y) + \frac{\partial I}{\partial x}h_x + \frac{\partial I}{\partial y}h_y\}^2}$$

Gather these expressions for different (x, y) and write as below:

$$\begin{bmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \\ \vdots & \vdots \end{bmatrix}_{N\times 2} \begin{bmatrix} h_x \\ h_y \end{bmatrix}_{2\times 1} - \begin{bmatrix} J(x,y) - I(x,y) \\ \cdots \end{bmatrix}_{N\times 1}$$

What is N ?

$$\boldsymbol{A}\,\boldsymbol{u}\ -\boldsymbol{b}$$

Find the $\boldsymbol{u} = (h_x, h_y)$ that minimizes:

$$(\boldsymbol{A}\,\boldsymbol{u} - \boldsymbol{b})^T \boldsymbol{W}\,(\boldsymbol{A}\,\boldsymbol{u} - \boldsymbol{b})$$

$$\sum_{(x,y)\in Ngd(x_0,y_0)} W(x-x_0, y-y_0)\{I(x,y) - J(x,y) + \frac{\partial I}{\partial x}h_x + \frac{\partial I}{\partial y}h_y\}^2$$

Gather these expressions for different (x, y) and write as below:

$$\begin{bmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \\ \vdots & \vdots \end{bmatrix}_{N\times 2} \begin{bmatrix} h_x \\ h_y \end{bmatrix}_{2\times 1} - \begin{bmatrix} J(x,y) - I(x,y) \\ \cdots \end{bmatrix}_{N\times 1}$$

$$\boldsymbol{A}\,\boldsymbol{u} - \boldsymbol{b}$$

Find the $\boldsymbol{u}$ = (h_x, h_y) that minimizes:

$$(\boldsymbol{A}\,\boldsymbol{u} - \boldsymbol{b})^T \boldsymbol{W}\,(\boldsymbol{A}\,\boldsymbol{u} - \boldsymbol{b})$$

This is just a least squares problem. We solve for $\boldsymbol{u}$:

$$\underbrace{\mathbf{A}^{\mathrm{T}}\mathbf{W}\mathbf{A}}_{2\,\times\,2}\mathbf{u} = \underbrace{\mathbf{A}^{\mathrm{T}}\mathbf{W}\mathbf{b}}_{2\,\times\,1}$$

$$\left[\begin{array}{cc} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \\ \vdots & \vdots \end{array}\right]_{N\times 2} \left[\begin{array}{c} h_x \\ h_y \end{array}\right]_{2\times 1} - \left[\begin{array}{c} J(x,y) - I(x,y) \\ \dots \end{array}\right]_{N\times 1}$$

$$\boldsymbol{A}\,\boldsymbol{u} - \boldsymbol{b}$$

Solve for $\boldsymbol{u} = (h_x, h_y)$:

$$\underbrace{\mathbf{A}^{\mathrm{T}}\mathbf{W}\mathbf{A}}_{2 \times 2}\mathbf{u} = \underbrace{\mathbf{A}^{\mathrm{T}}\mathbf{W}\mathbf{b}}_{2 \times 1}$$

By inspection, this linear system can be written as follows:

$$\begin{bmatrix} \sum_{x,y} W(\cdot)(\frac{\partial I}{\partial x})^2 & \sum_{x,y} W(\cdot)(\frac{\partial I}{\partial x})(\frac{\partial I}{\partial y}) \\ \sum_{x,y} W(\cdot)(\frac{\partial I}{\partial x})(\frac{\partial I}{\partial y}) & \sum_{x,y} W(\cdot)(\frac{\partial I}{\partial y})^2 \end{bmatrix} \begin{bmatrix} h_x \\ h_y \end{bmatrix} = \begin{bmatrix} \sum_{x,y} W(\cdot)(J(x,y) - I(x,y))\frac{\partial I}{\partial x} \\ \sum_{x,y} W(\cdot)(J(x,y) - I(x,y))\frac{\partial I}{\partial y} \end{bmatrix}$$

Exercise: When we can solve for $\boldsymbol{u}$?

Solve for $\boldsymbol{u}$ = (h_x, h_y):

$$\underbrace{\mathbf{A^T W A}}_{2 \times 2}\mathbf{u} = \underbrace{\mathbf{A^T W b}}_{2 \times 1}$$

$$\begin{bmatrix} \sum_{x,y} W(\cdot)(\frac{\partial I}{\partial x})^2 & \sum_{x,y} W(\cdot)(\frac{\partial I}{\partial x})(\frac{\partial I}{\partial y}) \\ \sum_{x,y} W(\cdot)(\frac{\partial I}{\partial x})(\frac{\partial I}{\partial y}) & \sum_{x,y} W(\cdot)(\frac{\partial I}{\partial y})^2 \end{bmatrix} \begin{bmatrix} h_x \\ h_y \end{bmatrix} = \begin{bmatrix} \sum_{x,y} W(\cdot)(J(x,y) - I(x,y))\frac{\partial I}{\partial x} \\ \sum_{x,y} W(\cdot)(J(x,y) - I(x,y))\frac{\partial I}{\partial y} \end{bmatrix}$$

This is the 2^{nd} moment matrix from last lecture.

We can solve for u when the second moment matrix is invertible (equivalently, when its eigenvalues are both sufficiently greater than 0).

Iterative Method

Given the kth estimate of the shift $\boldsymbol{u}$ = (h_x, h_y) , we can update to get the (k+1)th estimate.

$$h_x^{k+1} \leftarrow h_x^k + h_x$$

$$h_y^{k+1} \leftarrow h_y^k + h_y$$

A few related problems…

1. "Optical flow" (Lucas and Kanade 1981)

2. "KLT" tracking (Kanade, Lucas, Tomasi, 1991)

3. Tracking using a more general motion model
   (Shi & Tomasi, 1994)

1. Optical Flow

Suppose we have a video $I(x, y, t)$ where t is the frame index.

We can compare frames $I(x, y, t)$ and $I(x, y, t + 1)$.

$$I(x + h_x, y + h_y, t + 1) \approx I(x, y, t) + \frac{\partial I}{\partial x} h_x + \frac{\partial I}{\partial y} h_y + \frac{\partial I}{\partial t}$$

What does this mean?

How would you estimate it ?

1. Optical Flow

Suppose we have a video $I(x, y, t)$ where t is the frame index.

We can compare frames $I(x, y, t)$ and $I(x, y, t + 1)$.

$$I(x + h_x, y + h_y, t + 1) \approx I(x, y, t) + \frac{\partial I}{\partial x} h_x + \frac{\partial I}{\partial y} h_y + \frac{\partial I}{\partial t}$$

$$\frac{\partial I(x, y, t)}{\partial t} \approx I(x, y, t + 1) - I(x, y, t)$$

1. Optical Flow (Lucas-Kanade)

General Image Registration problem:

$$\left[\begin{array}{cc} \sum_{x,y} W(\cdot)(\frac{\partial I}{\partial x})^2 & \sum_{x,y} W(\cdot)(\frac{\partial I}{\partial x})(\frac{\partial I}{\partial y}) \\ \sum_{x,y} W(\cdot)(\frac{\partial I}{\partial x})(\frac{\partial I}{\partial y}) & \sum_{x,y} W(\cdot)(\frac{\partial I}{\partial y})^2 \end{array}\right] \left[\begin{array}{c} h_x \\ h_y \end{array}\right] = \left[\begin{array}{c} \sum_{x,y} W(\cdot)(J(x,y) - I(x,y))\frac{\partial I}{\partial x} \\ \sum_{x,y} W(\cdot)(J(x,y) - I(x,y))\frac{\partial I}{\partial y} \end{array}\right]$$

Lucas-Kanade optical flow

$$\left[\begin{array}{cc} \sum_{x,y} W(\cdot)(\frac{\partial I}{\partial x})^2 & \sum_{x,y} W(\cdot)(\frac{\partial I}{\partial x})(\frac{\partial I}{\partial y}) \\ \sum_{x,y} W(\cdot)(\frac{\partial I}{\partial x})(\frac{\partial I}{\partial y}) & \sum_{x,y} W(\cdot)(\frac{\partial I}{\partial y})^2 \end{array}\right] \left[\begin{array}{c} h_x \\ h_y \end{array}\right] = -\left[\begin{array}{c} \sum_{x,y} W(\cdot)(\frac{\partial I}{\partial t})\frac{\partial I}{\partial x} \\ \sum_{x,y} W(\cdot)(\frac{\partial I}{\partial t})\frac{\partial I}{\partial y} \end{array}\right]$$

1. Optical Flow (Lucas-Kanade)

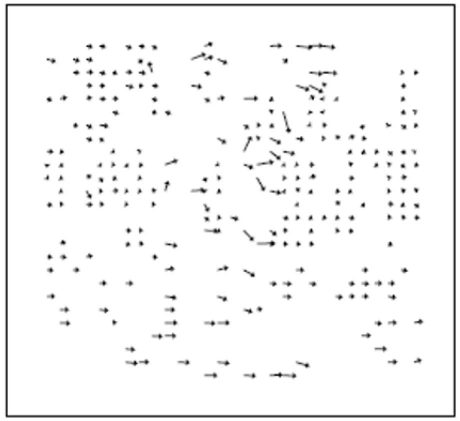
The goal is to compute a motion vector (v_x, v_y) for each position (x, y), preferably from two (or a small number) of frames. The vector field is called the *"motion field".*

What are the challenging cases ? When are the estimates unreliable ?

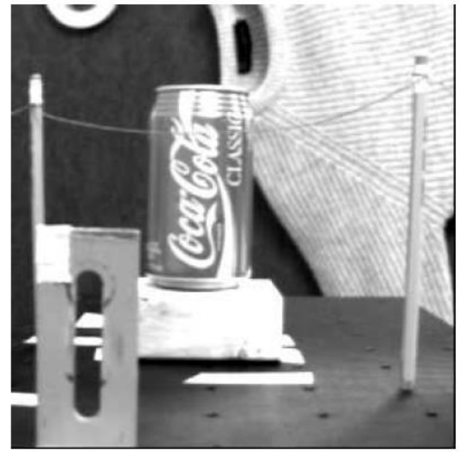
Dozens of optical flow methods have been proposed. In 1990's, the accuracy of these methods were compared using a *small* number of standard data sets, which illustrated challenging cases such as "occlusions" (object passing in front of another) and "textureless regions".

The first such comparison was by [Barron, Fleet, Beauchemin 1994].

(a) SRI Trees

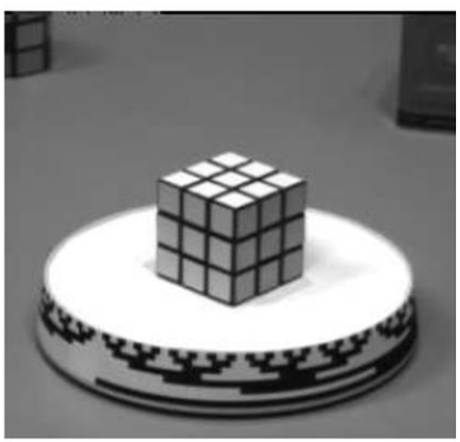


(a) SRI Trees

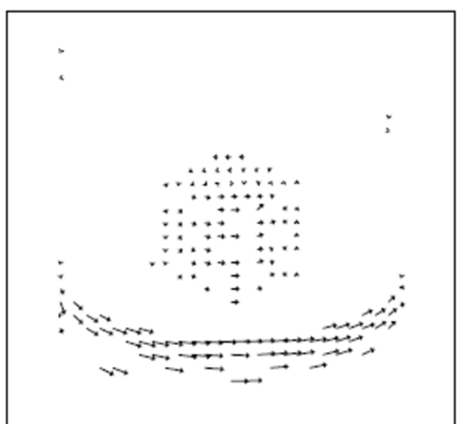


(b) NASA Sequence

(b) NASA Sequence

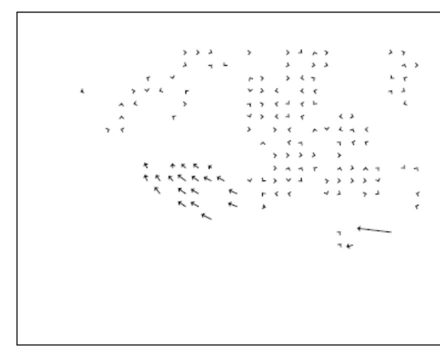


(c) Rubik Cube



(c) Rubik Cube

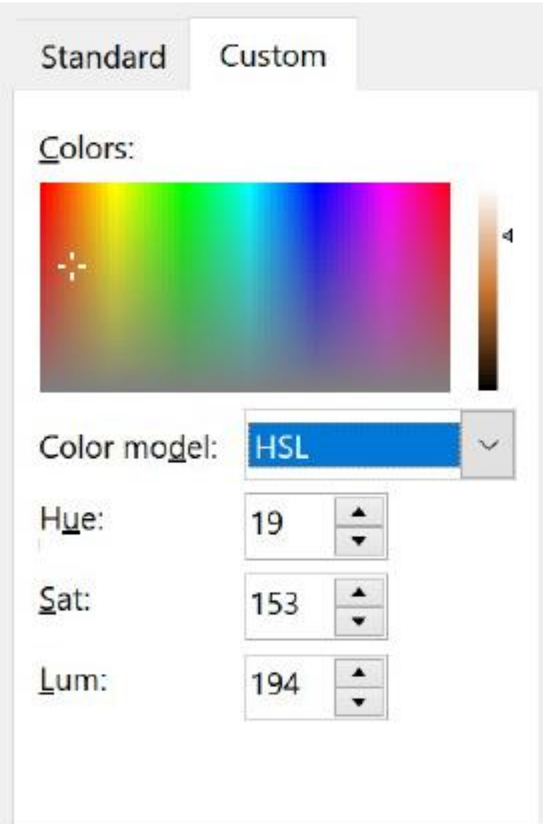
(d) Hamburg Taxi



(d) Hamburg Taxi

Color coding optical flow for better visualization

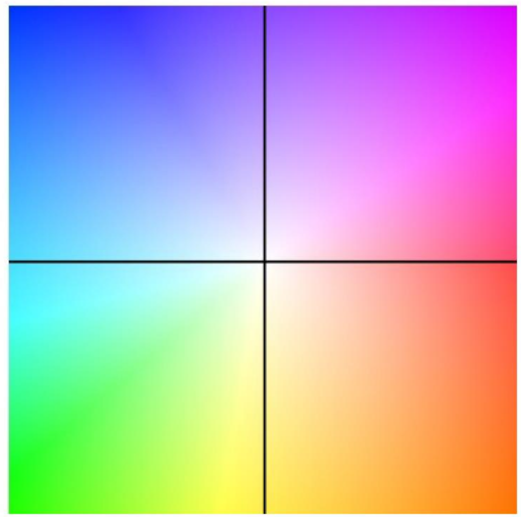
(started in 2000's)

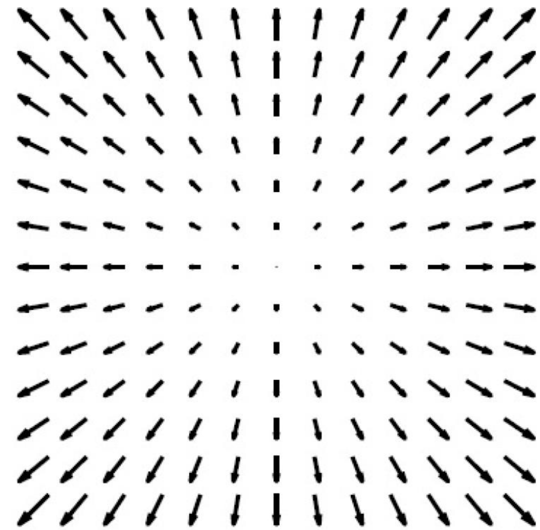


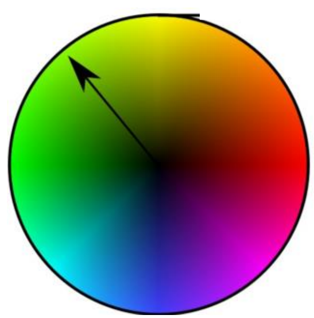
Color code for motion vectors (v_x, v_y):

Hue encodes motion direction.

Saturation encodes speed.







Alternatively, the speed could be encoded by luminance as shown here.

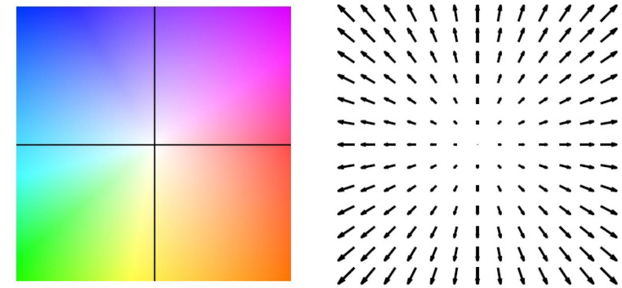
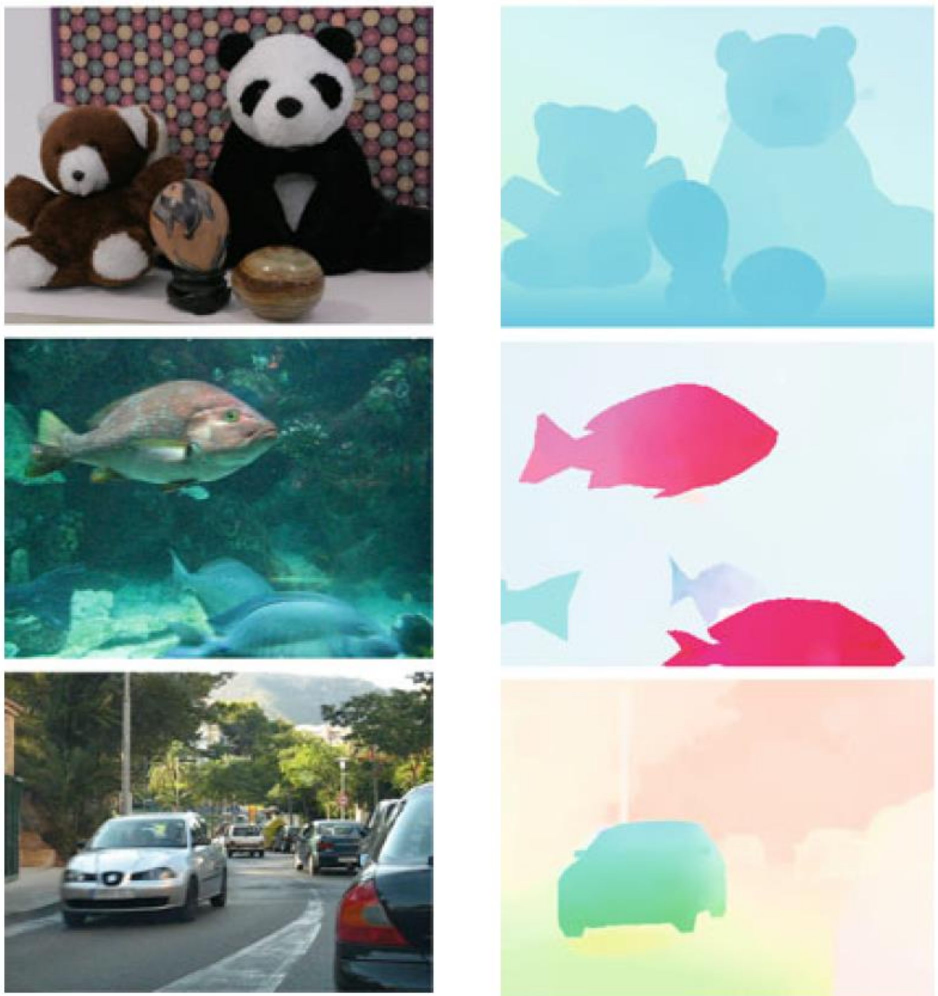


Image frame color coded motion field



What are the motion fields ?

What is happening in the 3D scene to produce this motion field?
(This is more relevant to second half of course.)

[Sun, Roth, Black 2014]

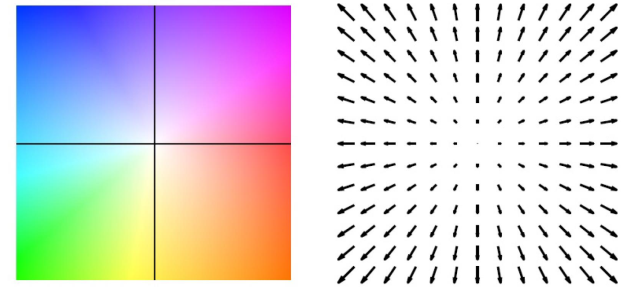
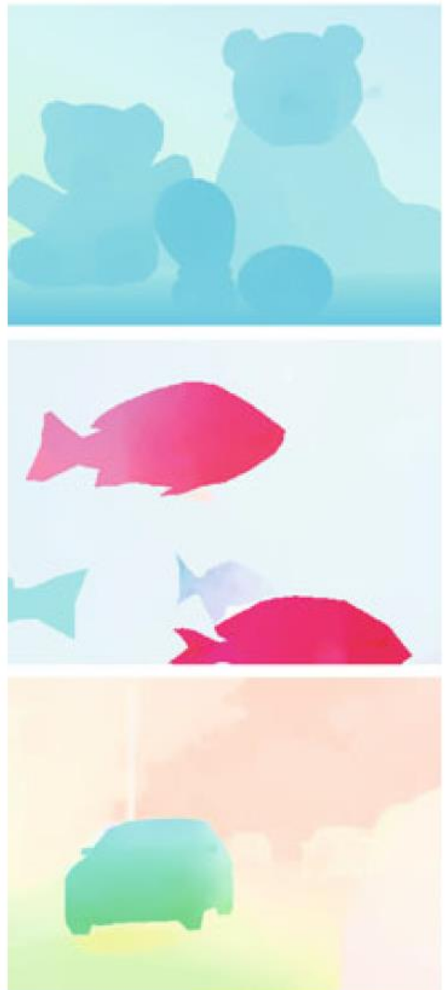


Image frame

color coded motion field



Camera moving laterally.
Objects static.

Objects (fish) moving.
Background static.
Camera static.

Objects (cars) moving.
Camera moving.

[Sun, Roth, Black 2014]

2. "KLT" tracking (Kanade, Lucas, Tomasi, 1991)

Track locally distinctive points over *multiple* frames.

This would require we discuss separate issues such as when to abandon the track. E.g. Do we compare $(k+1)^{th}$ frame to k^{th} frame or to 1^{st} frame ?

Which points on the cars would be tracked?

Note in a street scene surveillance video, most points are not moving.

Example

https://nanonets.com/blog/optical-flow/

3. Tracking using a more general motion model (Shi & Tomasi, 1994)

$$I(\mathbf{x} + \mathbf{h} + \mathbf{D}(\mathbf{x} - \mathbf{x_0})) = J(\mathbf{x})$$

translation

$$\begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}$$

scale + shear + rotation

$$I(\mathbf{x} + \mathbf{h} + \mathbf{D}(\mathbf{x} - \mathbf{x_0})) = J(\mathbf{x})$$

Any matrix $\mathbf{D}$ can always be written as a product of matrices below. (More details on this later in course.)

$$\left[\begin{array}{cc} D_{11} & D_{12} \\ D_{21} & D_{22} \end{array} \right]$$

$$\left[\begin{array}{cc} s_x & 0 \\ 0 & 1 \end{array} \right] \left[\begin{array}{cc} 1 & 0 \\ 0 & s_y \end{array} \right] \quad \left[\begin{array}{cc} 1 & a \\ 0 & 1 \end{array} \right] \left[\begin{array}{cc} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{array} \right]$$

scale (includes reflections if scale < 0)

shear

rotation

$$I(\mathbf{x} + \mathbf{h} + \mathbf{D}(\mathbf{x} - \mathbf{x_0})) = J(\mathbf{x})$$

The minimization problem now has six variables:
$(h_x, h_y, D_{11}, D_{12}, D_{21}, D_{22})$:

$$\sum_{\mathbf{x} \in Ngd(\mathbf{x_0})} W(\mathbf{x} - \mathbf{x_0})(I(\mathbf{x} + \mathbf{h} + \mathbf{D}(\mathbf{x} - \mathbf{x_0})) - J(\mathbf{x}))^2$$

Exercises: How would you set up a least squares problem to solve this?

Hint: take a first order Taylor series expansion.

3. *Tracking* using a more general motion model (Shi & Tomasi, 1994)

$$I(\mathbf{x} + \mathbf{h} + \mathbf{D}(\mathbf{x} - \mathbf{x_0})) = J(\mathbf{x})$$

To emphasize…. You typically only want to *track* a limited number of points.

This is different from *optical flow*, where you estimate the motion at all pixels and for two frames.

COMP 558 Overview

Part 1 : 2D Vision

RGB

Image filtering

Edge detection

Least Squares Estimation

Robust Estimation: Hough transform & RANSAC

Features 1: corners

Image Registration: the Lucas-Kanade method

Scale space

Histogram-based Tracking

Features 2: SIFT, HOG

Features 3: CNN's

Object classification and detection

Segmentation (TBD)

Part 2 : 3D Vision

Linear perspective, camera translation

Vanishing points, camera rotation

Homogeneous coordinates, camera intrinsics

Least Squares methods (eigenspaces, SVD)

Camera Calibration

Homographies & rectification

Stereo and Epipolar Geometry

Stereo correspondence

Cameras and Photography

RGBD Cameras