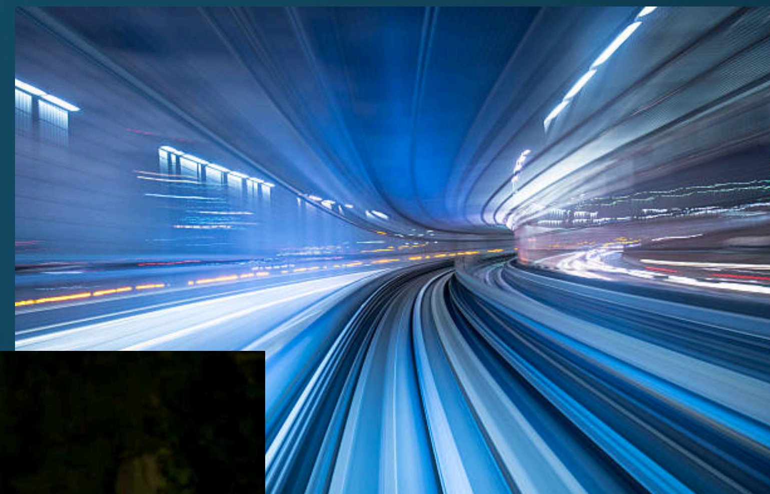
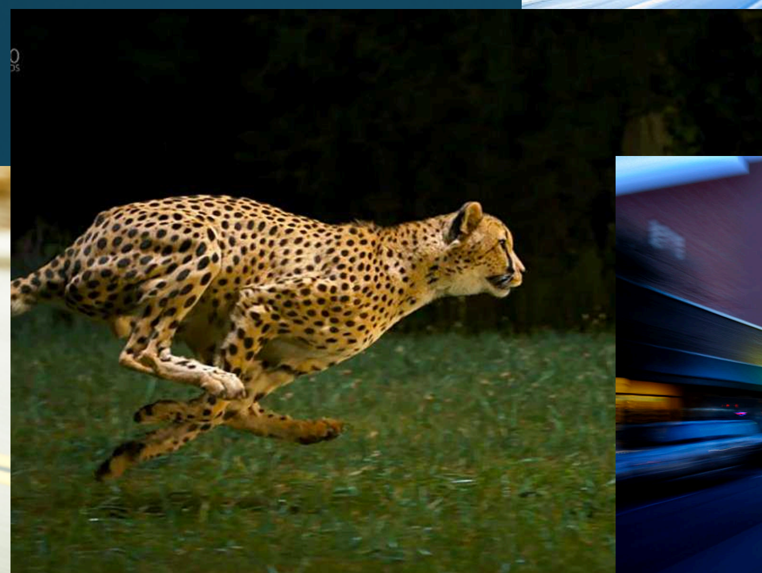
CSCI 4360/6360 Data Science II

Dense Motion Analysis

Motion analysis

- Our world is in motion



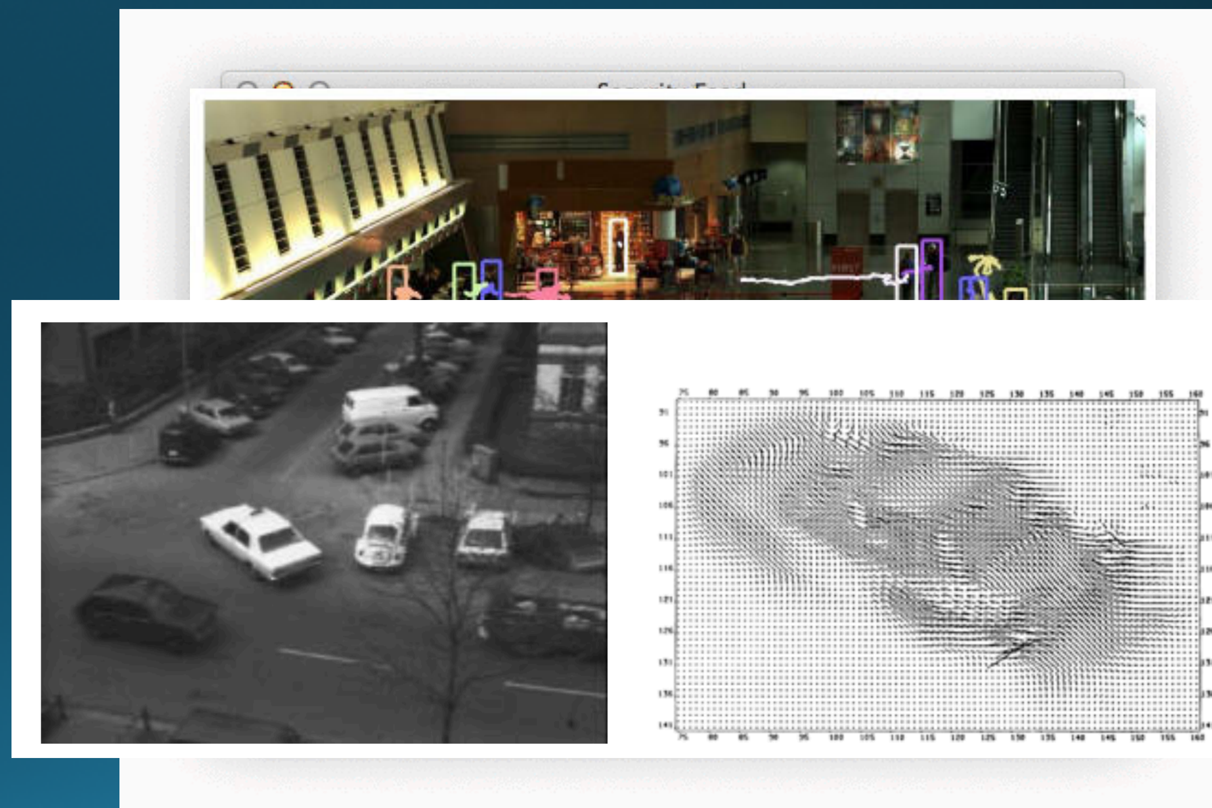



Motion analysis

- Core problem in computer vision

Motion analysis

- Object tracking
- Trajectory analysis
- Object finding
- Video enhancement, stabilization, 3D reconstruction, object recognition

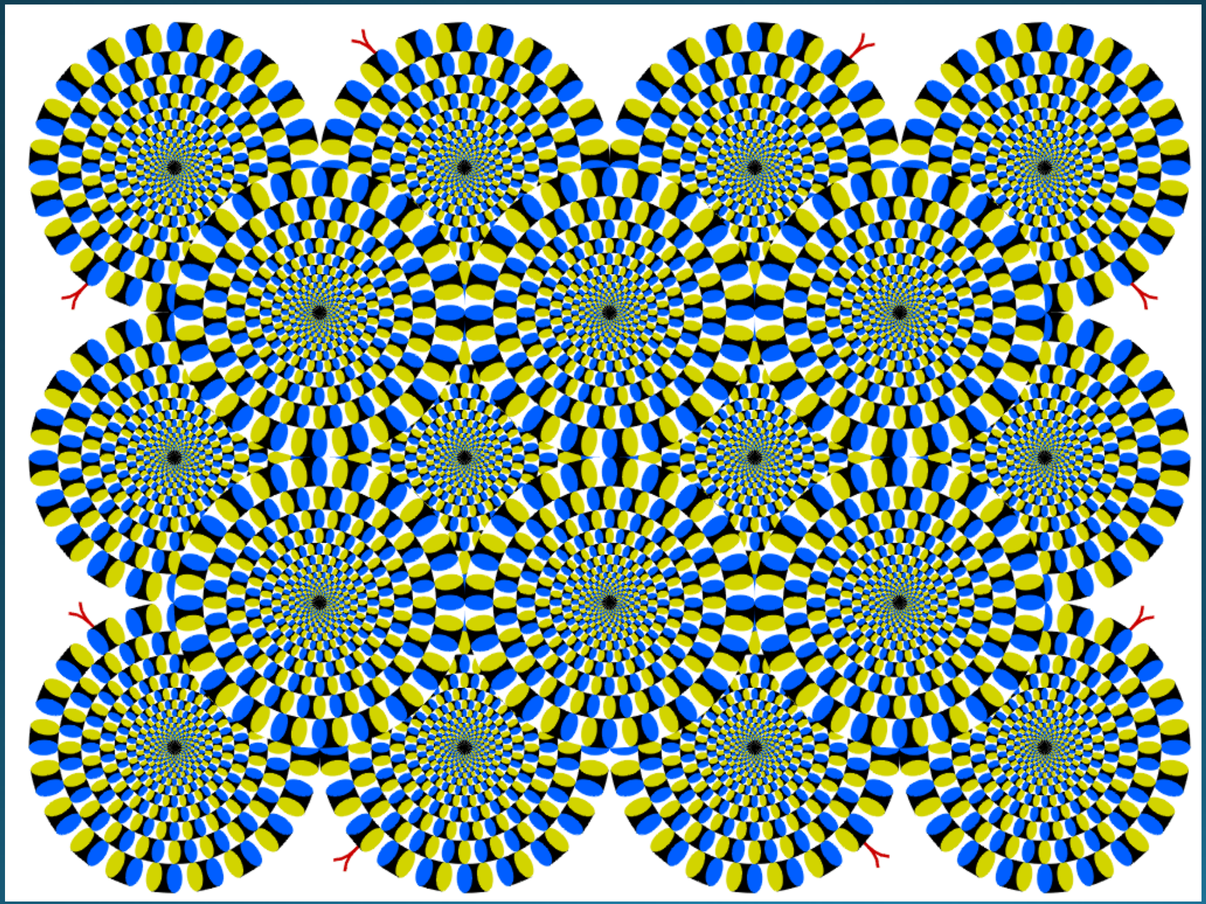


Perception vs Representation

- We can *perceive* motion where none exists, or not perceive motion where motion exists

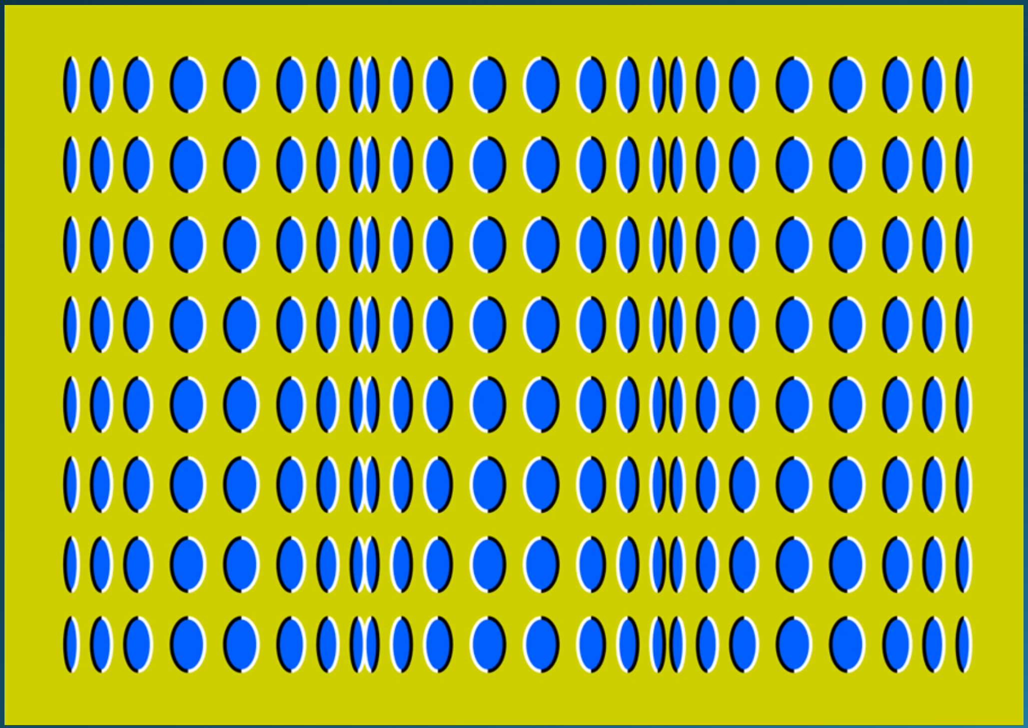
Perception vs Representation

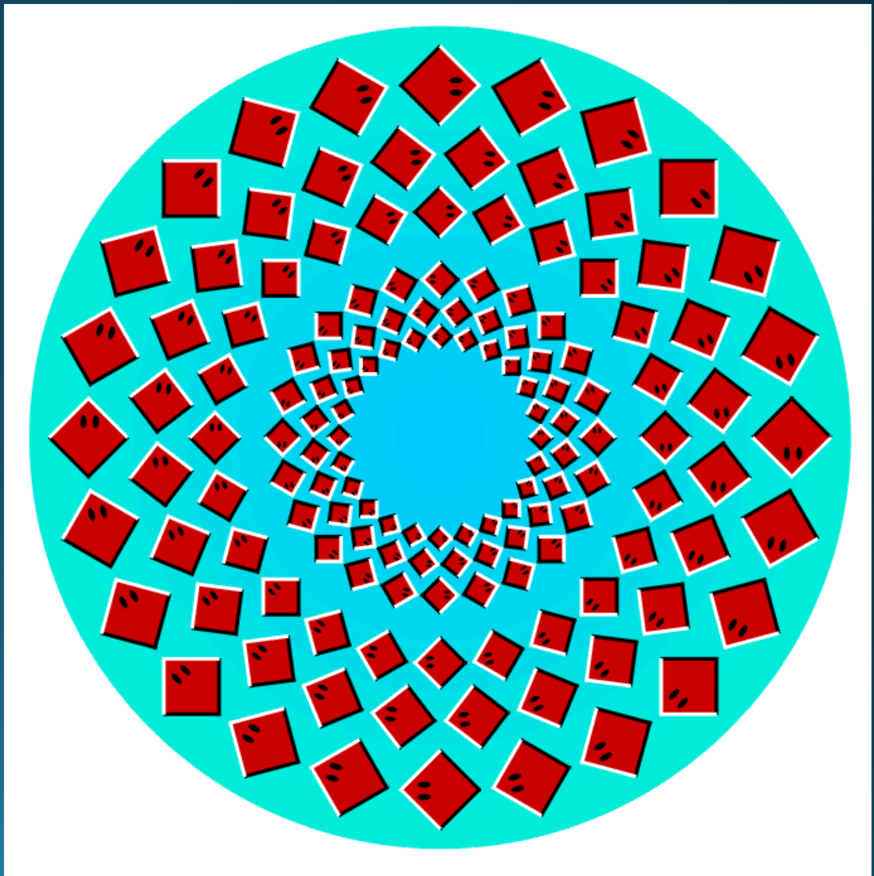
(I promise this is a static image)



Perception vs Representation

- Other examples



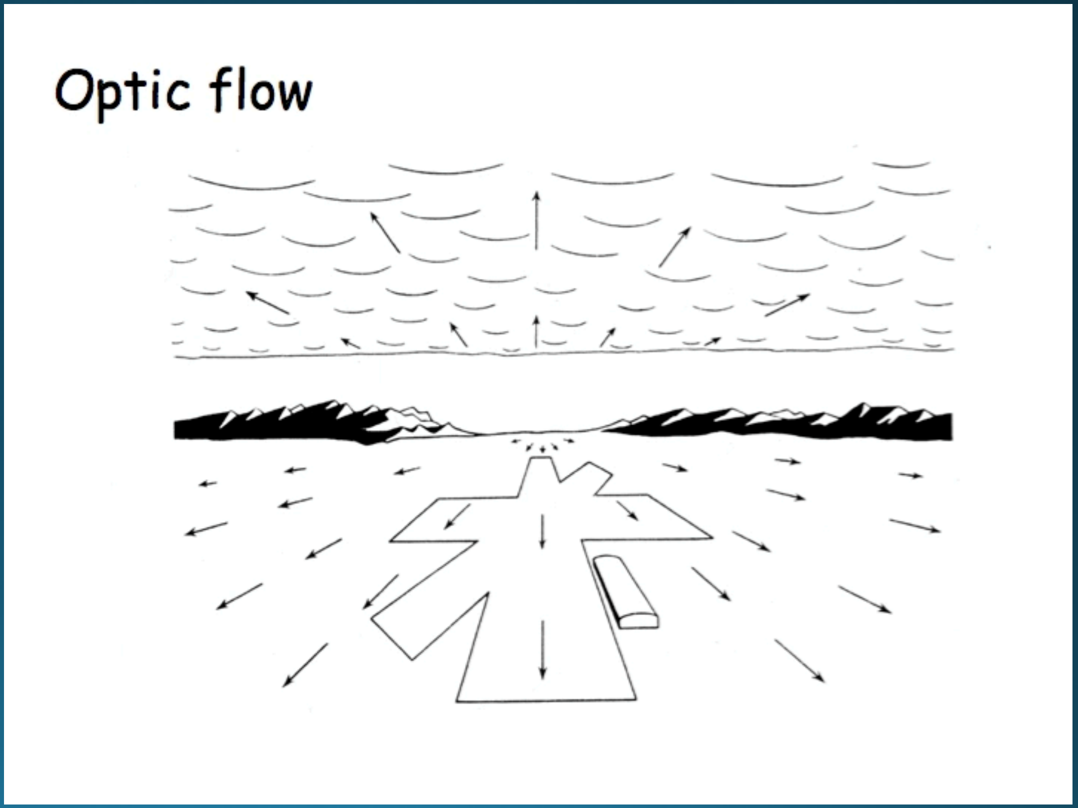


Perception vs Representation

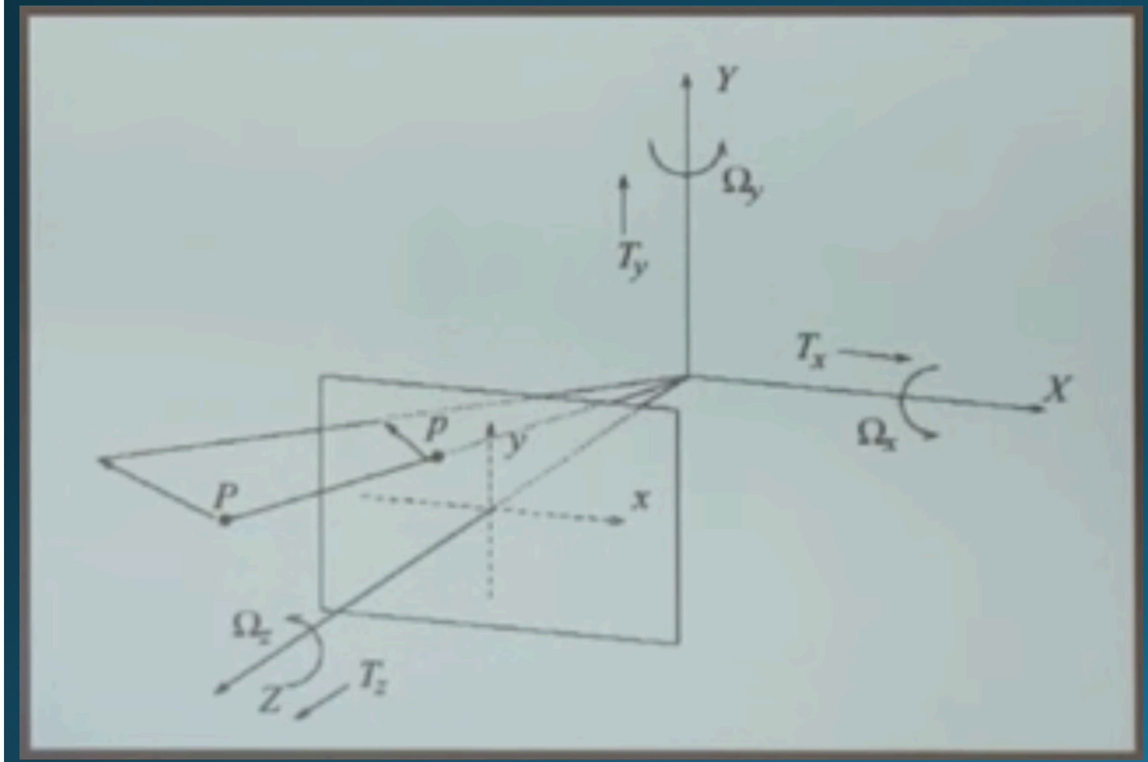
- Shapeless or transparent objects, or limited sight, are problematic
- Computer would not see motion in the previous images (which is good)
- … computer doesn’t “see” in the human sense
- Point being: **computers only analyze motion of opaque, solid objects**
- Key: **motion representation**

Representing Motion

- We perceive *optic flow*
- Pattern of flow (vectors)
- *Ecological optics* – J.J. Gibson



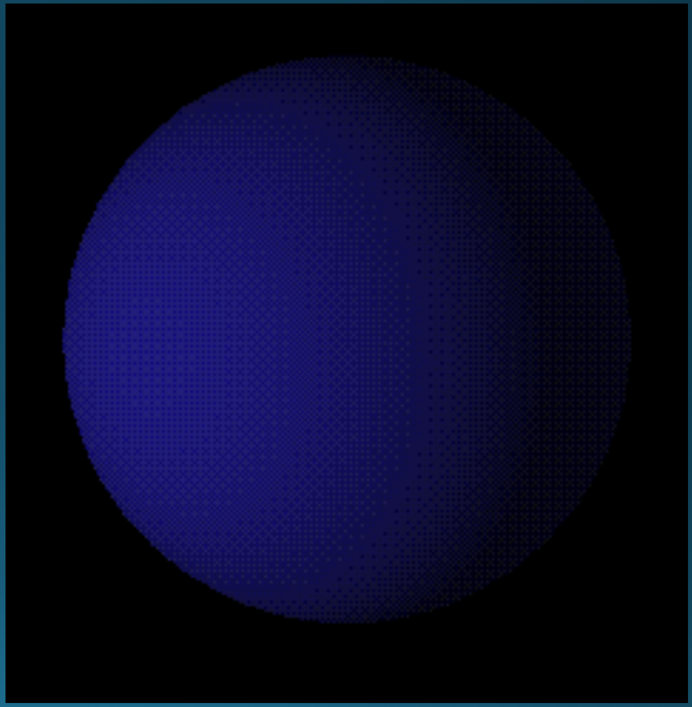
Representing Motion



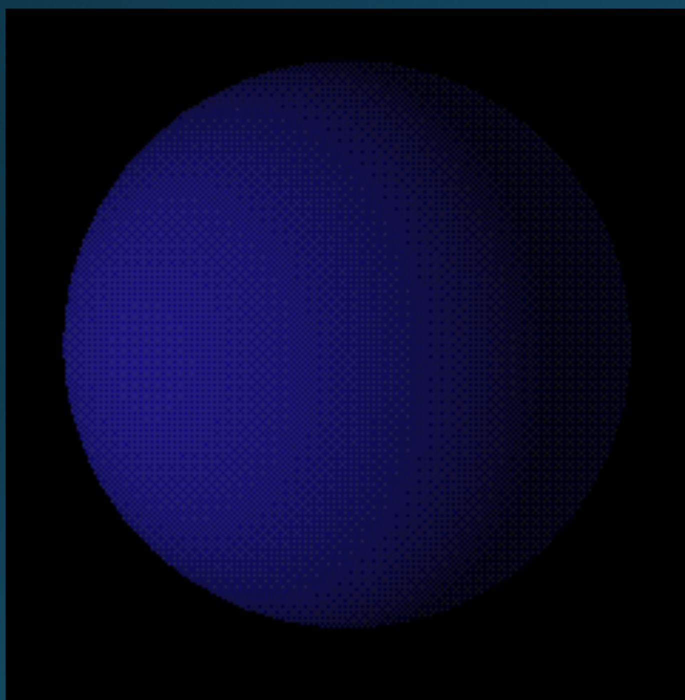
- Deviations
  - 3D motion of object is represented as 2D projection—losing 1 dimension of information
  - Optical flow = 2D velocity describing *apparent* motion

Thought Experiment 1

- We have a matte ball, rotating
- **What does the 2D motion field look like?**
- **What does the 2D optical flow field look like?**



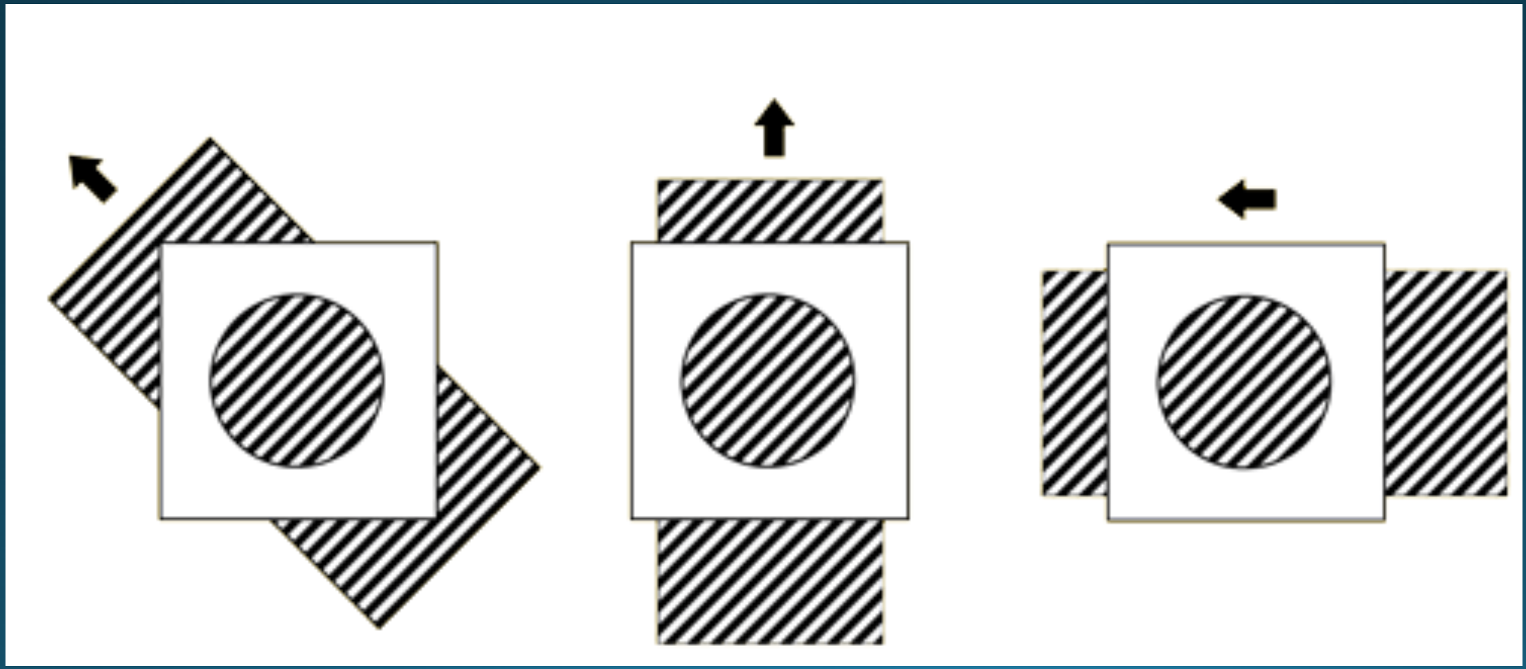
Thought Experiment 2



- We have a matte ball, *stationary*
- **What does the 2D motion field look like?**
- **What does the 2D optical flow field look like?**

Just to throw a wrench in things…

- The **Aperture Problem:** lighting is not the only source of error.



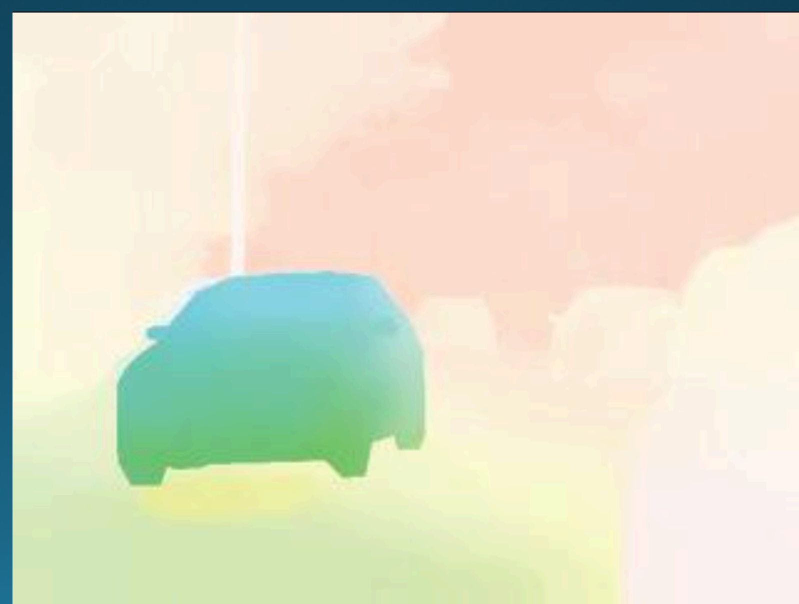
Aside

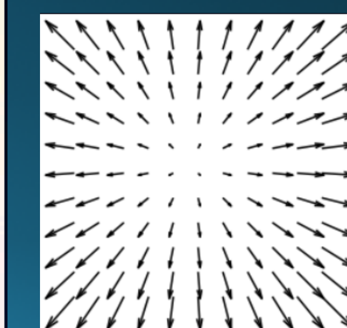
- With all these limitations and pitfalls, it's important to keep the following items in mind (with thanks to Dr. Michael Black):

- We are, more or less, intentionally forgetting any physics we might know
- We are dealing with **images**
- We're hoping the 2D flow is *related* to the structure of the world and can be a viable proxy for the motion field
- Fixing the above is important—**you could work on it!**

Optical Flow

- Motion, or *displacement*, at all pixels
  - Magnitude: saturation
  - Orientation: hue





Optical Flow Goals

- Find a mapping for each pixel (x_1, y_1) -> (x_2, y_2)
  - Seems simple enough…?
- Motion types
  - Translation

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 + a \\ y_1 + b \end{bmatrix}$$

  - Similarity

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = s \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} x_1 + a \\ y_1 + b \end{bmatrix}$$

  - Affine

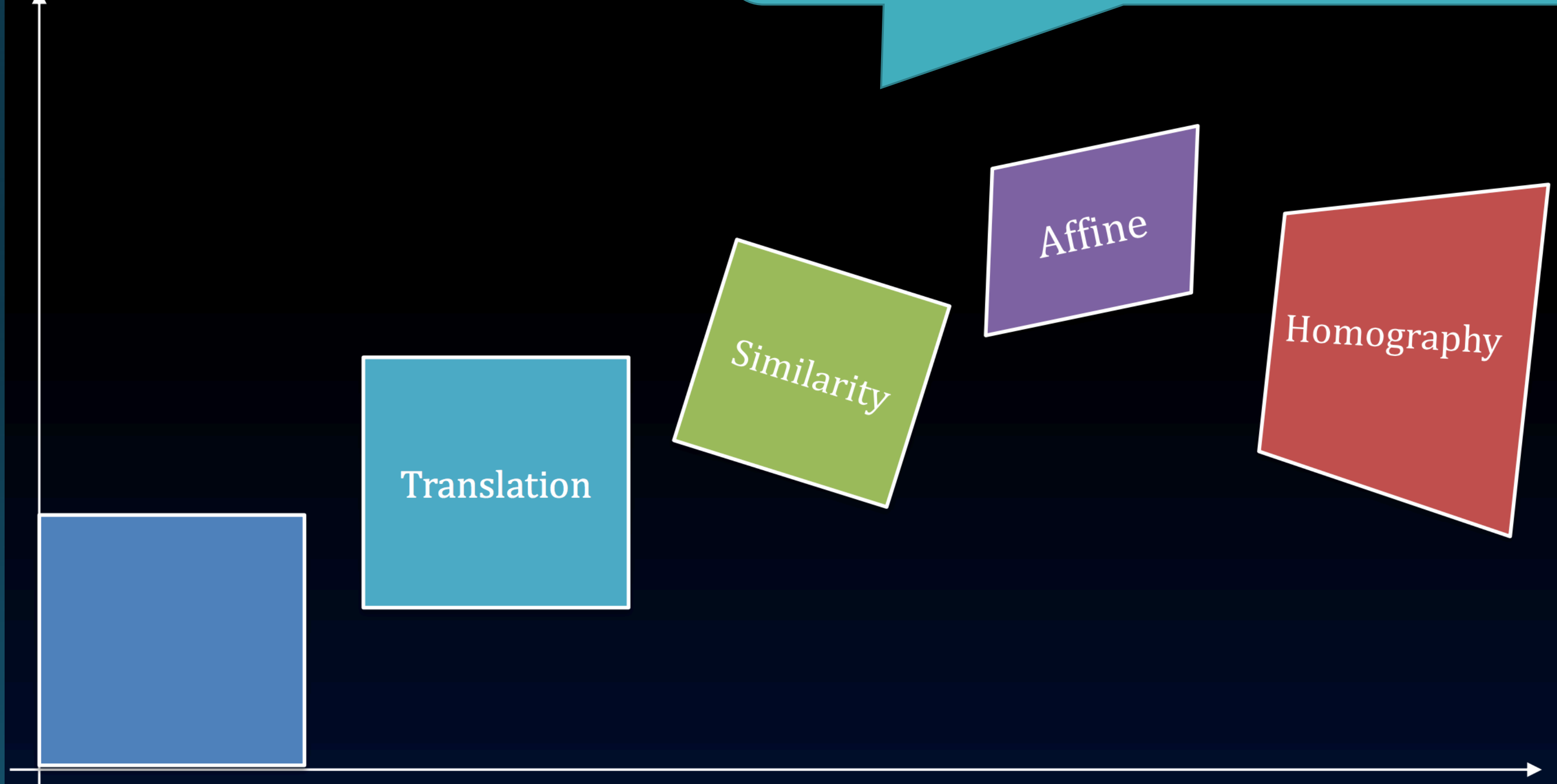
$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} ax_1 + by_1 + c \\ dx_1 + ey_1 + f \end{bmatrix}$$

  - Homography

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \frac{1}{z} \begin{bmatrix} ax_1 + by_1 + c \\ dx_1 + ey_1 + f \end{bmatrix}, z = gx_1 + hy_1$$

Motion Types

This is known as **parametric motion**: powerful in its expressivity, but limited in its ability to describe arbitrary motion in videos.

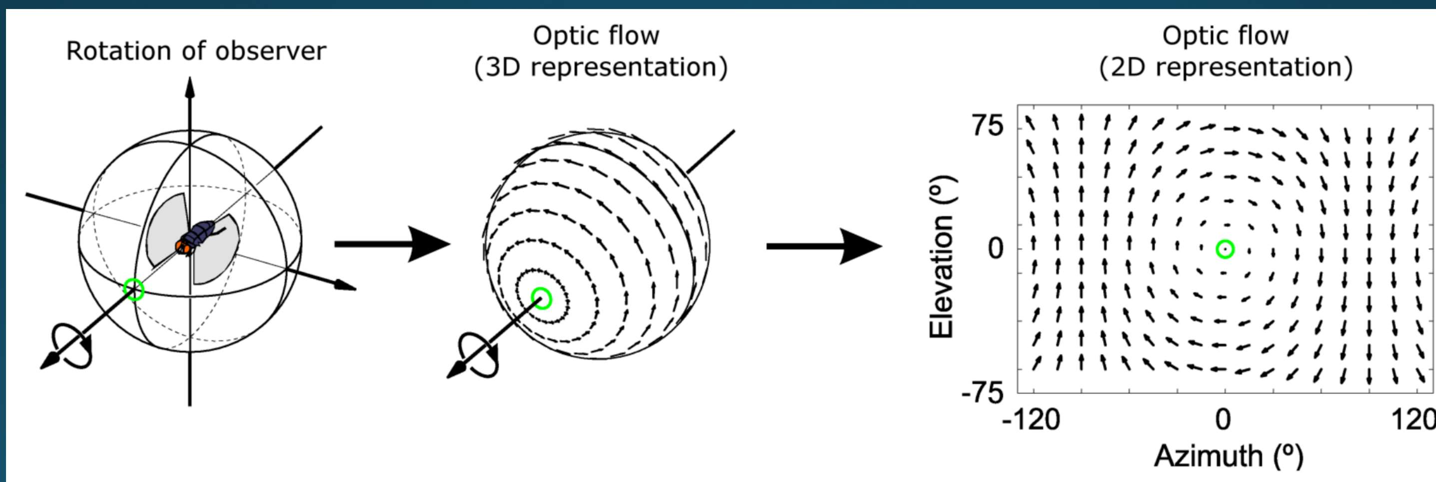


Optical Flow Definition

- Image pixel value at time *t* and location **x** = (*x*, *y*)
- Horizontal *u* and vertical *v* components of the flow

$$I(x, y, t)$$

$$u(x, y) \quad v(x, y)$$

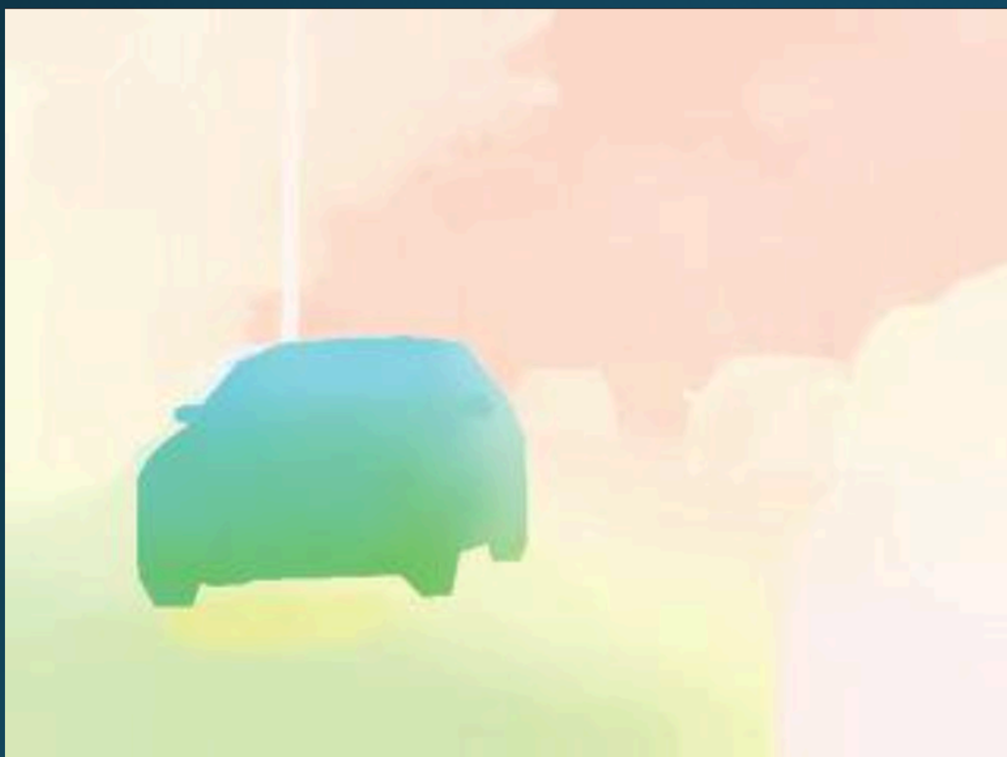


Optical Flow Assumptions

- **Brightness Constancy**
- Any one patch from frame 1 should look more or less the same as a corresponding spatial patch from frame 2

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

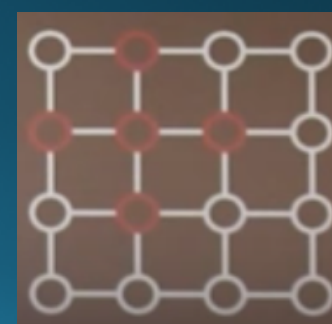
Optical Flow Assumptions



- **Spatial Smoothness**
- Neighboring pixels in an image are likely to belong to the same surface
  - Surfaces are mostly smooth
  - Neighboring pixels have similar flow

$$u_p = u_n$$

$$n \in G(p)$$



Objective Function

- Brightness constancy ("data term")

$$E_D(u,v) = \sum_s (I(x_s + u_s, y_s + v_s, t+1) - I(x,y,t))^2$$

- New developments?
  - Squared error implies Gaussian noise!

Objective Function

- Spatial term for the flow fields *u* and *v*

$$E_S(u,v) = \sum_{n \in G(s)} (u_s - u_n)^2 + \sum_{n \in G(s)} (v_s - v_n)^2$$

- New developments?
  - Flow field is smooth
  - Deviations from smooth are Gaussian
  - First-order smoothness is all that matters
  - Flow derivative is approximated by first differences

Objective Function

$$E(u,v) = E_D(u,v) + \lambda E_S(u,v)$$

$$E(u,v) = \sum_s (I(x_s + u_s, y_s + v_s, t+1) - I(x,y,t))^2$$
$$+ \lambda \left(\sum_{n \in G(s)} (u_s - u_n)^2 + \sum_{n \in G(s)} (v_s - v_n)^2 \right)$$

Objective Function

- So to solve for flow field, we just take derivative, set to 0, and solve for *u* and *v*, right?

$$E_D(u,v) = \sum_s (I(x_s + u_s, y_s + v_s, t+1) - I(x,y,t))^2$$

???

Linear approximation

- Taylor series expansion
  - *dx = u, dy = v, dt = 1*

$$E_D(u,v) = \sum_s (I(x_s + u_s, y_s + v_s, t+1) - I(x,y,t))^2$$

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

$$u\frac{\partial}{\partial x}I(x,y,t) + v\frac{\partial}{\partial y}I(x,y,t) + \frac{\partial}{\partial t}I(x,y,t) = 0$$

Constraint equation

$$u\frac{\partial}{\partial x}I(x,y,t) + v\frac{\partial}{\partial y}I(x,y,t) + \frac{\partial}{\partial t}I(x,y,t) = 0$$

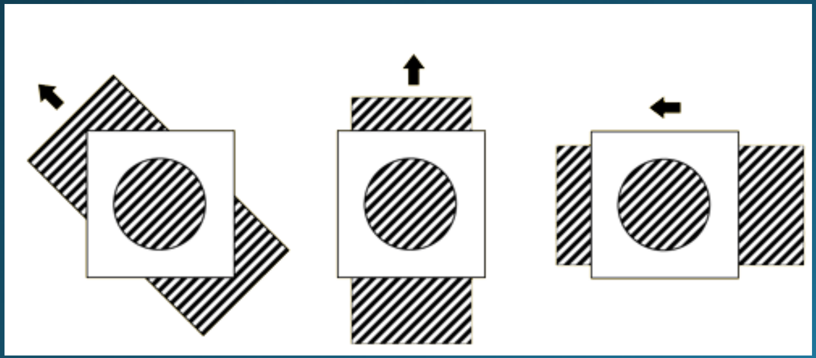
- ...but really, we write it this way:

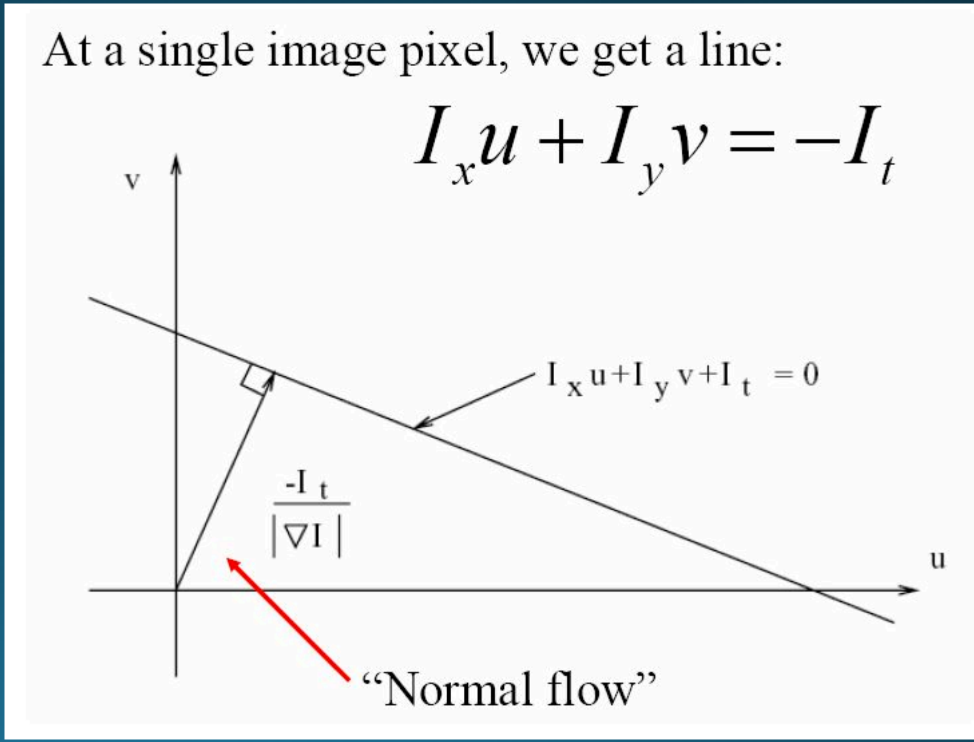
$$I_x u + I_y v + I_t = 0$$

- More new developments
  - Flow is small
  - Image is a differentiable function
  - First-order Taylor series is a good approximation

Form of the constraint equation

- One equation, two unknowns
  - A line
- We know the solution is somewhere along the line
- **Ill-posed problem: hence, the Aperture Problem**





Nevertheless, they persisted

- Horn and Schunck, 1981

$$E(u,v) = \sum_s (I_{x,s}u_s + I_{y,s}v_s + I_{t,s})^2 + \lambda \sum_{n \in G(s)} \left((u_s - u_n)^2 + (v_s - v_n)^2\right)$$

- Take partial derivatives with respect to *u* and *v*; set to 0

$$0 = \sum_s (I_{x,s}^2 u_s + I_{x,s}I_{y,s}v_s + I_{x,s}I_{t,s}) + \lambda \sum_{n \in G(s)} (u_s - u_n)$$

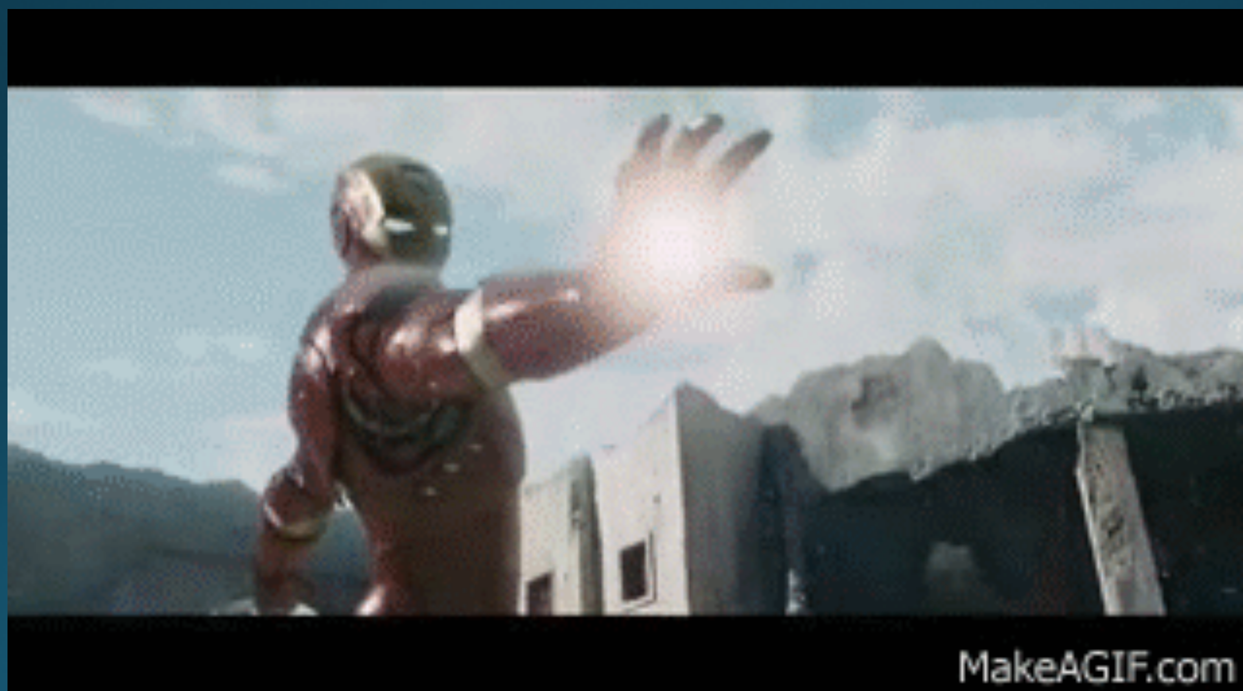
$$0 = \sum_s (I_{x,s}I_{y,s}u_s + I_{y,s}^2 v_s + I_{y,s}I_{t,s}) + \lambda \sum_{n \in G(s)} (v_s - v_n)$$

Revisiting assumptions

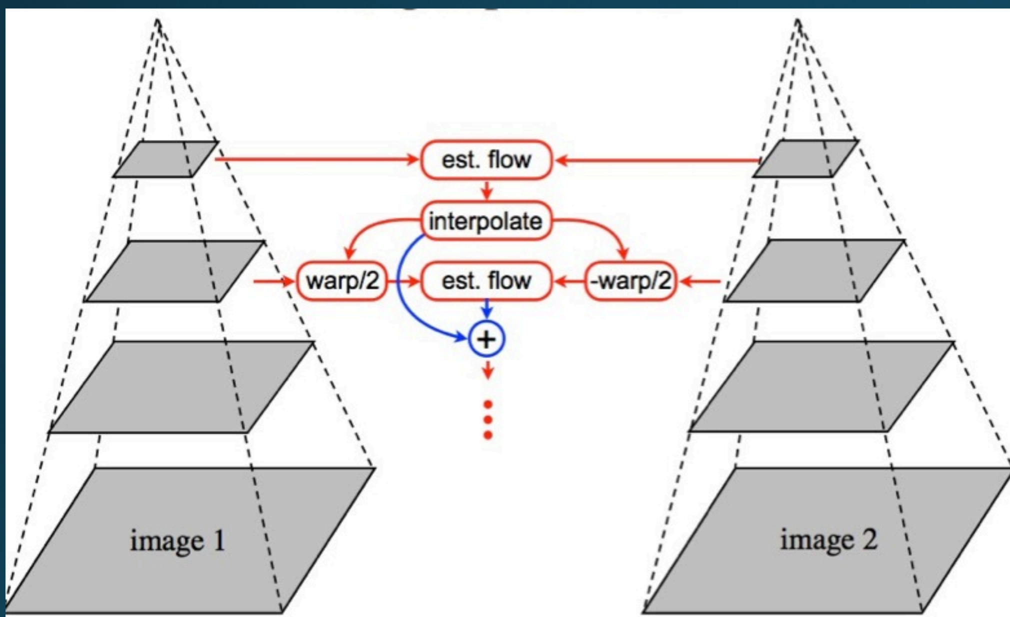
- Many of the Horn & Schunck '81 problems can be attributed to the fact that they were attempting dense image processing on 1981 computers
- Still, the problems outlined by the assumptions can cause problems in the real-world (aperture problem, ill-posed optimization, assumption of small motion, etc)
- Lots of these assumptions are still outstanding problems but have been addressed, at least in part
- (Check out the 2013 talk by Dr. Michael Black!)

“Flow is small”

- Have you ever *seen* a Marvel movie?



Coarse-to-fine

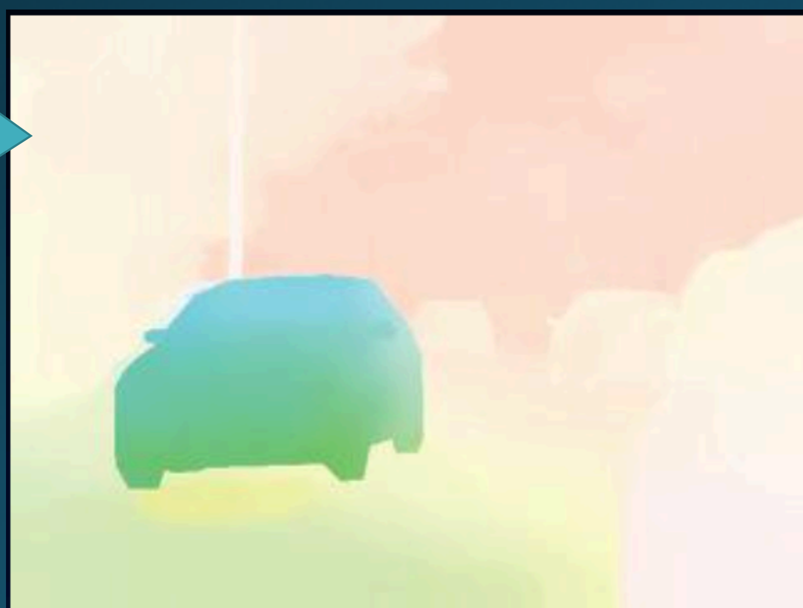


- Build an image “pyramid”
  - Exactly how this is done varies considerably
- Bottom line: flow calculated in original image is **much smaller** at top of pyramid (i.e., assumptions hold)
- Most optical flow algorithms do something like this

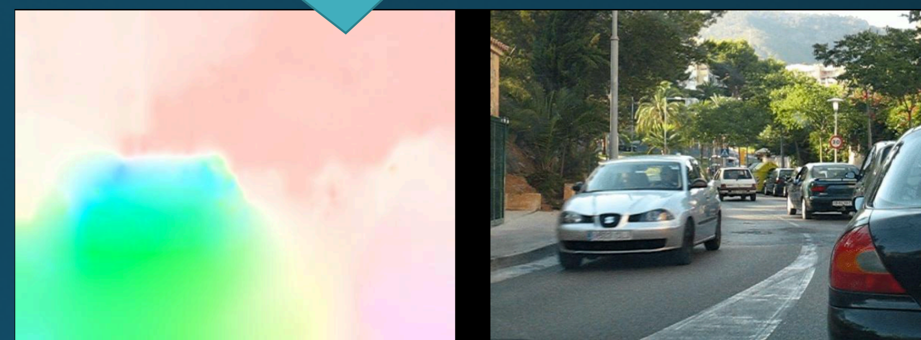
Coarse-to-fine

- This one “small” modification to Horn & Schunck actually gives pretty good results!

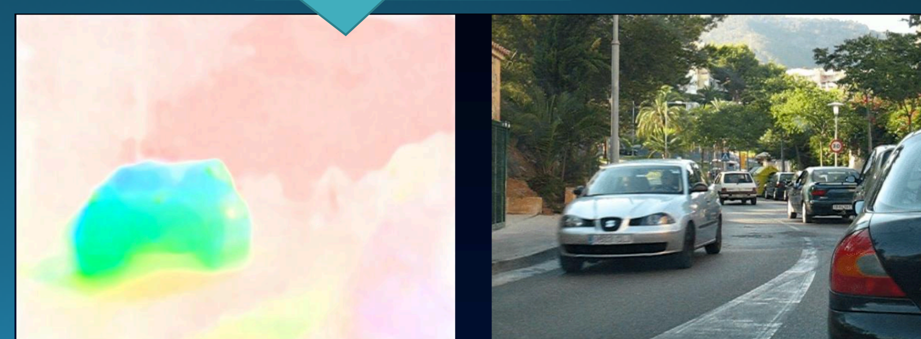
Ground Truth



Original



Coarse-to-fine



“Flow is smooth”

- Does brightness constancy hold?
- Are spatial derivatives of optical flow *actually* Gaussian?

- As machine learning practitioners, how would we answer these questions?

- **Need ground truth—unfortunately, these [largely] don’t exist**

Durien Open Movie Project

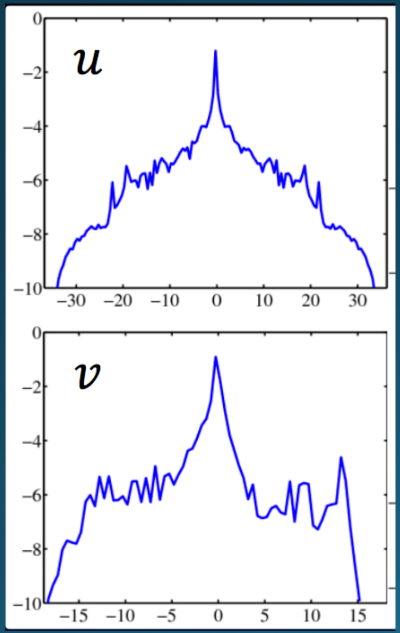
- *Sintel* (full movie—go watch!)
- Made with Blender
- All assets openly available—**including ground truth optical flow fields**
  - 1628 frames of ground truth flow
  - 1024x436 resolution
  - max velocity over 100 ppf
  - separated into training/testing

CS Mantra

- We solve one problem (need of ground-truth optical flow) by adding an additional abstraction layer (assume flow statistics of *Sintel* will generalize)
- ...which usually introduces a new problem
- **Will these flow statistics be at all useful for optical flow models outside of action movies?**

Flow Statistics

- In general, optical flow fields are sparse (i.e., most flow fields are 0)

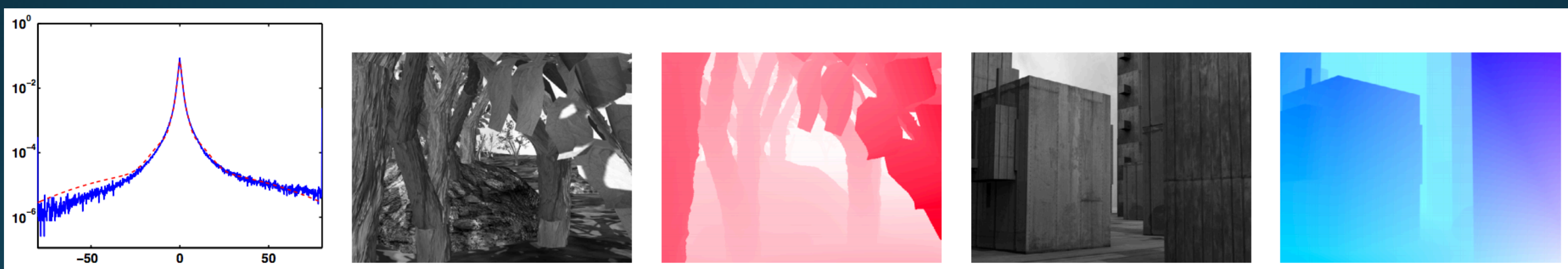


Horn and Schunck

Flow Statistics

- Using the flow statistics from training data, we can determine that brightness constancy **usually holds**

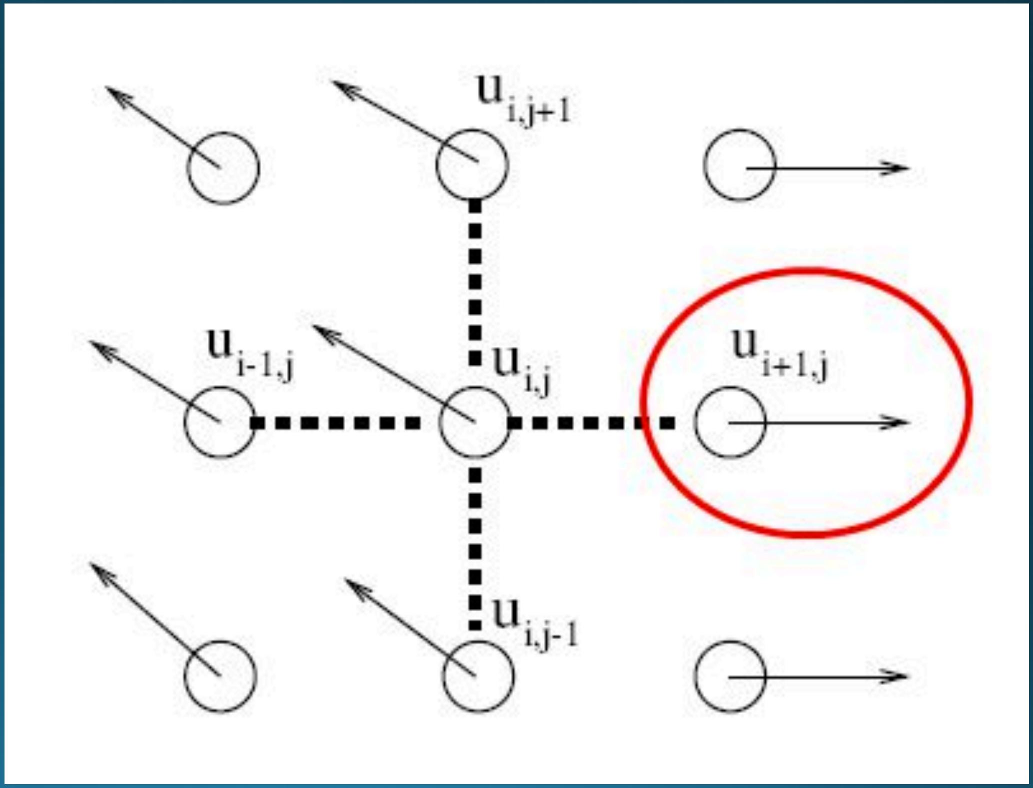
$$I_1(i, j) - I_2(i + u_{i,j}, j + v_{i,j})$$



- Spark peak at 0
- Heavy tails are violations of brightness constancy

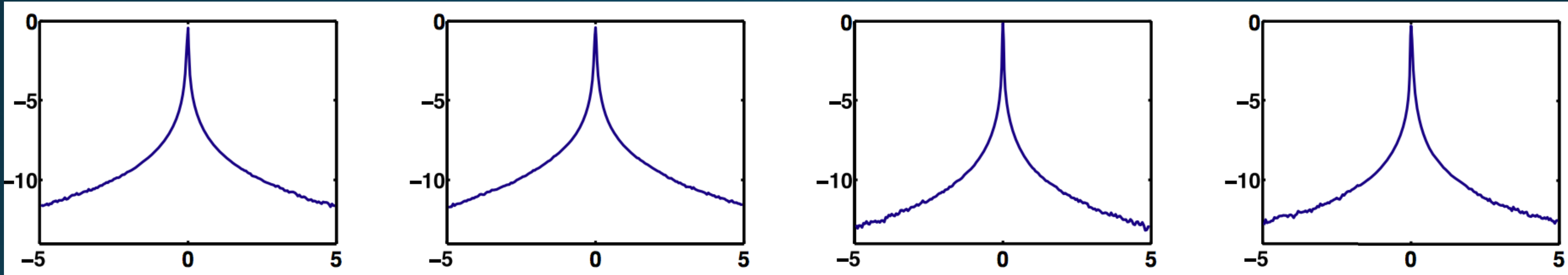
“Neighboring pixels move together”

- Except when they don’t
- Could consider these pixels as “spatial outliers”
- But want to consider them as part of **different surfaces** with **different motionsn**



Spatial statistics

- Spatial derivatives of the optical flow field *u* and *v*

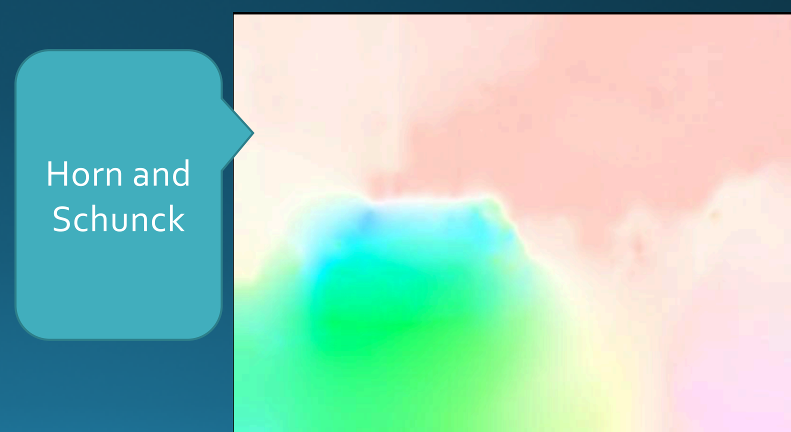


(a) $\partial u/\partial x$. (b) $\partial u/\partial y$. (c) $\partial v/\partial x$. (d) $\partial v/\partial y$.

- Similar story: flow is *usually* smooth, but motion boundaries create have heavy tails

Markov Random Fields

- The heavy tails on the spatial statistics are why optical flow has such problems with *object boundaries*
  - Quadratic smoothness term in objective
- Horn & Schunck [inadvertently?] kicked off 30+ years of research into Markov Random Fields
- Need a “robust” formulation that can handle multiple surfaces moving distinctly from each other



Robust Formulation

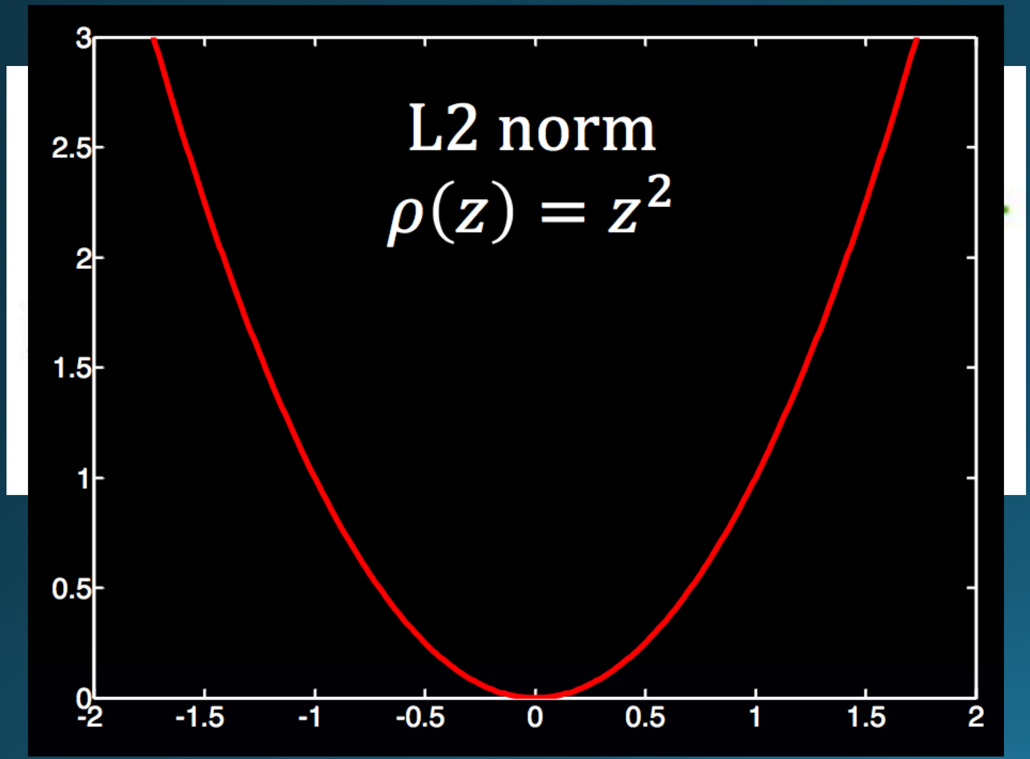
- Replace quadratic terms in original energy function with a new error function that gives *less* weight to *large* errors

$$E(u,v) = \sum_{s} \rho(I_{x,s}u_s + I_{y,s}v_s + I_{t,s}, \sigma_D)$$
$$+ \lambda \sum_{n \in G(s)} (\rho(u_s - u_n, \sigma_S) + \rho(v_s - v_n), \sigma_S))$$

- Note the rho functions and sigmas

$$\rho(x,\sigma) = \frac{x^2}{x^2 + \sigma^2}$$

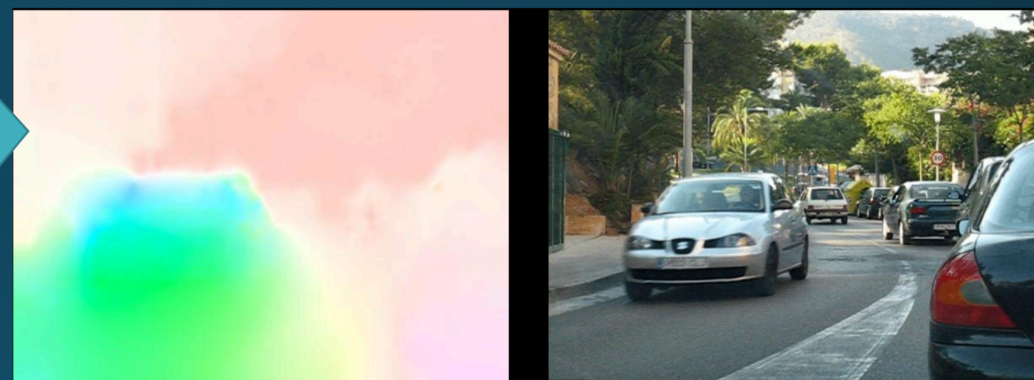
Robust Formulation

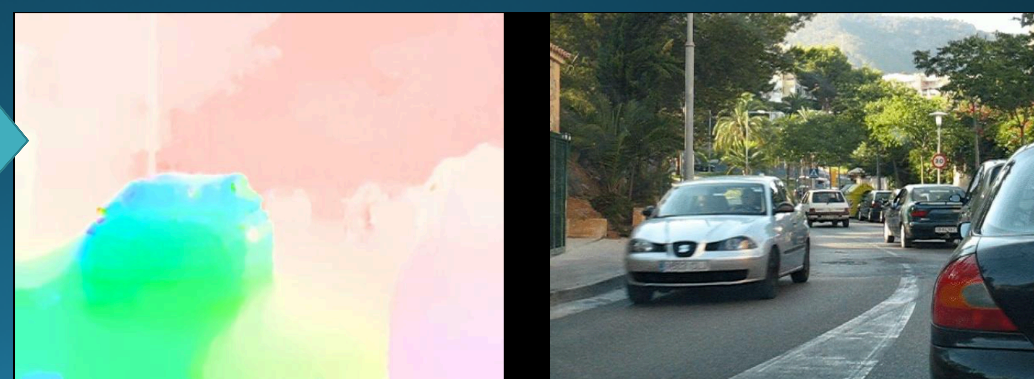


- Previous L2 (squared error) is sensitive to outliers
  - Outliers = occasional large flow derivatives
- New error function **saturates** at larger magnitudes
  - Is **robust to** outliers

Robust Formulation

- Object boundaries are considerably sharper
- Success! Go home?





Robust Formulation

- Optimization is *considerably* more difficult
- Non-linear in the flow term
- No closed-form solution

- Approaches
  - Gradient descent
  - Graduated non-convexity
  - Iteratively re-weighted least squares

Current Methods

- Current methods employ a combination of
  - Coarse-to-fine (image pyramids)
  - Median filtering (convolutions)
  - Graduated non-convexity
  - Image pre-processing
  - Bicubic interpolation (sparse to dense)
- Layers and segmentation (Sevilla-Lara *et al* 2016, CVPR)
- Pyramid networks (Ranjan *et al* 2016, CVPR)
- Deep convolutional networks (Dosovitskiy and Fischer *et al* 2015, ICCV)

References

- Dr. Ce Liu's Computer Vision course (lectures 19 and 20) http://people.csail.mit.edu/torralba/courses/6.869/6.869.computervision.htm
- Dr. Richard Szeliski's book (chapter 8) http://szeliski.org/Book/
- Dr. Michael Black's Optical Flow talk (2013) https://www.youtube.com/watch?v=tIwpDuqJqcE
- Sun *et al*, "Learning Optical Flow", ECCV 2008 http://people.seas.harvard.edu/~dqsun/publication/2008/ECCV2008.pdf
- Sonia Rao, UGA '20 Computer Science