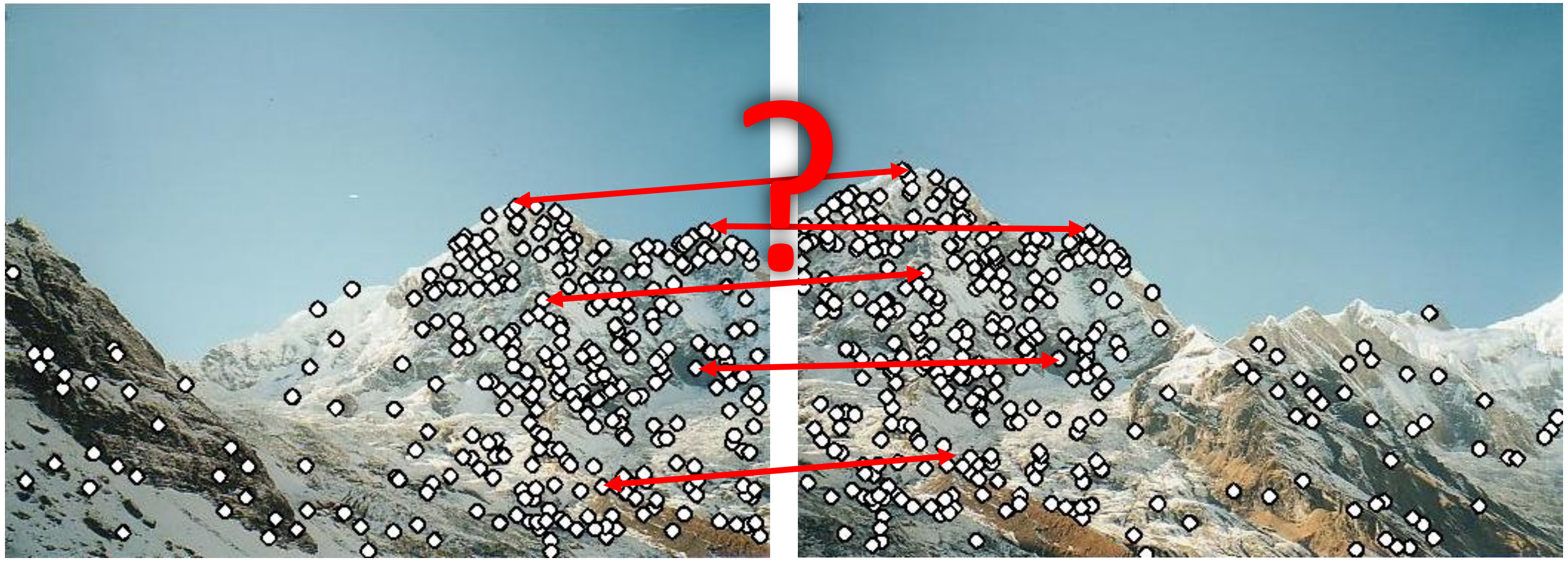
Correspondence

Matching feature points

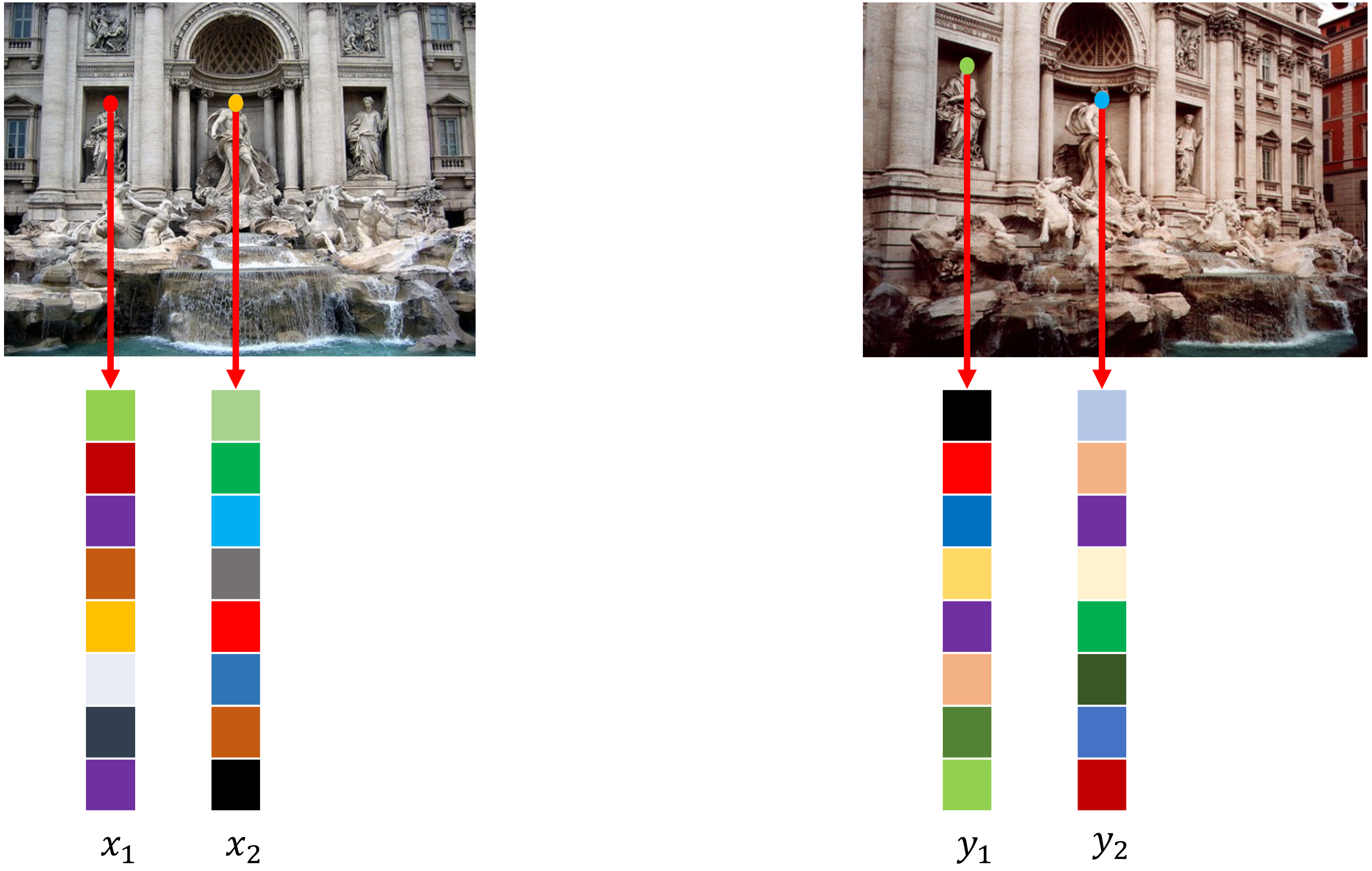
We know how to detect good points
Next question: **How to match them?**



Two interrelated questions:

1. How do we *describe* each feature point?
2. How do we *match* descriptions?

Feature descriptor

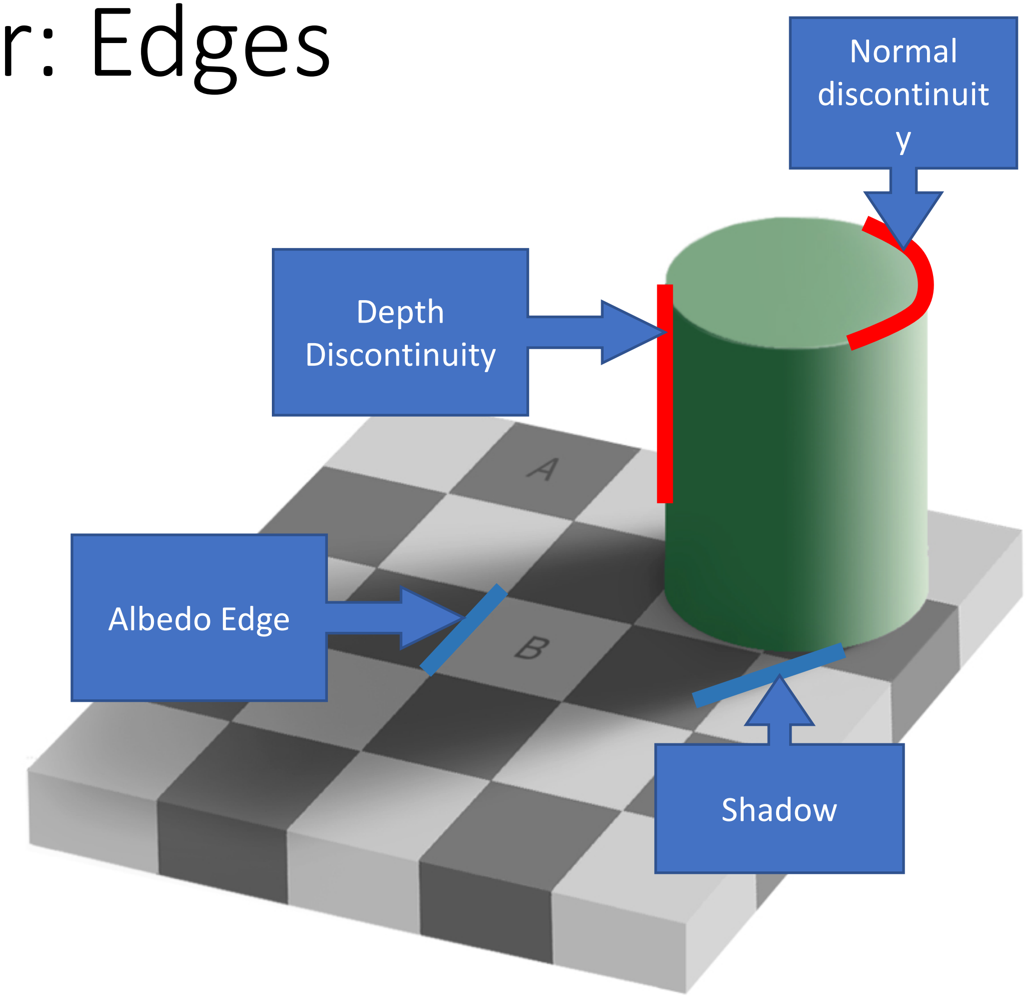


Feature matching

- Measure the distance between (or similarity between) every pair of descriptors

	y_1	y_2
x_1	$d(x_1, y_1)$	$d(x_1, y_2)$
x_2	$d(x_2, y_1)$	$d(x_2, y_2)$

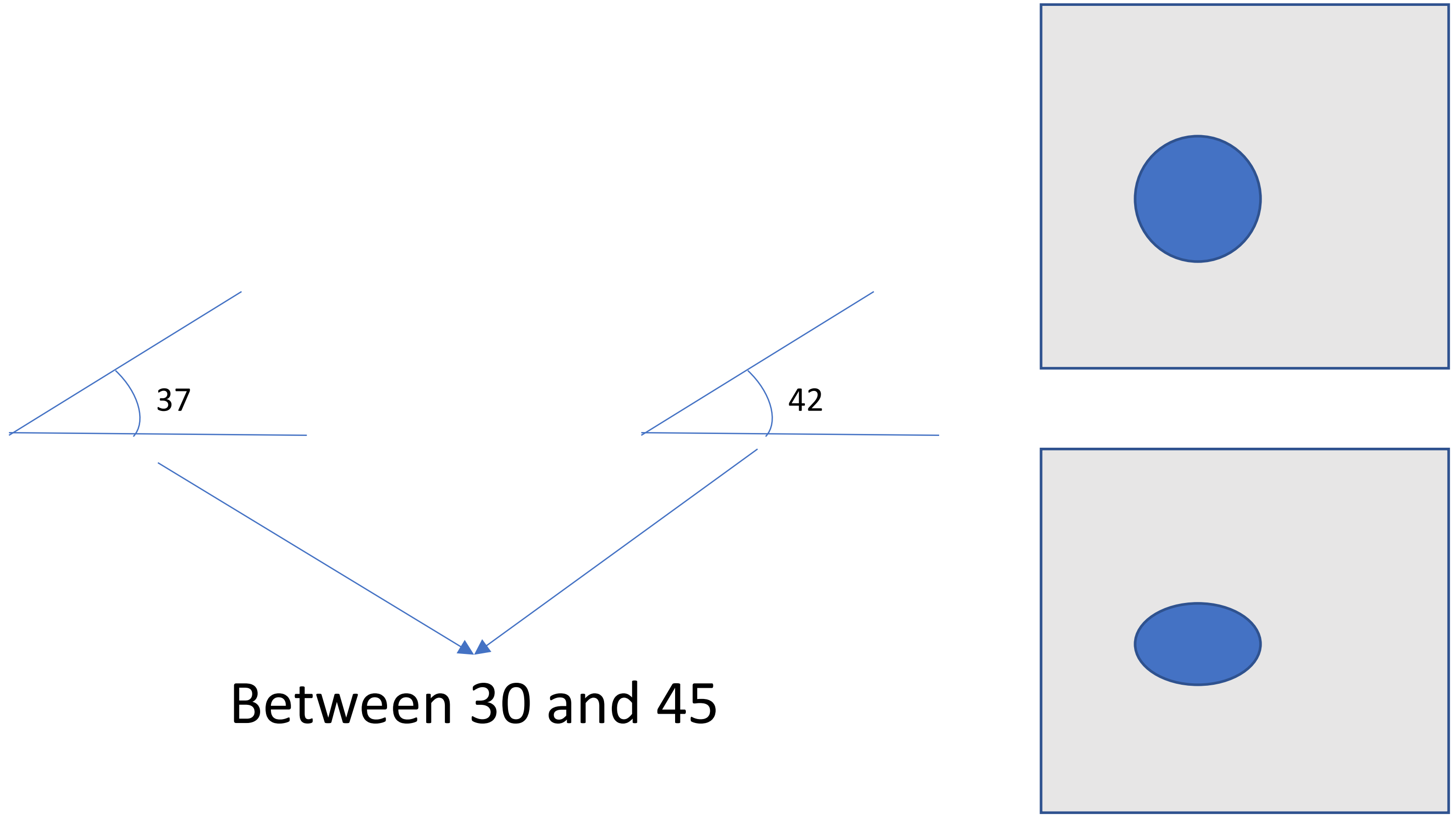
Better representation than color: Edges



Towards a better feature descriptor

- Match *pattern of edges*
  - Edge orientation – clue to shape
- Be resilient to *small deformations*
  - Deformations might move pixels around, but slightly
  - Deformations might change edge orientations, but slightly

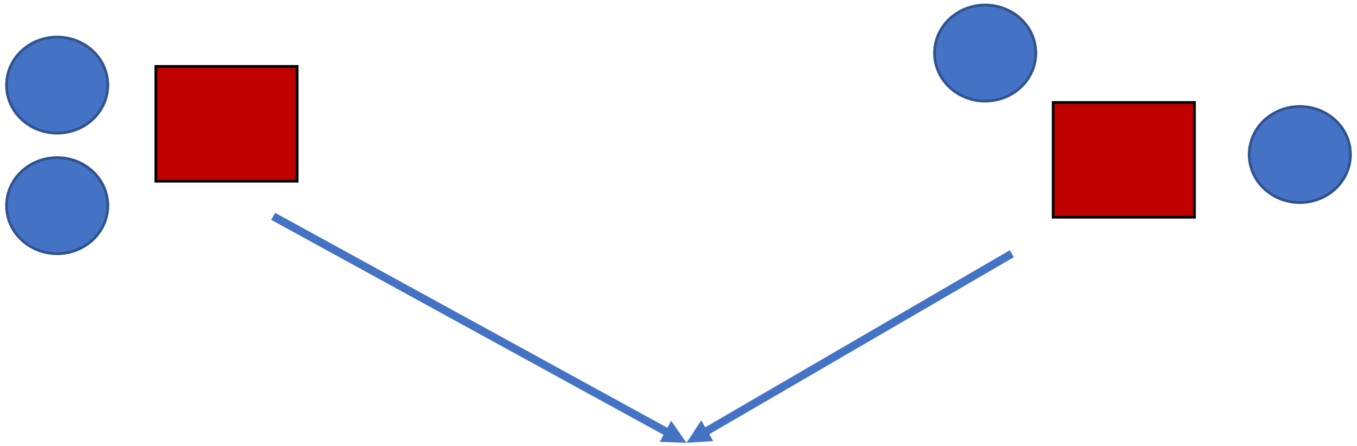
Invariance to deformation by quantization



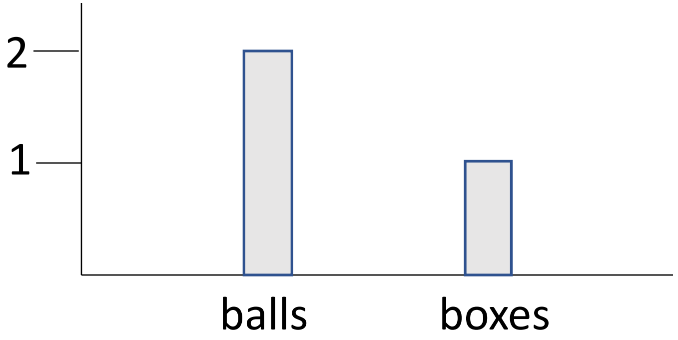
Invariance to deformation by quantization

$$g(\theta) = \begin{cases} 0 & \text{if } 0 < \theta < 2\pi/N \\ 1 & \text{if } 2\pi/N < \theta < 4\pi/N \\ 2 & \text{if } 4\pi/N < \theta < 6\pi/N \\ & \dots \\ N-1 & \text{if } 2(N-1)\pi/N \end{cases}$$

Spatial invariance by histograms



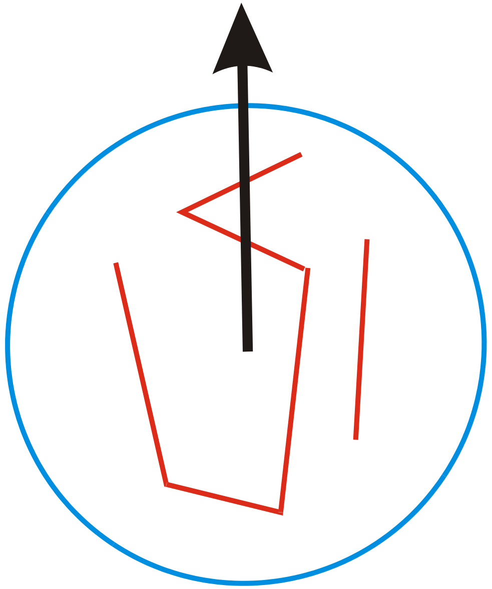
2 blue balls, one red box

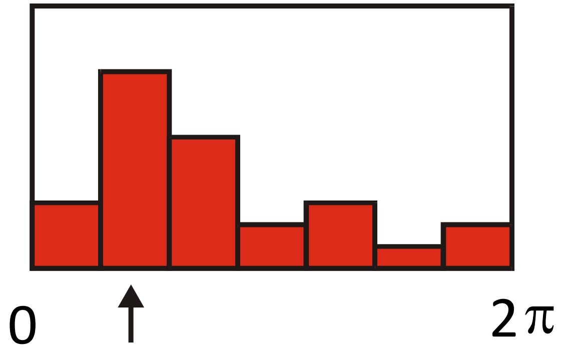


Rotation Invariance by Orientation Normalization

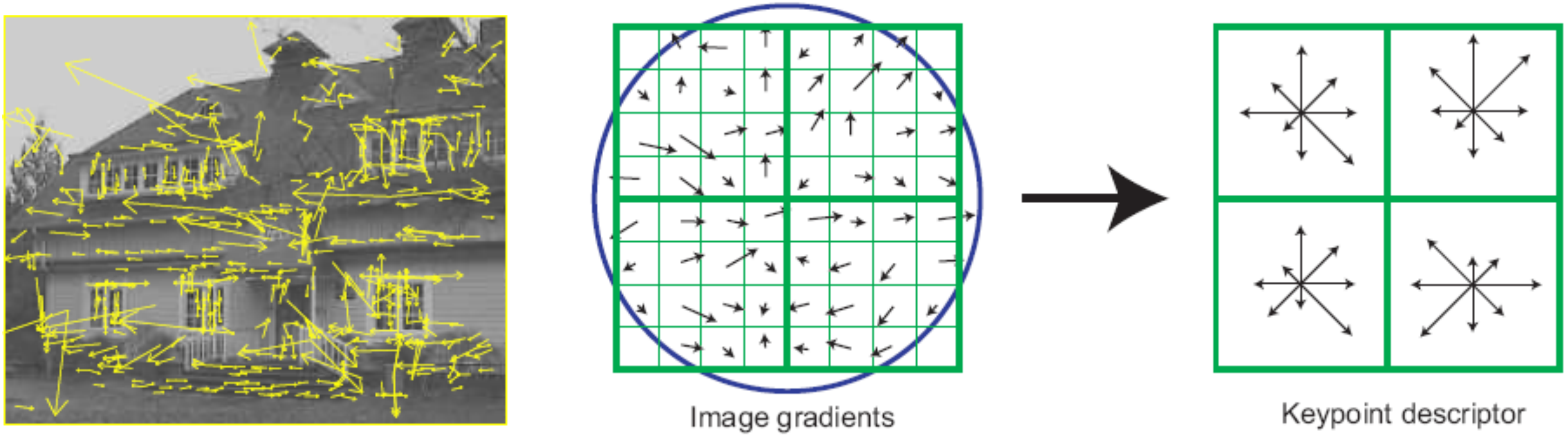
[Lowe, SIFT, 1999]

- Compute orientation histogram
- Select dominant orientation
- Normalize: rotate to fixed orientation





The SIFT descriptor



SIFT – Lowe IJCV 2004

Feature matching

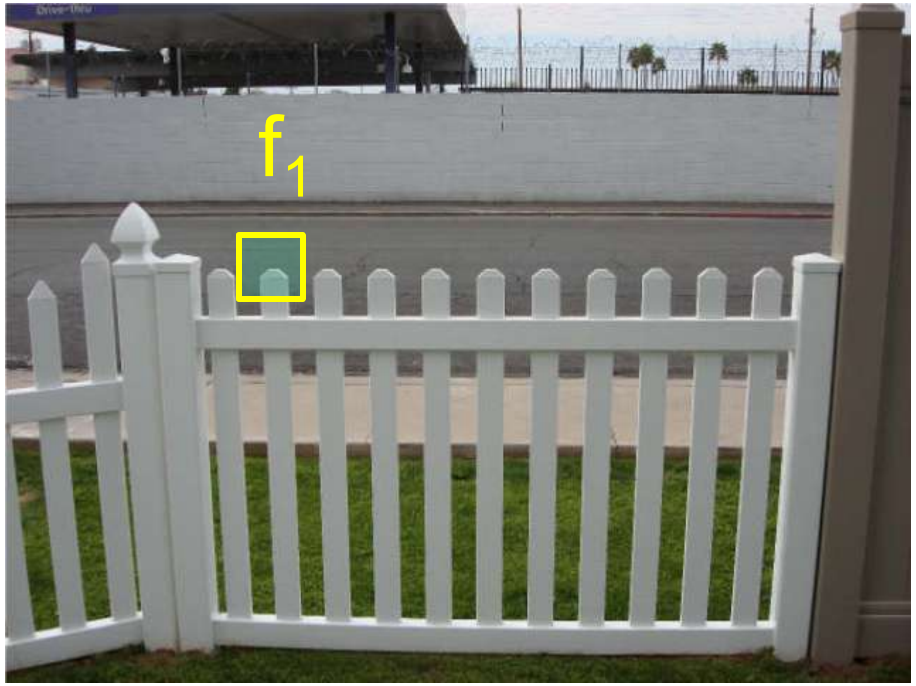
Given a feature in I_1, how to find the best match in I_2?

1. Define distance function that compares two descriptors
2. Test all the features in I_2, find the one with min distance

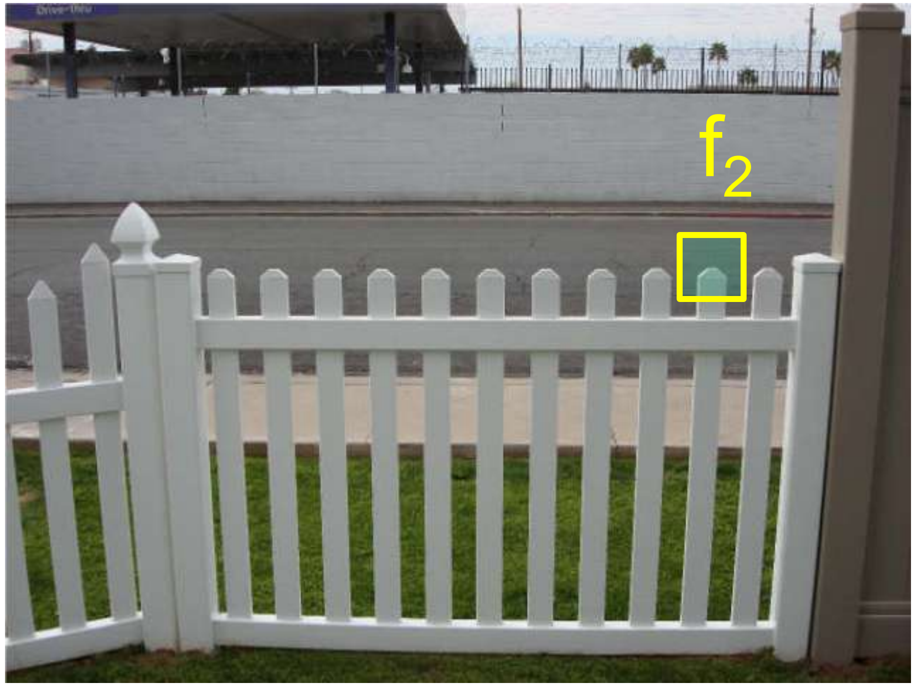
Feature distance

How to define the difference between two features f_1, f_2?

- Simple approach: L_2 distance, $||f_1 - f_2||$
- can give good scores to ambiguous (incorrect) matches



I_1

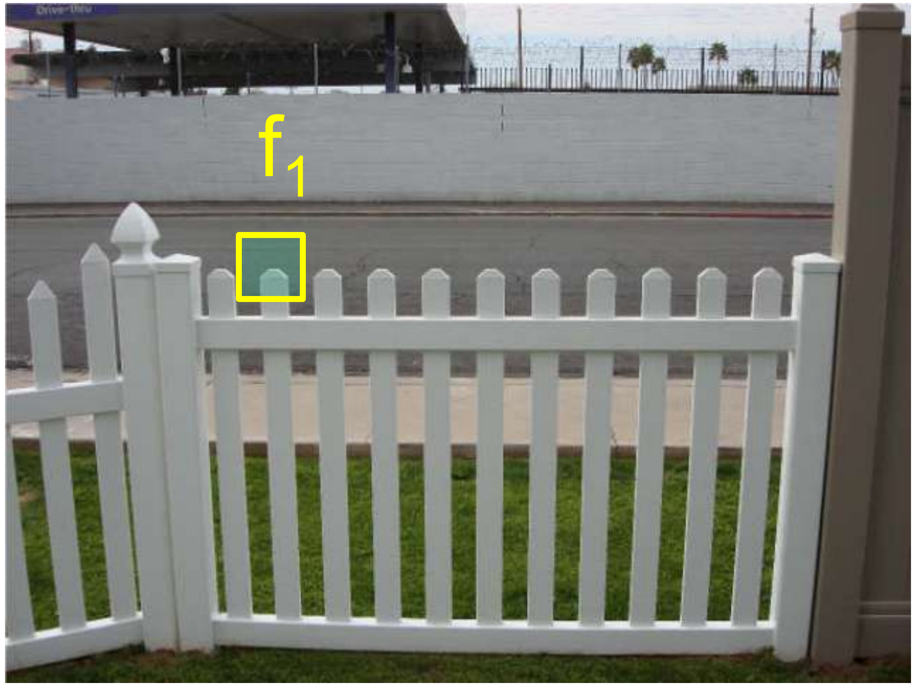


I_2

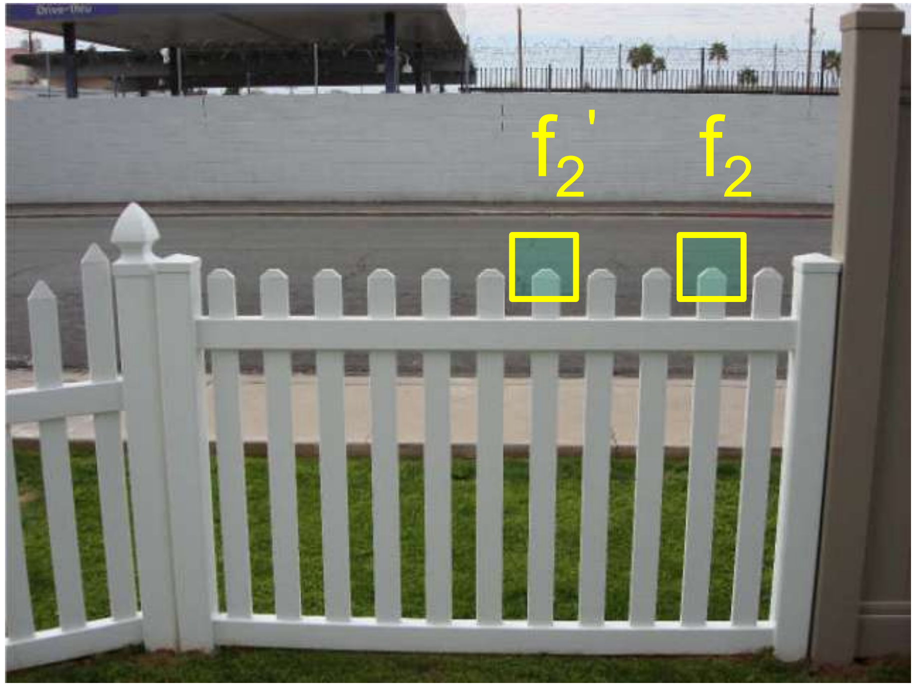
Feature distance

How to define the difference between two features f_1, f_2?

- Better approach: ratio distance = $||f_1 - f_2|| / || f_1 - f_2' ||$
  - f_2 is best SSD match to f_1 in I_2
  - f_2' is 2^{nd} best SSD match to f_1 in I_2
  - gives large values for ambiguous matches



I_1



I_2

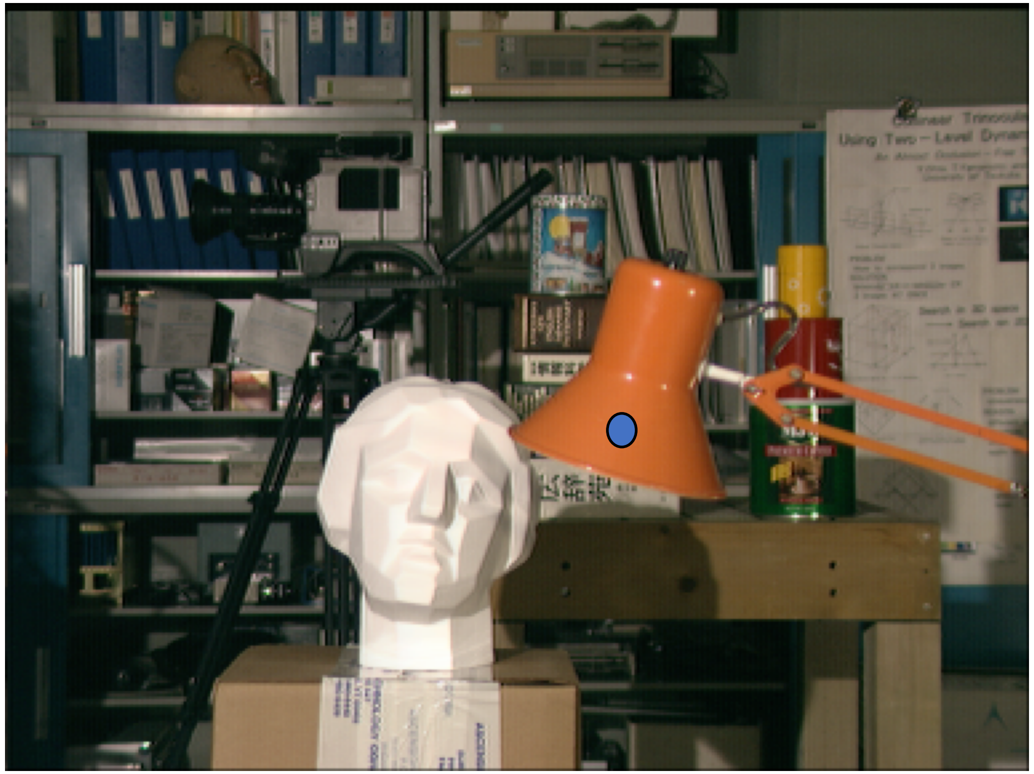
Structure from motion

- Given a bunch of images
- Get correspondences
  - Run interest point detector
  - Get SIFT descriptors
  - Match to get correspondences
- Use correspondences for
  - Estimating F/E, R, t
  - Estimating 3D structure of the world

Dense correspondence

- What if we need depth for every pixel?
- Setup: assume rectified images
  - Correspondence only along scan-lines
  - Can be represented using disparity

Dense correspondence



Dense correspondence

- Goal: Assign disparity value to each pixel
- Problem: most pixels will be ambiguous
- Solution: propagate from unambiguous to ambiguous pixels
- Basic idea: nearby pixels likely to have same disparity (*smoothness*)

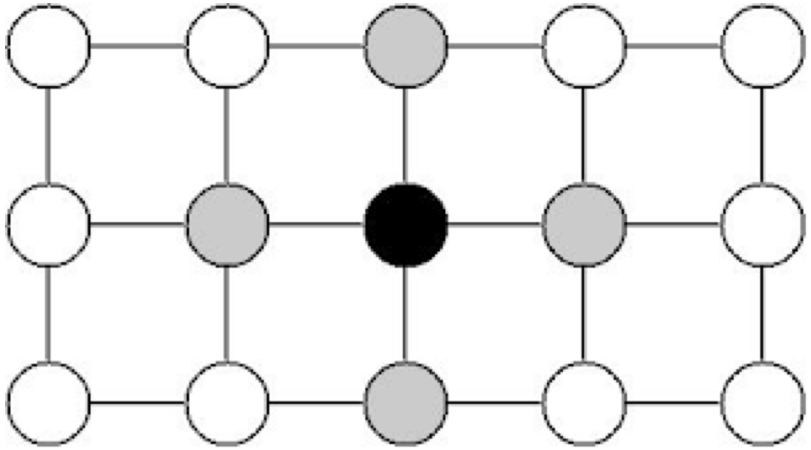
Dense correspondence

- Goal:
  - Assign disparity value to each pixel
- Basic idea:
  - Disparity image should be *smooth*
- Energy minimization
  - min E(d), where d is disparity image
  - $E(d) = E_{data}(d) + E_{smoothness}(d)$
- $E_{data}(d)$: scores based on NCC (for example)
- $E_{smoothness}(d) = \sum_{i,j} \rho(d(i,j) - d(i,j+1)) + \rho(d(i,j) - d(i+1,j))$

Markov Random Fields

- Probabilistic model
- Undirected graphical model

$$P(d) \propto e^{-E(d)}$$

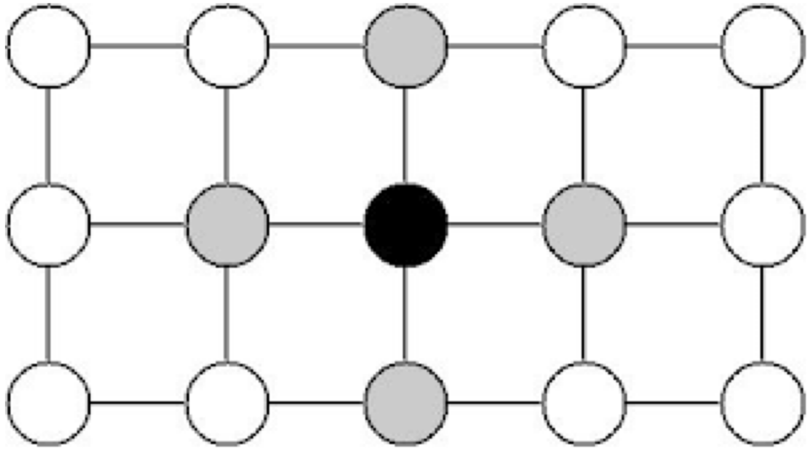


- Undirected graph with nodes and edges
- Unary potential on nodes = *data term*
- Binary potential on edges = *smoothness term*

$$E(d) = \sum_{(i,j)\in\mathcal{V}} \phi_u(d(i,j)) + \sum_{((i,j),(k,l))\in\mathcal{E}} \phi_b(d(i,j), d(k,l))$$

Optimizing MRFs

- NP-Hard
- Approximate solutions
  - Message passing
  - Graph cut-based solutions



Dense correspondence with MRFs

Dense correspondence

- Goal: Assign disparity value to each pixel
- Problem: most pixels will be ambiguous
- Solution: propagate from unambiguous to ambiguous pixels
- Basic idea: nearby pixels likely to have same disparity (*smoothness*)

Dense correspondence

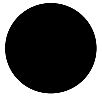
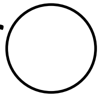
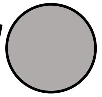
- Obtain disparity through optimization

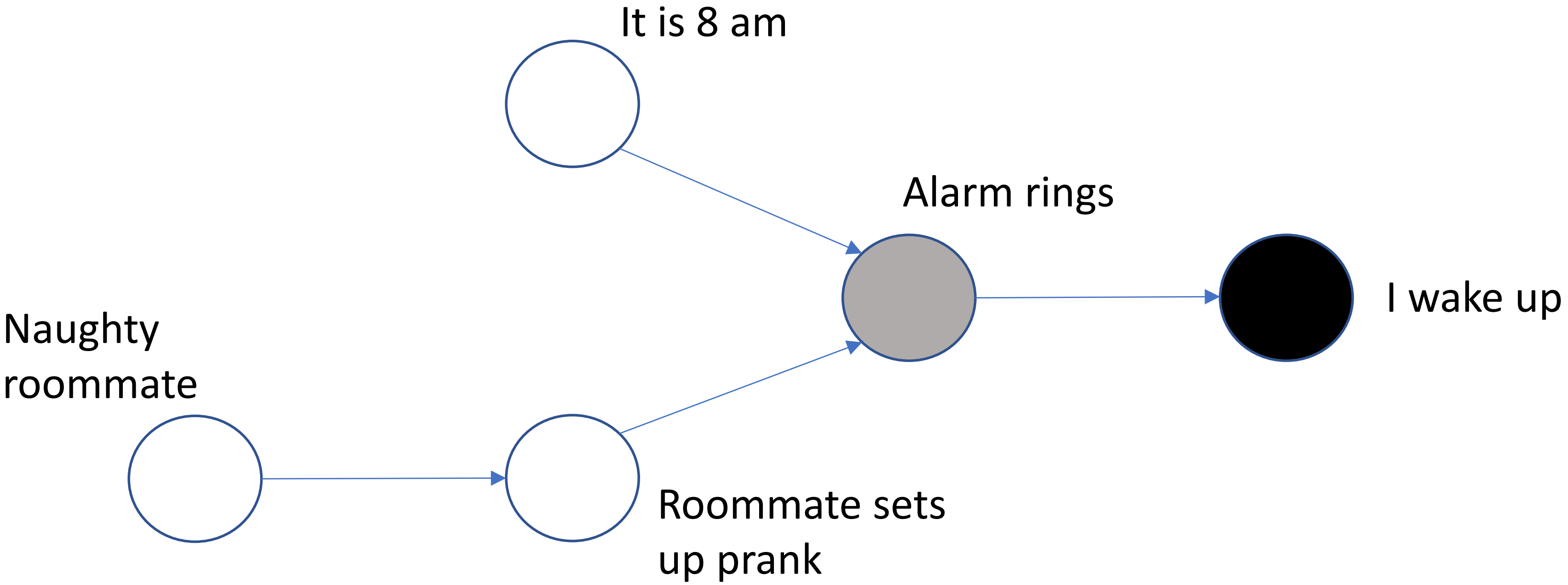
$$\mathbf{d}^* = \arg\min_{\mathbf{d}} \sum_{(i,j)} E_{data}(d(i,j)) + \sum_{(i,j),(k,l)\in\mathcal{N}} E_{smooth}(d(i,j), d(k,l))$$

Based on e.g
NCC distance

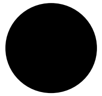
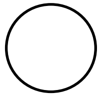
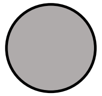
$(d(i,j) – d(k,l))^2$

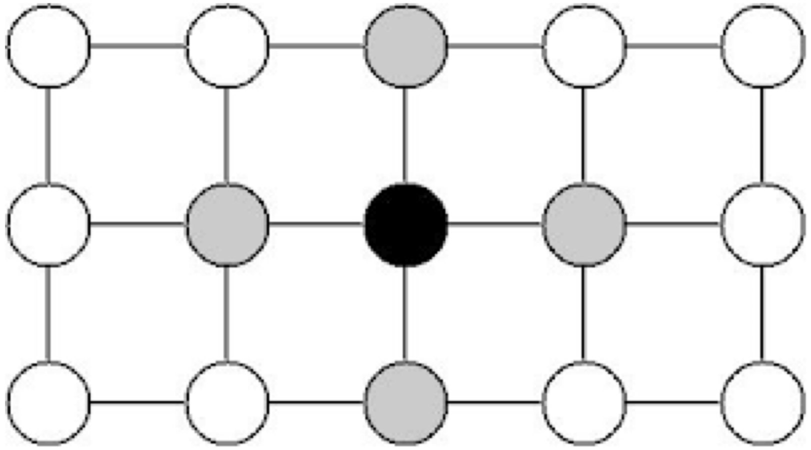
Detour: Graphical models

- Probabilistic models with graphs
- Nodes are variables
- Edges determine dependency structure
-  *independent of*  *given all* 

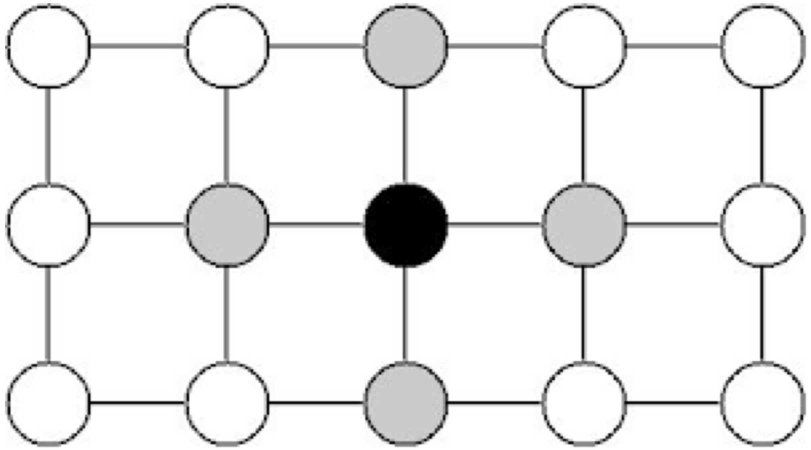


Markov Random Fields (MRFs)

- Probabilistic model
- Represented by graph
- Each node is random variable
- Edges represent *dependence structure*
  -  *independent of*  *given all* 



Markov Random Fields (MRFs)



- Hammersley-clifford theorem

$$P(X = \mathbf{x}) \propto e^{-E(\mathbf{x})}$$

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \phi_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \psi_{ij}(x_i, x_j)$$

Unary potential

Binary potential

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} P(X = \mathbf{x}) = \arg\min_{\mathbf{x}} E(\mathbf{x})$$

Dense correspondence as MRFs

- Obtain disparity through optimization
- Random variable: disparity
- Find *most likely disparity*

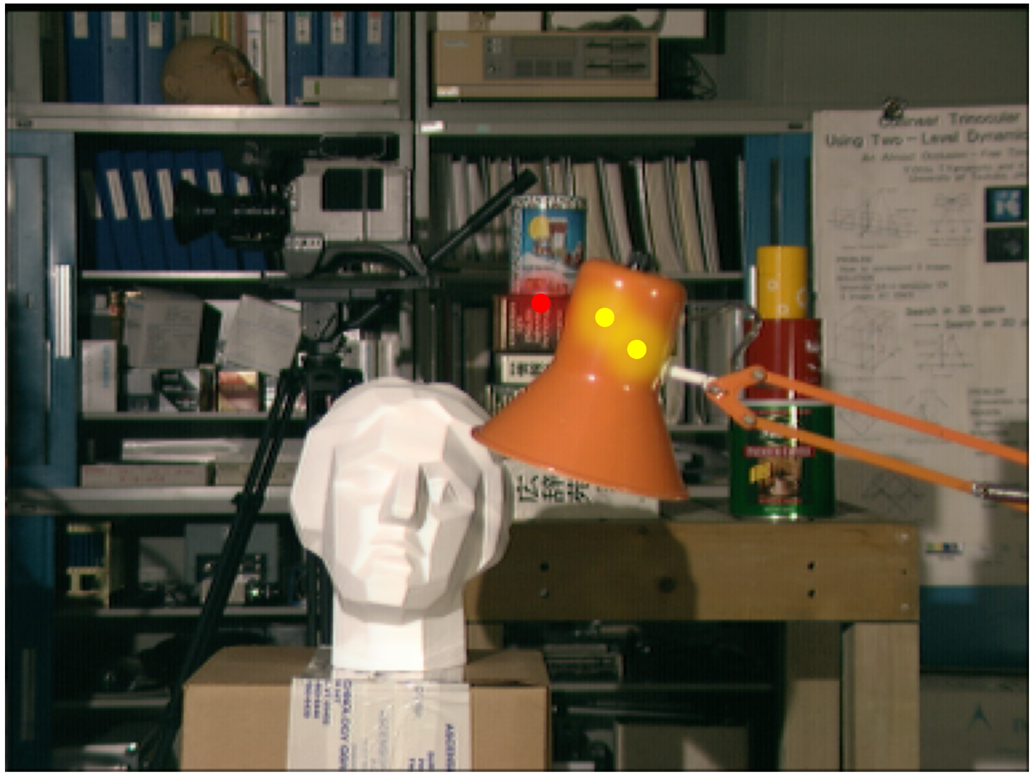
$$\mathbf{d}^* = \arg\min_{\mathbf{d}} \sum_{(i,j)} E_{data}(d(i,j)) + \sum_{(i,j),(k,l)\in\mathcal{N}} E_{smooth}(d(i,j), d(k,l))$$

Based on e.g NCC distance

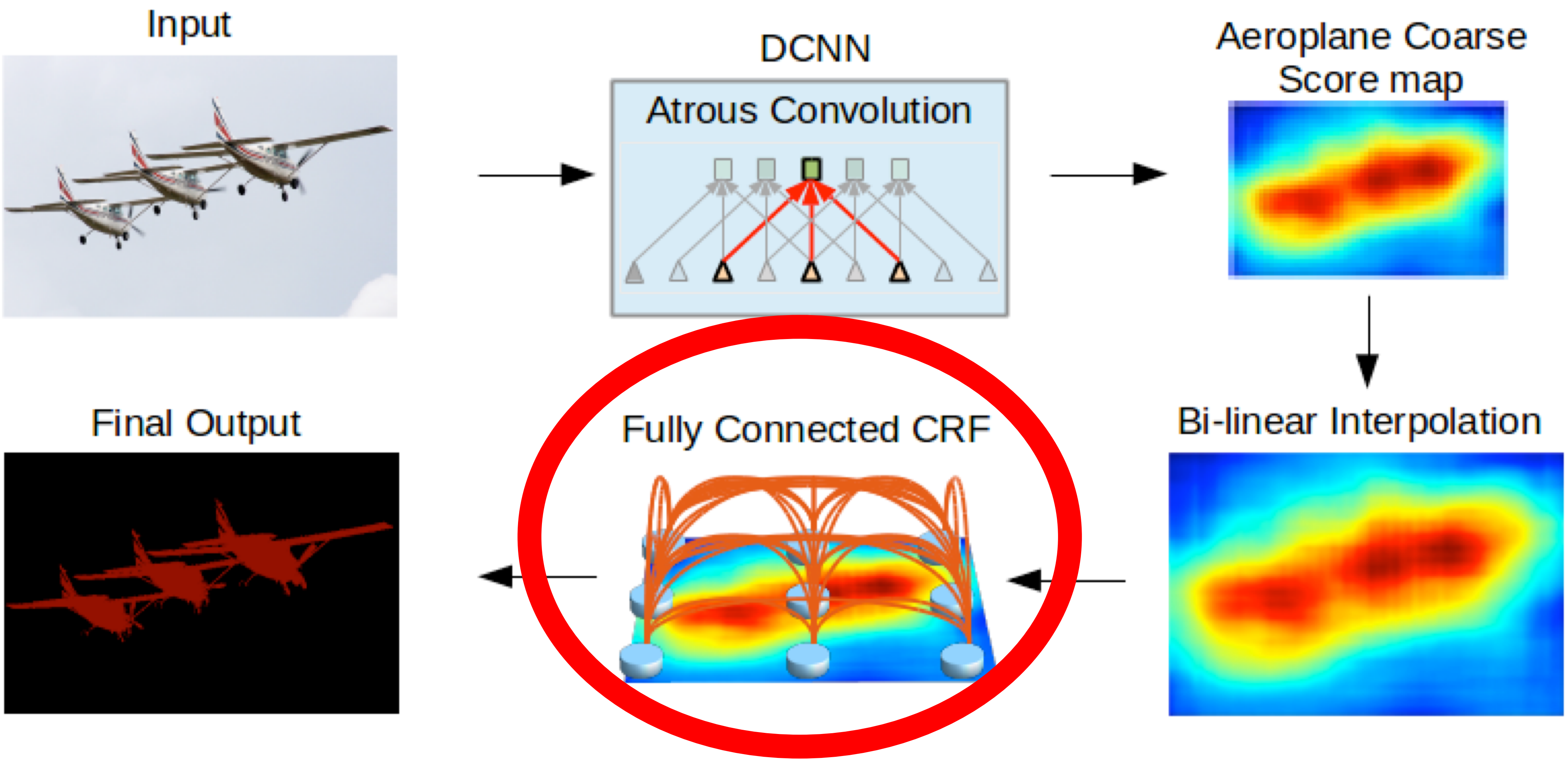
$(d(i,j) - d(k,l))^2$

Aligning depth boundaries to image boundaries

- Some pairs more likely to have same disparity
- $w(i,j)\ (d(i,j) - d(k,l))^2$
- $w(i,j) = 0$ for edges
- *Conditional* Random Field (CRF)



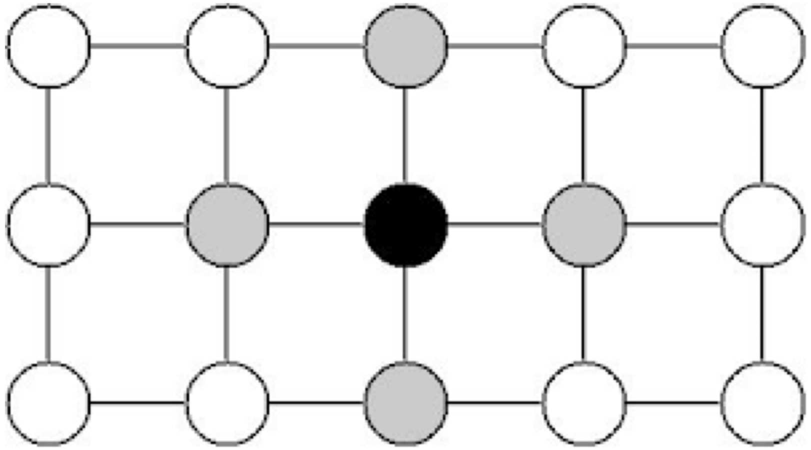
Other applications of MRFs / CRFs



Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs.
Liang-Chieh Chen*, George Papandreou*, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. In *ICLR*, 2015

Optimizing MRFs

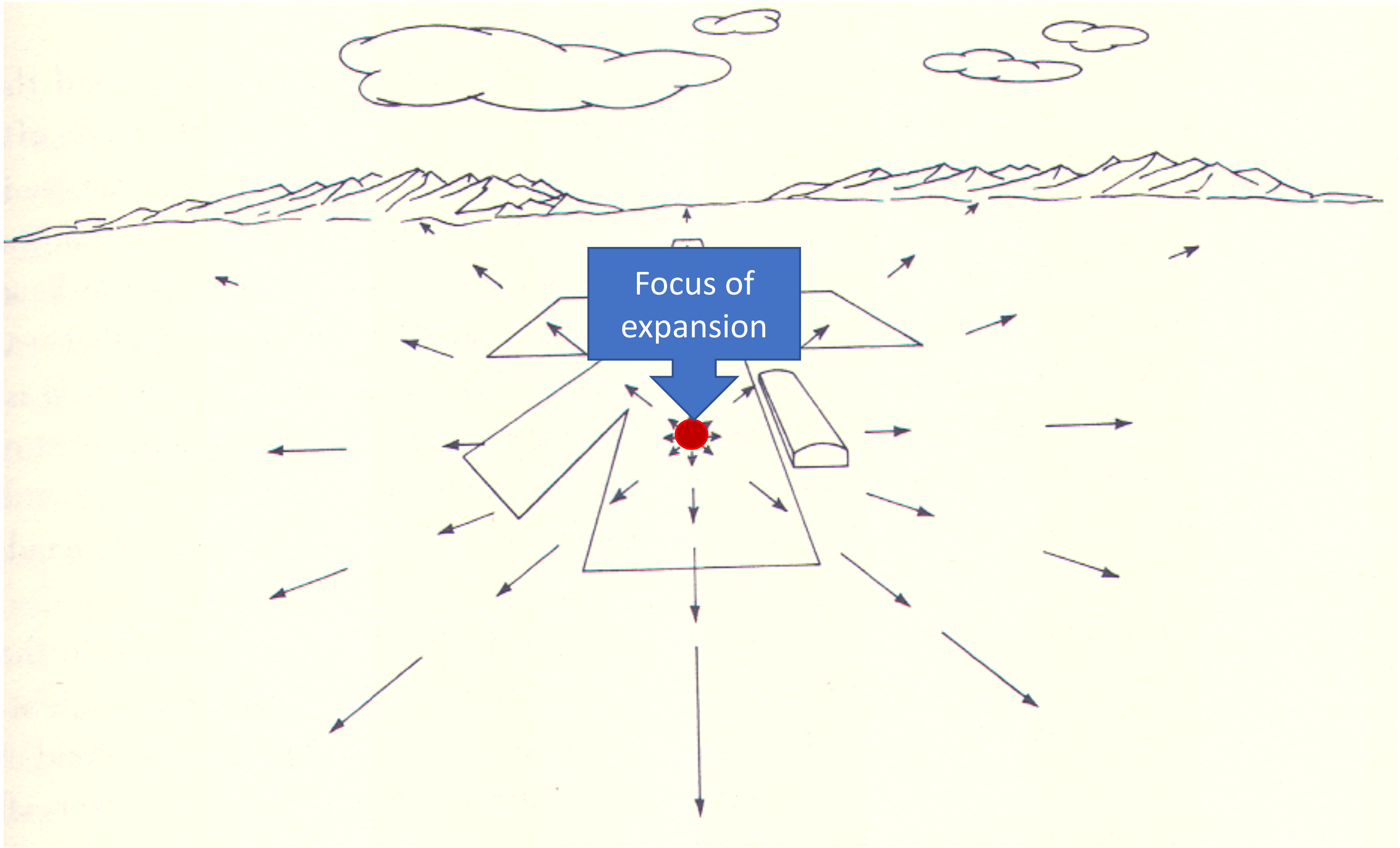
- NP-Hard
- Approximate solutions
  - Message passing
  - Mean field-based inference
  - Graph cut-based solutions



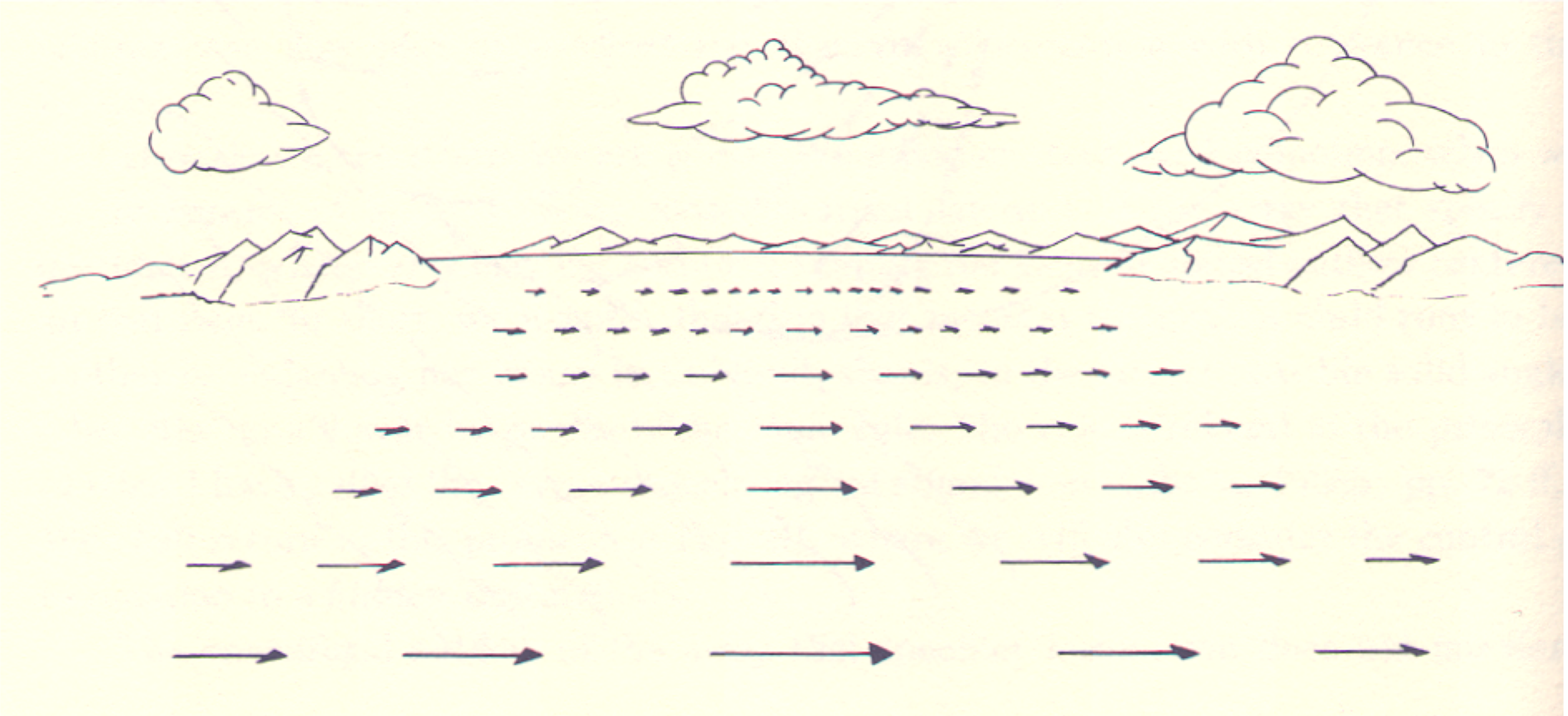
A comparative study of energy minimization methods for markov random fields with smoothness-based priors. Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M. and Rother, C. In *TPAMI,* 2008.

Dense correspondence with MRFs

Optical flow



J. J. Gibson



Optical flow due to camera motion

- Consider camera translating and rotating

$$\mathbf{P} = (X, Y, Z)^T$$

$$x = \frac{X}{Z} \qquad y = \frac{Y}{Z}$$

$$\dot{\mathbf{P}} = -\mathbf{t} - \omega \times \mathbf{P}$$

Optical flow due to camera motion

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \frac{1}{Z}\begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix}\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + \begin{bmatrix} xy & -(1+x^2) & y \\ 1+y^2 & -xy & -x \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

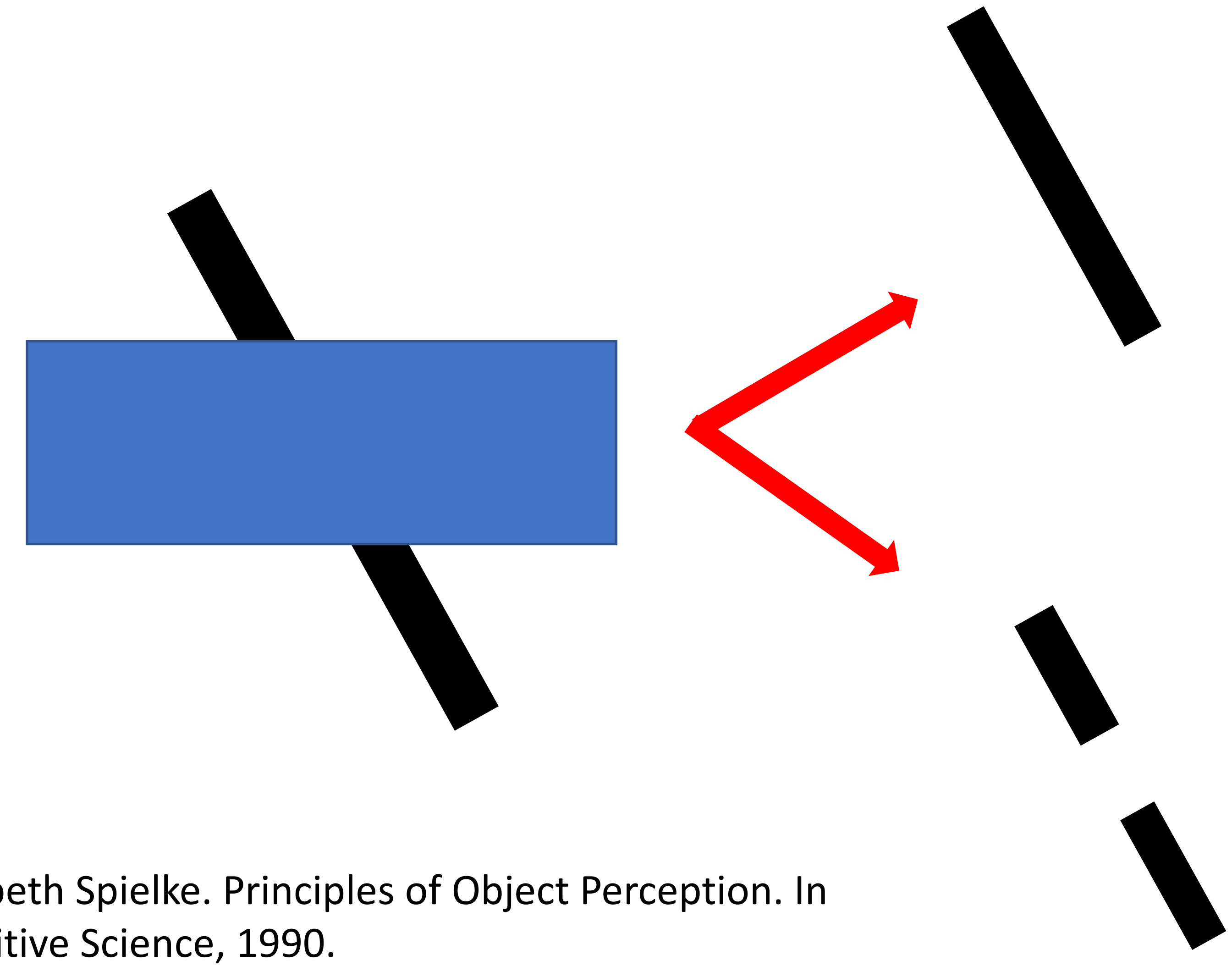
Optical flow for moving scenes

Optical flow for moving scenes

Optical flow for moving scenes

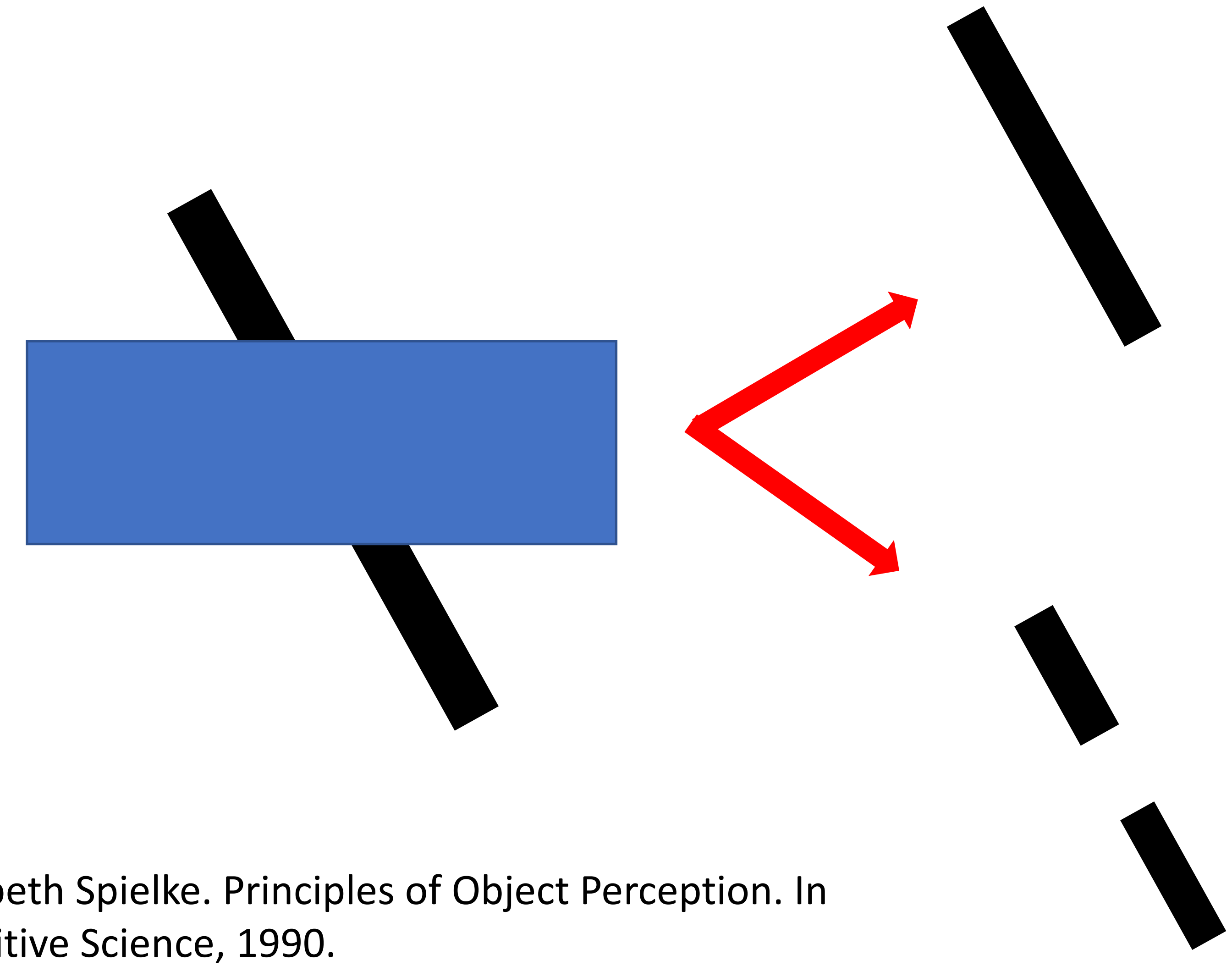
- Optical flow helps *grouping*
- *Gestalt principle of common fate*
  - *Things that move together belong together*

Motion segmentation in humans



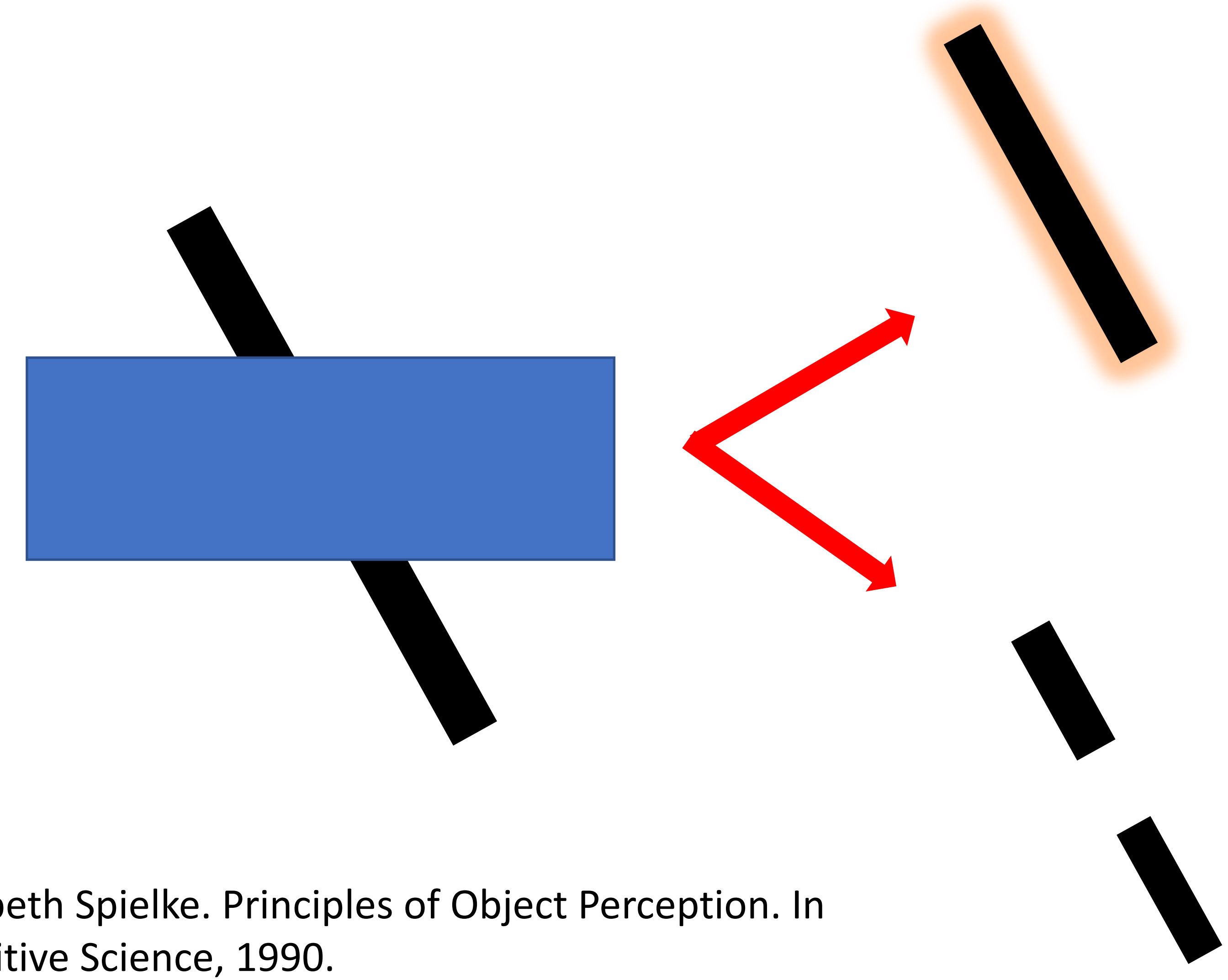
Elizabeth Spielke. Principles of Object Perception. In Cognitive Science, 1990.

Motion segmentation in humans

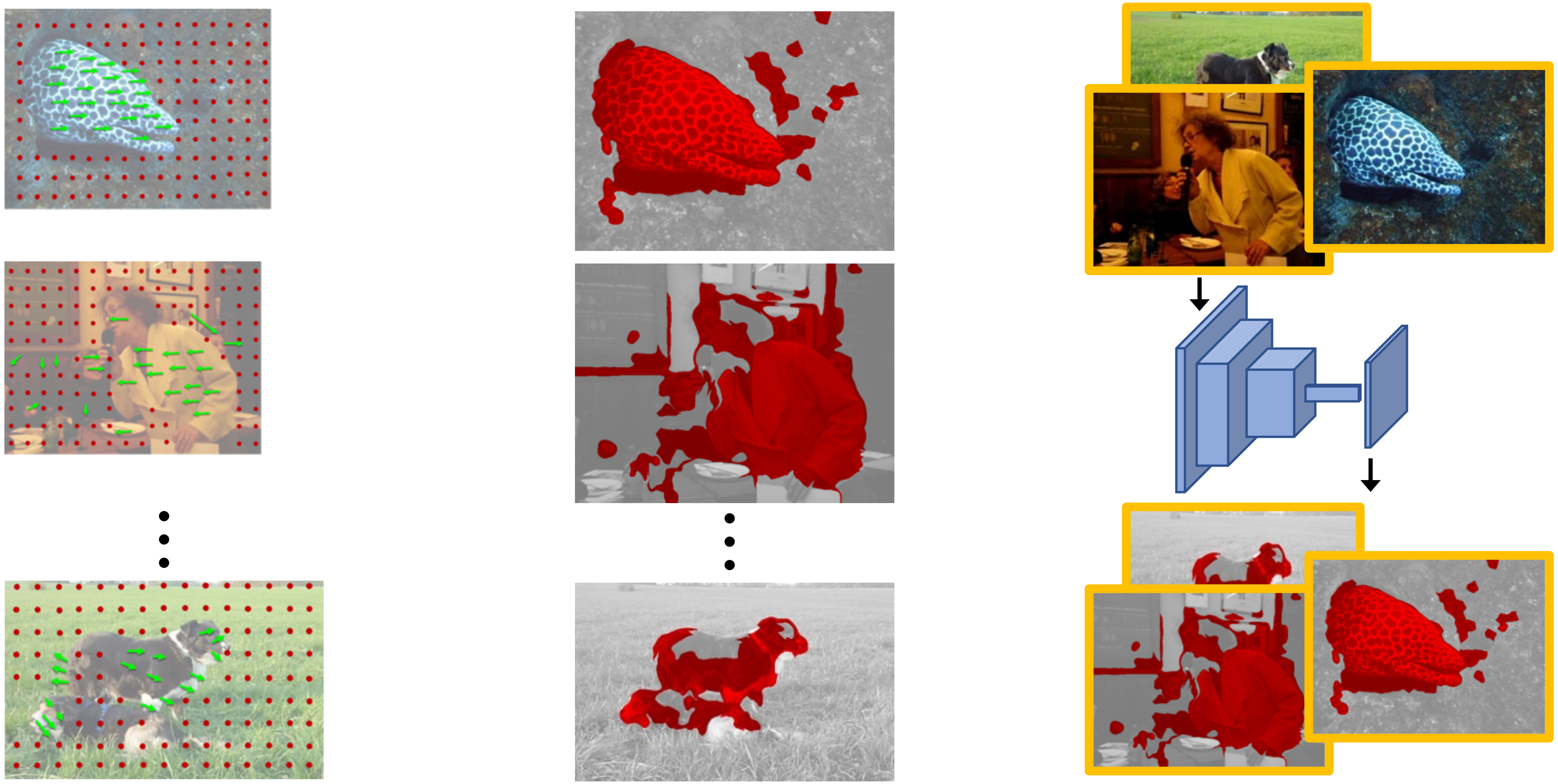


Elizabeth Spielke. Principles of Object Perception. In Cognitive Science, 1990.

Motion segmentation in humans



Elizabeth Spielke. Principles of Object Perception. In Cognitive Science, 1990.



1. Collect videos

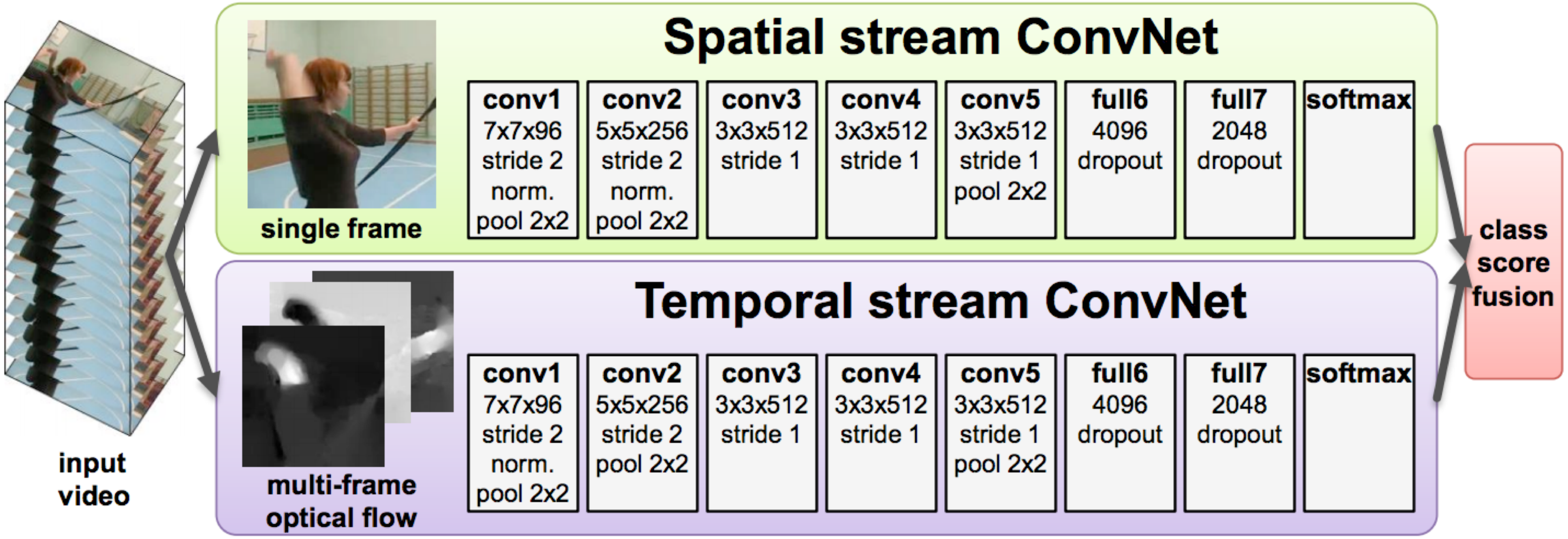
2. Segment using motion

3. Train ConvNet

Learning Features by Watching Objects Move. D. Pathak, R. Girshick, P. Dollar, T. Darrell, B. Hariharan. *CVPR*, 2017.

Optical flow for moving scenes

- Motion is cue for recognition
  - Gestures, actions, …



Two-Stream Convolutional Networks for Action Recognition in Videos. Simonyan and Zisserman. In *NIPS 2014.*

Optical flow for moving scenes

- Motion is cue for recognition
  - Gestures, actions, …

Model	Accuracy
Without optical flow	73.0%
With optical flow	**88.0%**

Two-Stream Convolutional Networks for Action Recognition in Videos. Simonyan and Zisserman. In *NIPS 2014.*

Estimating optical flow

- Yet another correspondence problem!
- But:
  - Bad: scene can move
  - Good: changes are usually small (classic optical flow problem: <1 pixel)

Optical flow constraint equation

- Image intensity *continuous* function of x, y, t
- In time dt, pixel (x,y,t) moves to (x + u dt, y + v dt, t + dt)

$$\min_{u,v}(I(x + u\Delta t, y + v\Delta t, t + \Delta t) - I(x, y, t))^2$$

$$\equiv \min_{u,v}(I(x, y, t) + I_x u\Delta t + I_y v\Delta t + I_t\Delta t - I(x, y, t))^2$$

$$\equiv \min_{u,v}(I_x u\Delta t + I_y v\Delta t + I_t\Delta t)^2$$

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

- Optical flow constraint equation: One equation, two variables

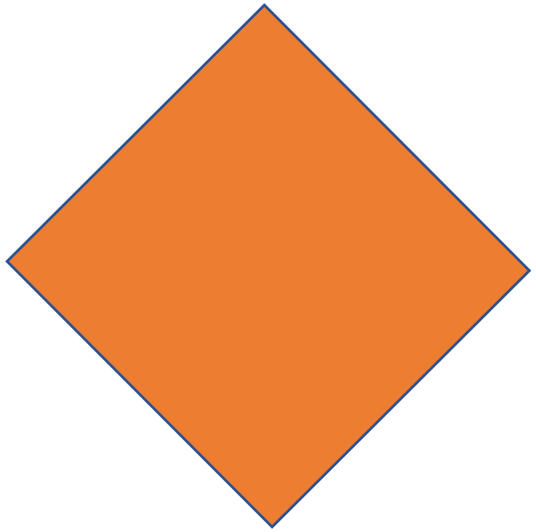
Lucas-Kanade

- Assume all pixels in patch have the same flow
- When will this produce a unique solution?

$$\begin{pmatrix} \nabla I(x_1, y_1)^T \\ \vdots \\ \nabla I(x_n, y_n)^T \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = - \begin{pmatrix} I_t(x_1, y_1) \\ \vdots \\ I_t(x_n, y_n) \end{pmatrix}$$

Aperture problem

Aperture problem



Lucas-Kanade

$$\begin{pmatrix} \nabla I(x_1, y_1)^T \\ \vdots \\ \nabla I(x_n, y_n)^T \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = - \begin{pmatrix} I_t(x_1, y_1) \\ \vdots \\ I_t(x_n, y_n) \end{pmatrix}$$

- Equation of the form Ax = b
- Solve using Normal equations: $x = (A^T A)^{-1} A^T b$
- Need $A^T A$ to be invertible - corners!

Lucas-Kanade

- What if we consider the whole image as one patch?
  - Constant optical flow for the entire image?
- Better: what if we consider flow as a *parametric function* of pixel location?
  - e.g. affine $\begin{bmatrix} u \\ v \end{bmatrix} = A\mathbf{x} + b$
  - More generally: $\begin{bmatrix} u \\ v \end{bmatrix} = f(\mathbf{x}, \theta)$
  - “Motion models”

Lucas-Kanade

$$\min_{\theta} \sum_{\mathbf{x}} (I(\mathbf{x} + f(\mathbf{x}, \theta)dt, t + dt) - I(\mathbf{x}, t))^2$$

- Solve by iterating on θ
- Newton iteration
- Can we remove the parametric assumption?

Baker, Simon, and Iain Matthews. "Lucas-kanade 20 years on: A unifying framework." *International journal of computer vision* 56.3 (2004): 221-255.

Horn-Schunk

$$E(\mathbf{u}, \mathbf{v}) = E_{data}(\mathbf{u}, \mathbf{v}) + E_{smoothness}(\mathbf{u}, \mathbf{v})$$

$$E(\mathbf{u}, \mathbf{v}) = \int\int (I(x + u(x,y)\Delta t, y + v(x,y)\Delta t, t + \Delta t) - I(x,y,t))^2$$

Data

$$+ \alpha(\|\nabla u\|^2 + \|\nabla v\|^2)dxdy$$

Smoothness

Horn-Schunk

$$E(\mathbf{u},\mathbf{v}) = E_{data}(\mathbf{u},\mathbf{v}) + E_{smoothness}(\mathbf{u},\mathbf{v})$$

$$E(\mathbf{u},\mathbf{v}) = \int\int (I_x u + I_y v + I_t)^2 + \alpha(\|\nabla u\|^2 + \|\nabla v\|^2)dxdy$$

Data

Smoothness

Variational minimization

- u and v are *functions*
- Euler-lagrange equations
  - Similar to “gradient=0”

$$\min_q \int L(t, q(t), \dot{q}(dt))dt$$

$$\frac{\partial L}{\partial q} - \frac{d}{dt}\frac{\partial L}{\partial \dot{q}} = 0$$

Variational minimization

$$\min_q \int L(t, q(t), \dot{q}(dt))dt$$

$$\frac{\partial L}{\partial q} - \frac{d}{dt}\frac{\partial L}{\partial \dot{q}} = 0$$

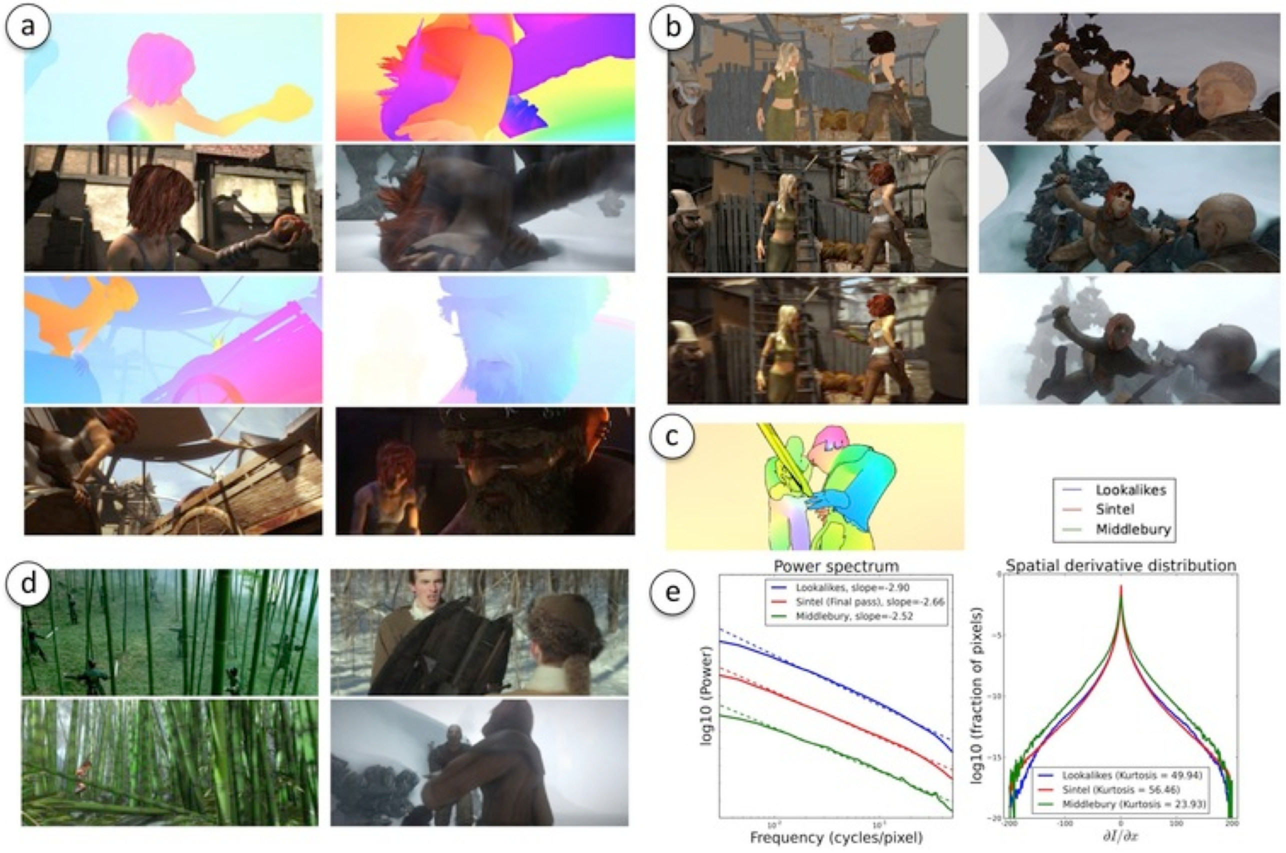
$$\min_{u,v} \int\int f(x, y, u, v, u_x, u_y, v_x, v_y)dxdy$$

$$\frac{\partial f}{\partial u} - \frac{d}{dx}\frac{\partial f}{\partial u_x} - \frac{d}{dy}\frac{\partial f}{\partial u_y} = 0$$

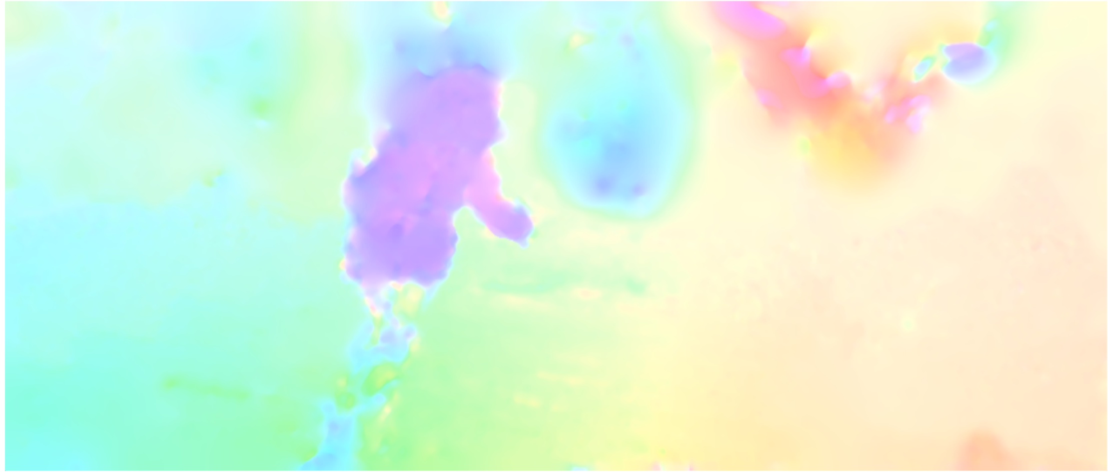
$$\frac{\partial f}{\partial v} - \frac{d}{dx}\frac{\partial f}{\partial v_x} - \frac{d}{dy}\frac{\partial f}{\partial v_y} = 0$$

MPI-Sintel

- Open-source animated movi
  “Sintel”
- “Naturalistic” video
- Ground truth optical flow
- Large motions
- Complex scenes



MPI-Sintel results

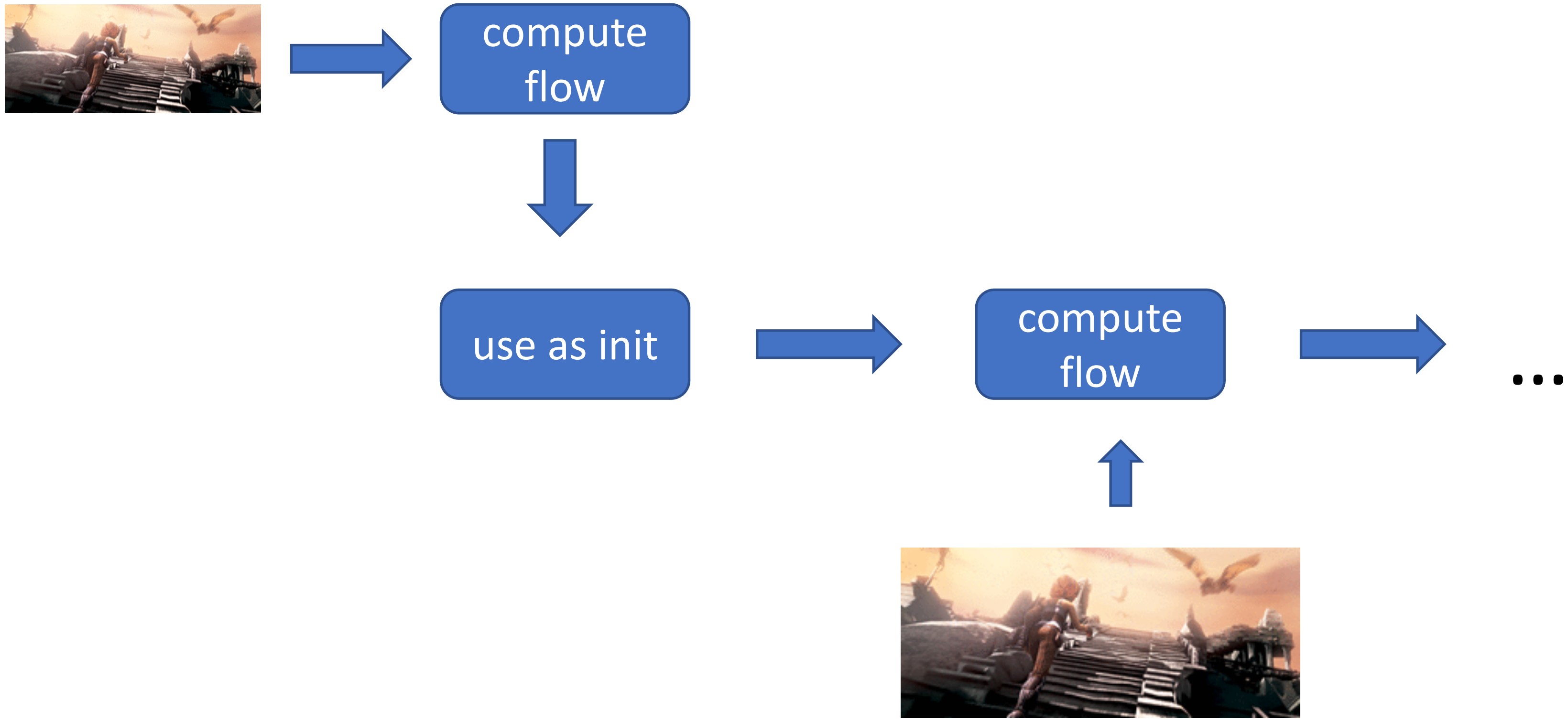


Optical flow with large displacements

- Optical flow constraint equation assumes differential optical flow
- “Large displacement”?
- Key idea: reducing resolution reduces displacement
- Reduce resolution, then upsample?
  - will lose fine details

Optical flow with large displacements

- Key idea 2: Use upsampled flow as *initialization*
- *Changes to initialization will be infinitesimal*



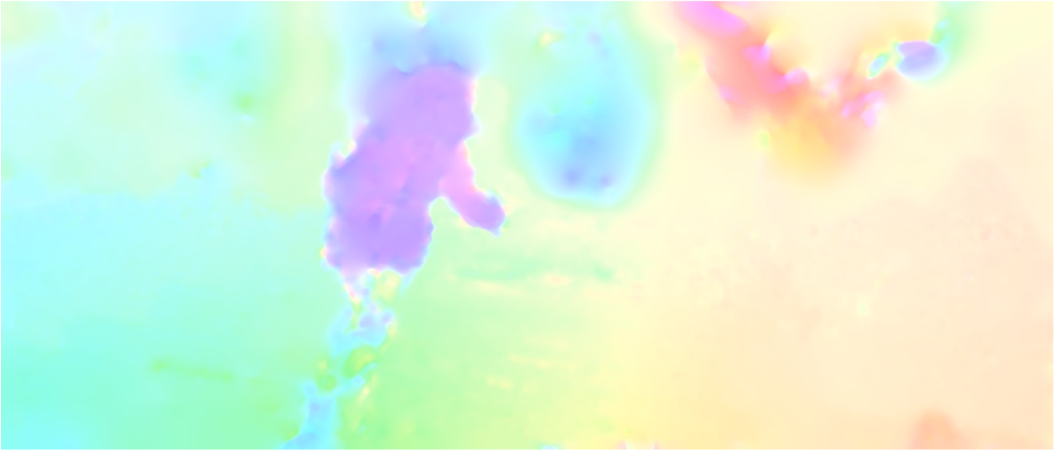
Brox, Thomas, et al. "High accuracy optical flow estimation based on a theory for warping." *Computer Vision-ECCV 2004* (2004)

Optical flow for large displacements

- Horn-schunk variants match using color - Bad!
- Use descriptor matching (e.g. SIFT) on sparse points
- Use smoothness to propagate

Brox, Thomas, Christoph Bregler, and Jitendra Malik. "Large displacement optical flow." *CVPR,* 2009.

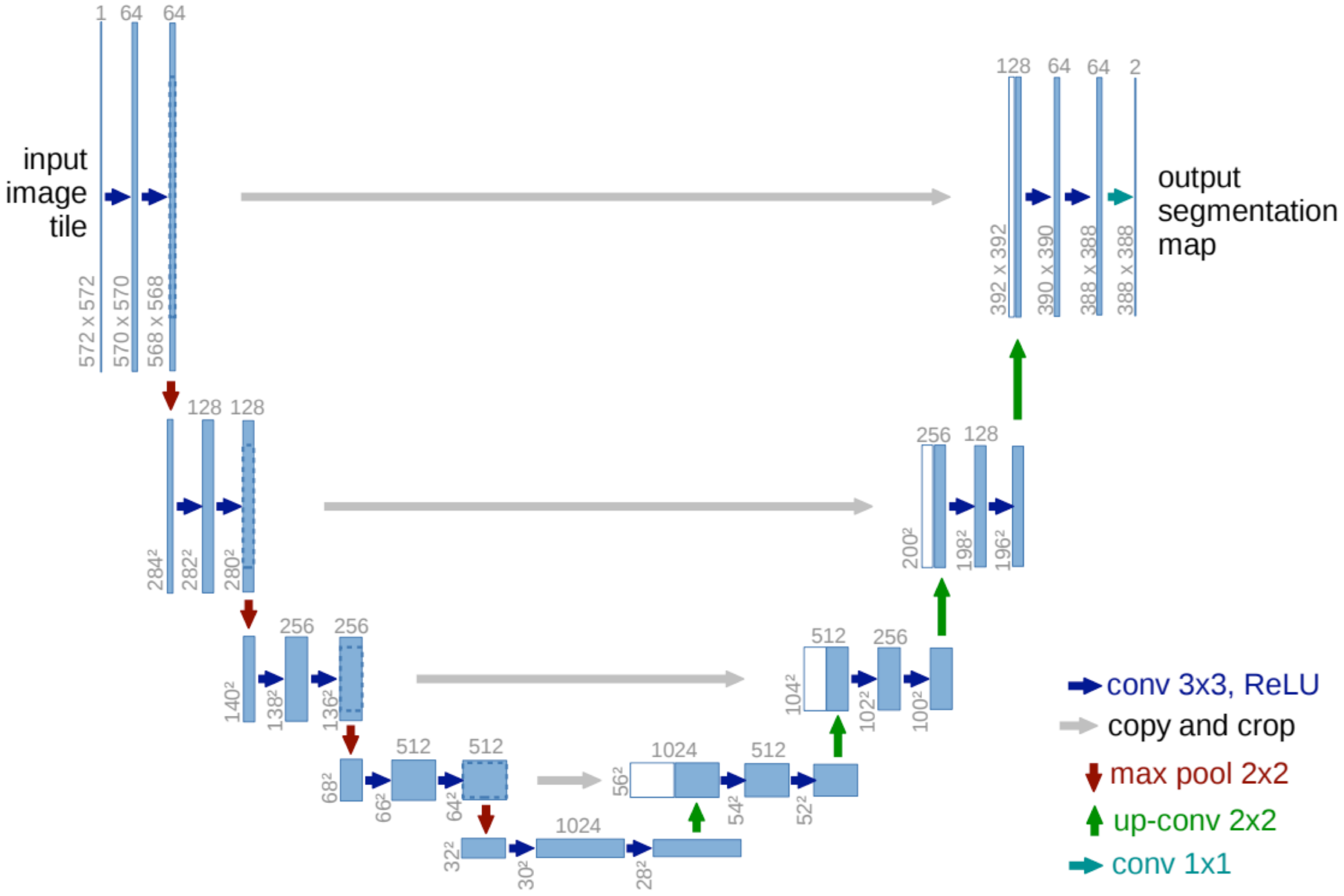
Large displacement optical flow (LDOF)



Coarse-to-fine processing

- A specific instance of a general idea
- Coarse scales:
  - Global / large structures
  - Long-range relationships
  - But: imprecise localization
- Fine scales:
  - Precise localization
  - But: aperture problem
- Idea: start from coarse scales, add fine scale detail

Coarse-to-fine processing



U-Net: Convolutional Networks for Biomedical Image Segmentation. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. In *MICCAI,* 2015.