

# Computer Vision

## Lecture 6 – Applications of Graphical Models

Kumar Bipin

BE, MS, PhD (MMMTU, IISc, IIIT-Hyderabad)

Robotics, Computer Vision, Deep Learning, Machine Learning, System Software



INTERNATIONAL INSTITUTE OF  
INFORMATION TECHNOLOGY  
HYDERABAD

# Agenda

**6.1** Stereo Reconstruction

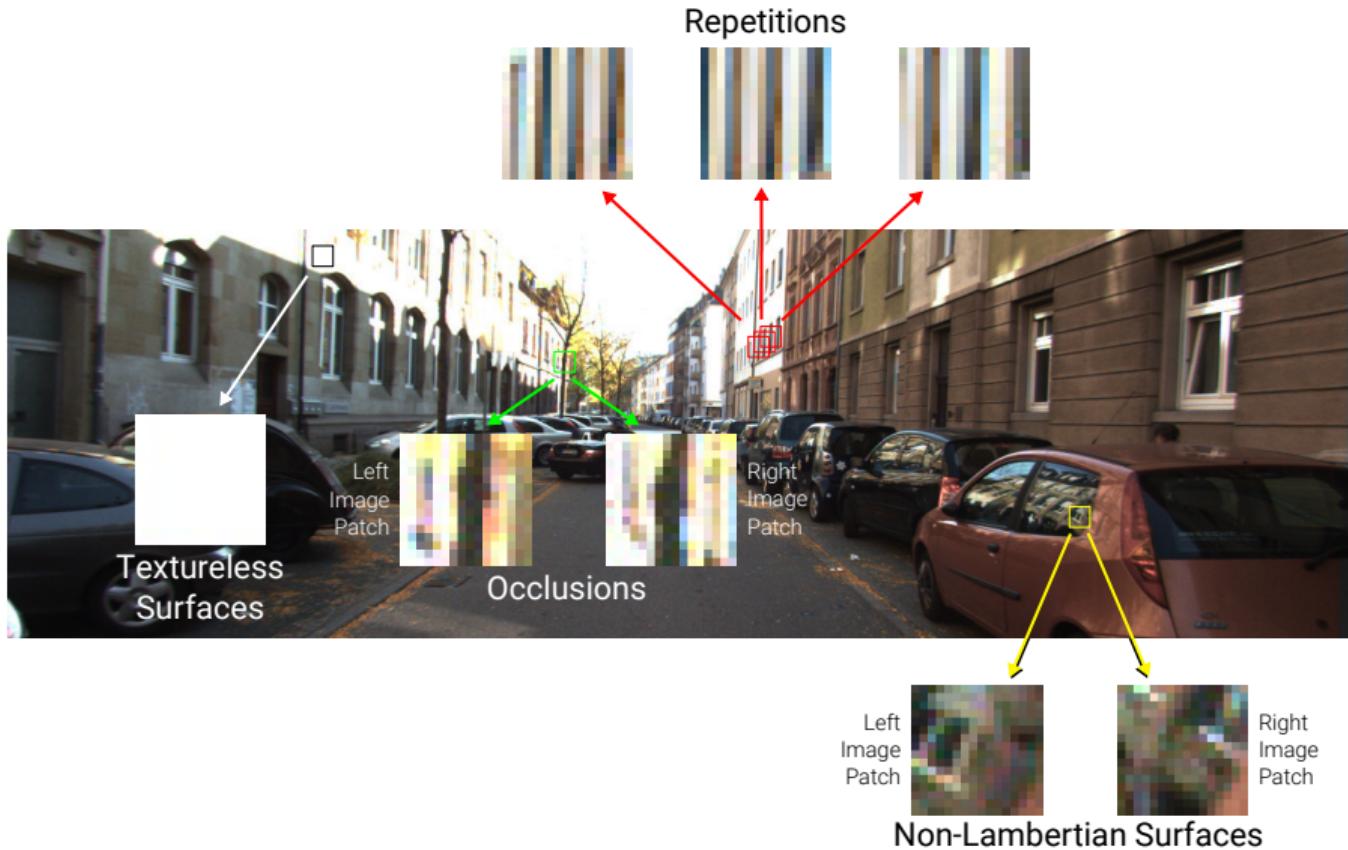
**6.2** Multi-View Reconstruction

**6.3** Optical Flow

# 6.1

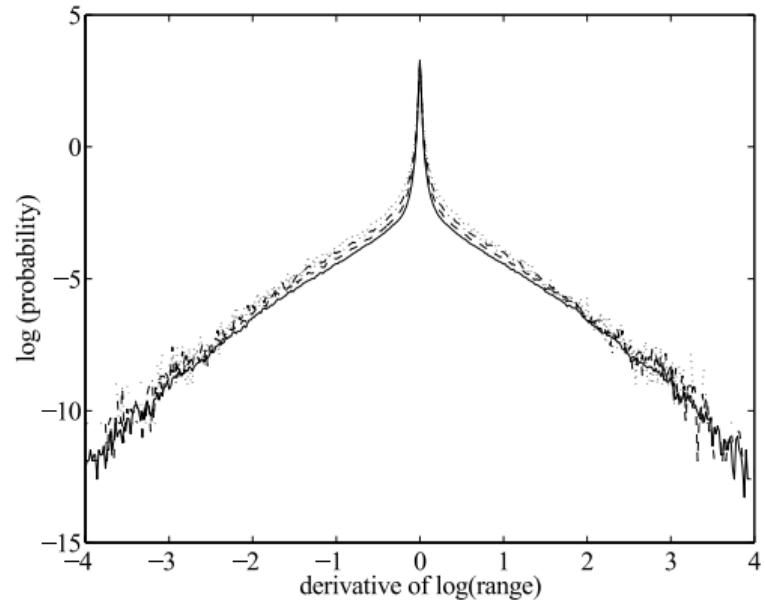
## Stereo Reconstruction

# Stereo Matching Ambiguities



# How does the real world look like?

- ▶ Analyze **real-world statistics** (e.g., Brown range image database)
- ▶ Conclusion: Depth varies slowly except at object discontinuities which are sparse



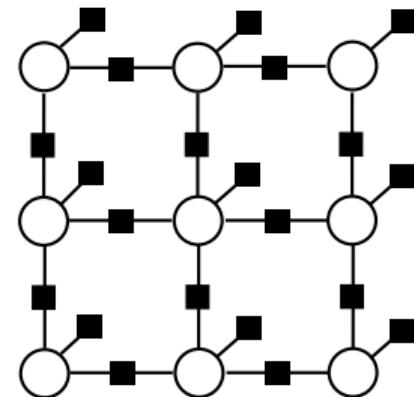
# Stereo MRF

Specify a loopy **Markov Random Field (MRF)** on a grid and solve for the whole disparity map  $\mathbf{D}$  at once (MAP solution = minimum of energy):

$$p(\mathbf{D}) \propto \prod_i f_{data}(d_i) \times \prod_{i \sim j} f_{smooth}(d_i, d_j)$$

Or equivalently:

$$p(\mathbf{D}) \propto \exp \left\{ - \underbrace{\left( \sum_i \psi_{data}(d_i) + \lambda \sum_{i \sim j} \psi_{smooth}(d_i, d_j) \right)}_{\text{Energy } E} \right\}$$

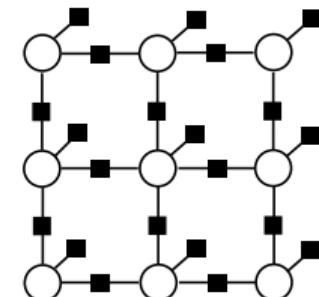


- ▶  $\psi(\cdot) := -\log f(\cdot)$ ,  $i \sim j$ : neighboring pixels on a 4-connected grid
- ▶  $\psi_{data}(d)$ : unary terms, and  $\psi_{smooth}(d, d')$  pairwise terms

# Stereo MRF

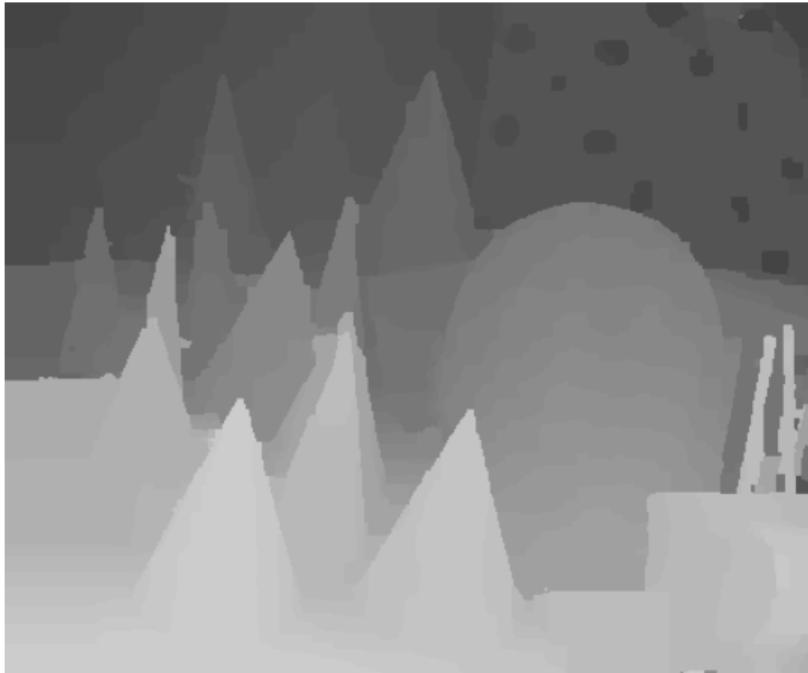
Markov Random Field (MRF) from last slide:

$$p(\mathbf{D}) \propto \exp \left\{ - \underbrace{\left( \sum_i \psi_{data}(d_i) + \lambda \sum_{i \sim j} \psi_{smooth}(d_i, d_j) \right)}_{\text{Energy } E} \right\}$$

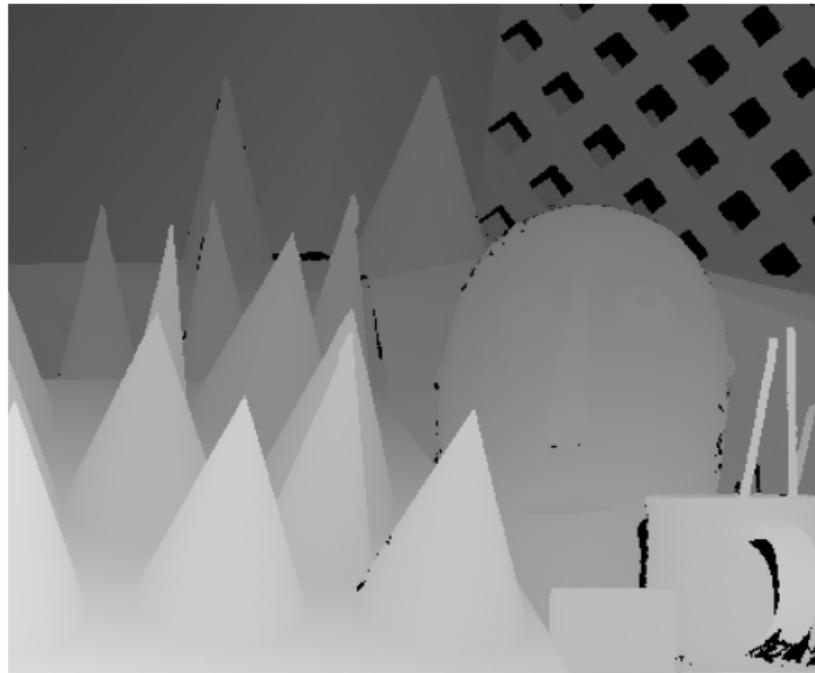


- ▶  $i \sim j$ : neighboring pixels on a 4-connected grid
- ▶ **Unary terms:** Matching cost  $\psi_{data}(d)$
- ▶ **Pairwise terms:** Encourages smoothness between adjacent pixels, e.g.:
  - ▶ Potts model:  $\psi_{smooth}(d, d') = [d \neq d']$
  - ▶ Truncated  $l_1$  penalty:  $\psi_{smooth}(d, d') = \min(|d - d'|, \tau)$
- ▶ The parameter  $\lambda$  is added to control the strength of the smoothness prior
- ▶ Solve MRF approximately for disparity map  $\mathbf{D}$  using BP, graph cuts, etc.

# Stereo MRF – Results



Inference Results



Ground Truth

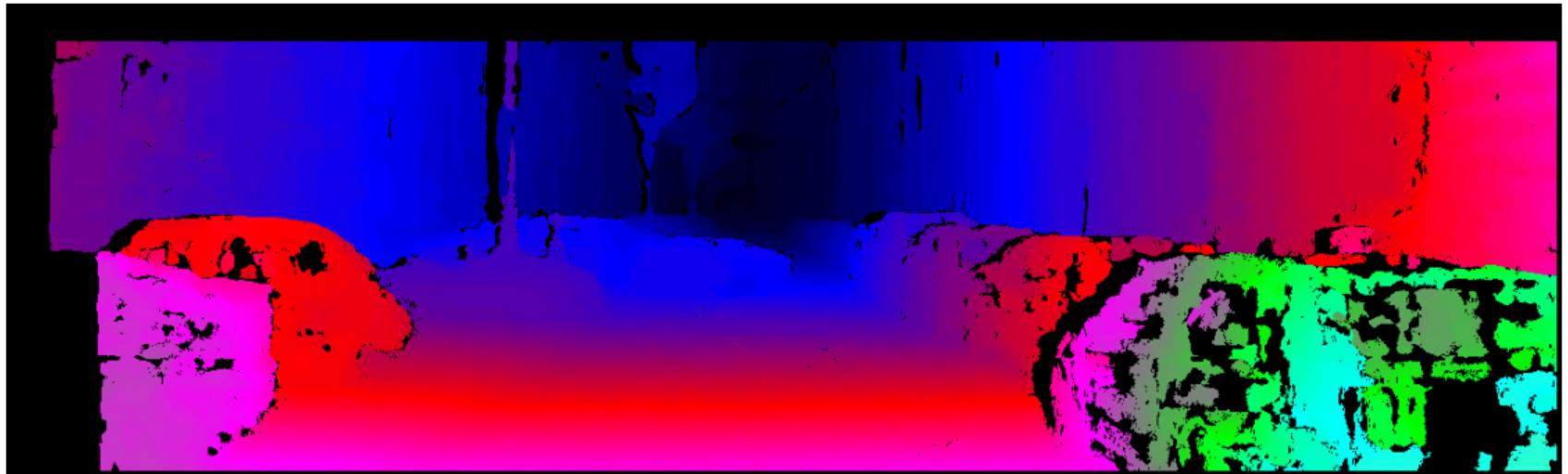
# Non-local Priors

# Non-local Priors



$$E(\mathbf{D}) = \underbrace{\sum_i \psi_i^A(d_i)}_{\text{Appearance}} + \lambda \underbrace{\sum_{i \sim j} \psi_{ij}^S(d_i, d_j)}_{\text{Smoothness}}$$

# Non-local Priors



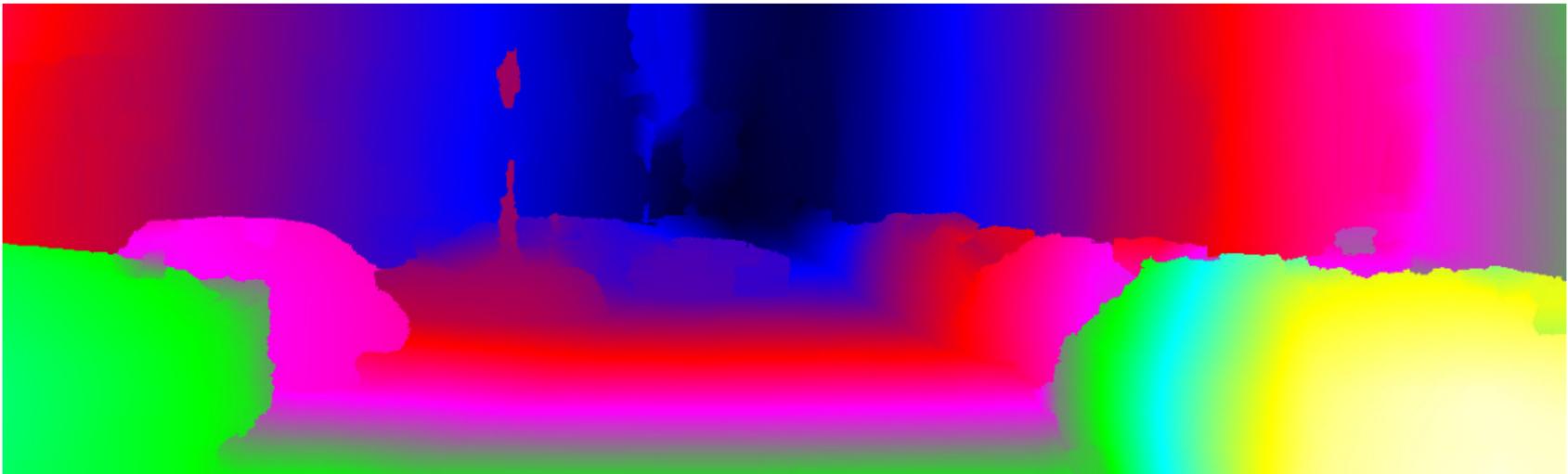
$$E(\mathbf{D}) = \underbrace{\sum_i \psi_i^A(d_i)}_{\text{Appearance}} + \lambda \underbrace{\sum_{i \sim j} \psi_{ij}^S(d_i, d_j)}_{\text{Smoothness}}$$

# Non-local Priors



$$E(\mathbf{D}, \mathbf{O}) = \underbrace{\sum_i \psi_i^{\mathcal{A}}(d_i)}_{\text{Appearance}} + \lambda_S \underbrace{\sum_{i \sim j} \psi_{ij}^{\mathcal{S}}(d_i, d_j)}_{\text{Smoothness}} + \lambda_{\mathcal{O}} \underbrace{\sum_k \psi_k^{\mathcal{O}}(o_k)}_{\text{Object Semantics}} + \lambda_C \underbrace{\sum_k \sum_i \psi_{ki}^{\mathcal{C}}(o_k, d_i)}_{\text{3D Consistency}}$$

# Non-local Priors



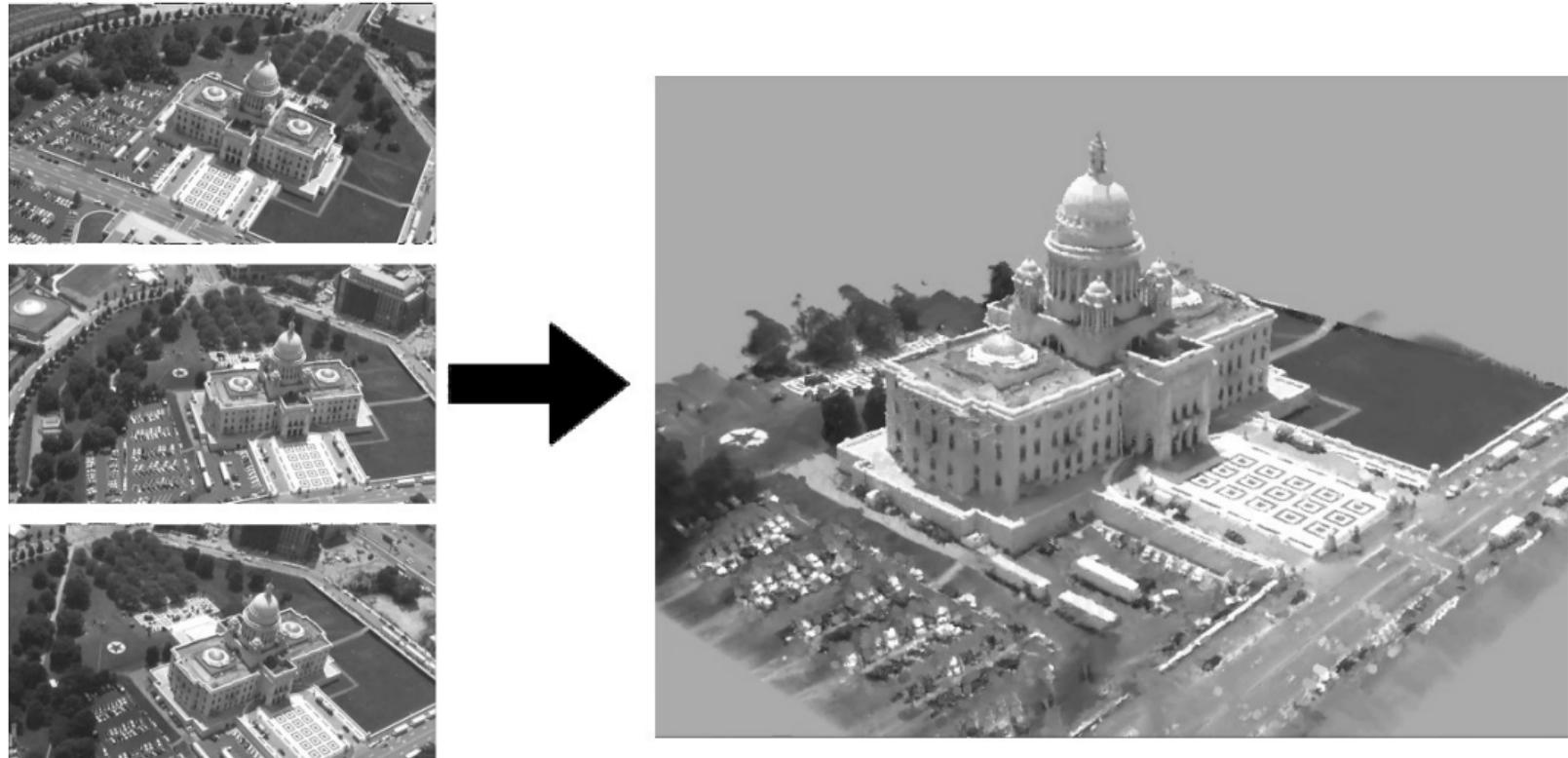
$$E(\mathbf{D}, \mathbf{O}) = \underbrace{\sum_i \psi_i^{\mathcal{A}}(d_i)}_{\text{Appearance}} + \lambda_S \underbrace{\sum_{i \sim j} \psi_{ij}^{\mathcal{S}}(d_i, d_j)}_{\text{Smoothness}} + \underbrace{\lambda_{\mathcal{O}} \sum_k \psi_k^{\mathcal{O}}(o_k)}_{\text{Object Semantics}} + \lambda_C \underbrace{\sum_k \sum_i \psi_{ki}^{\mathcal{C}}(o_k, d_i)}_{\text{3D Consistency}}$$

# Stereo Reconstruction Summary

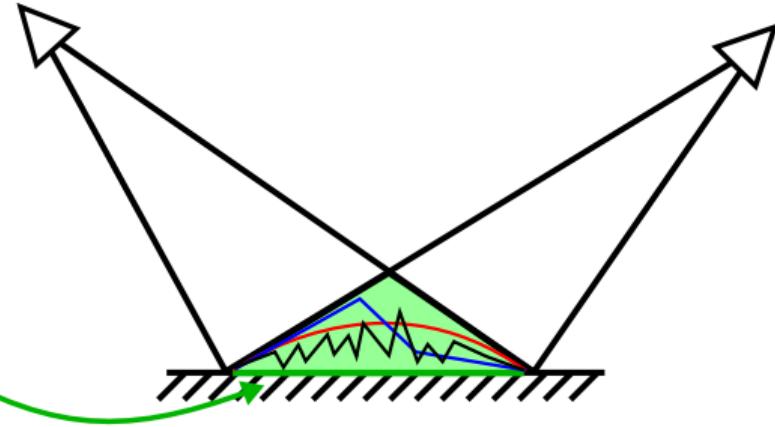
- ▶ Block matching suffers from **ambiguities**
- ▶ Choosing window size is problematic (tradeoff)
- ▶ Incorporating **smoothness constraints** can resolve some of the ambiguities and allows for choosing small windows (no bleeding artifacts)
- ▶ Can be formulated as **MAP inference in a discrete MRF**
- ▶ MAP solution can be obtained using belief propagation, graph cuts, etc.
- ▶ Integrating recognition cues can further regularize the problem



# Probabilistic Dense Multi-View 3D Reconstruction



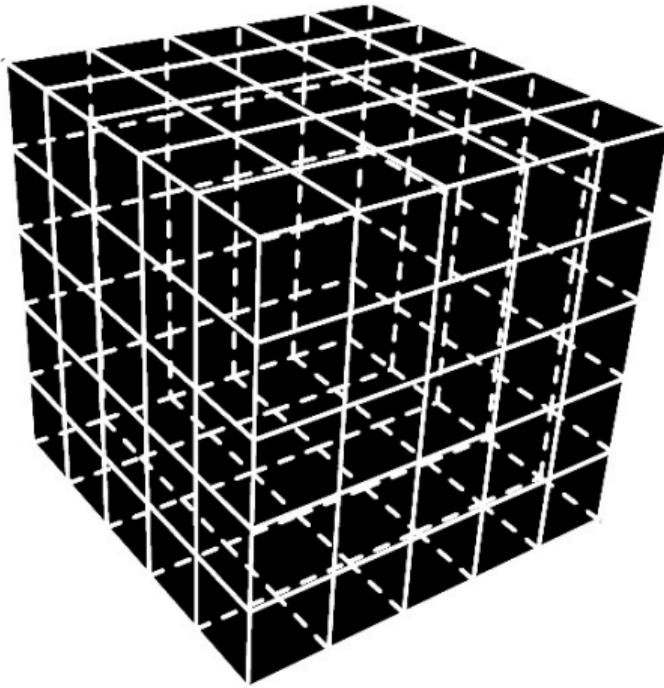
# Reconstruction Ambiguities



- ▶ Image-based 3D reconstruction is a highly **ill-posed** problem  
⇒ Coping with and exposing **uncertainty** is essential

**Can we formulate dense 3D reconstruction in a probabilistic way?**

# Representation



- ▶ **Voxel occupancy:**

$$o_i = \begin{cases} 1 & \text{if voxel } i \text{ is occupied} \\ 0 & \text{if voxel } i \text{ is empty} \end{cases}$$

- ▶ **Voxel appearance:**

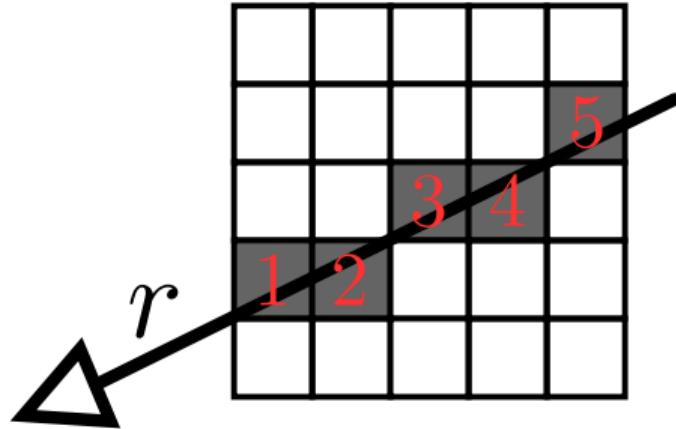
$$a_i \in \mathbb{R}$$

- ▶ **Shorthand notation:**

$$\mathbf{o}_r = \{o_1^r, \dots, o_{N_r}^r\}$$

$$\mathbf{a}_r = \{a_1^r, \dots, a_{N_r}^r\}$$

# Representation



- ▶ **Voxel occupancy:**

$$o_i = \begin{cases} 1 & \text{if voxel } i \text{ is occupied} \\ 0 & \text{if voxel } i \text{ is empty} \end{cases}$$

- ▶ **Voxel appearance:**

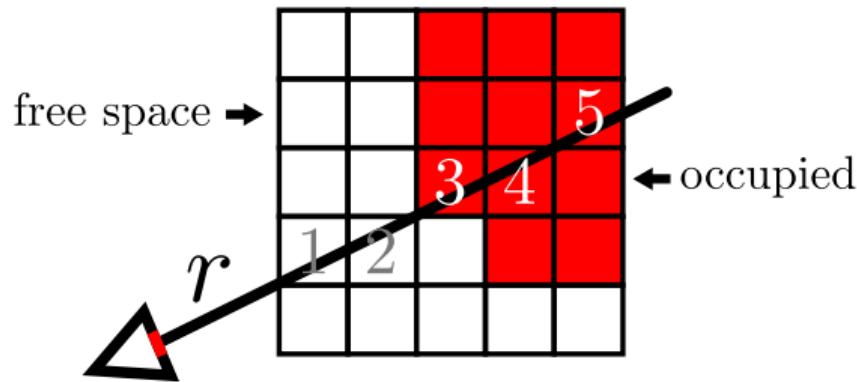
$$a_i \in \mathbb{R}$$

- ▶ **Shorthand notation:**

$$\mathbf{o}_r = \{o_1^r, \dots, o_{N_r}^r\}$$

$$\mathbf{a}_r = \{a_1^r, \dots, a_{N_r}^r\}$$

# Image Formation Process



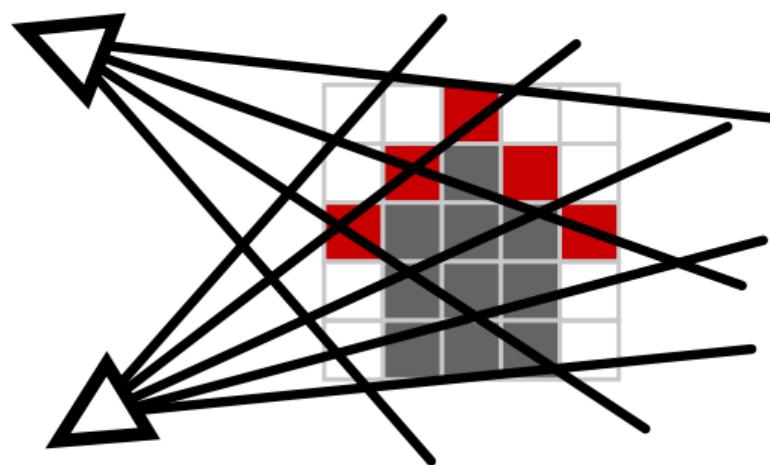
$$I_r = \sum_{i=1}^N o_i \prod_{j < i} (1 - o_j) a_i$$

- $I_r$ : intensity at pixel  $r$
- $o_i$ : occupancy of voxel  $i$
- $a_i$ : appearance of voxel  $i$

# Probabilistic Model

## Joint Distribution:

$$p(\mathbf{O}, \mathbf{A}) = \frac{1}{Z} \prod_{v \in \mathcal{V}} \underbrace{\varphi_v(o_v)}_{\text{unary}} \prod_{r \in \mathcal{R}} \underbrace{\psi_r(\mathbf{o}_r, \mathbf{a}_r)}_{\text{ray}}$$



# Probabilistic Model

## Joint Distribution:

$$p(\mathbf{O}, \mathbf{A}) = \frac{1}{Z} \prod_{v \in \mathcal{V}} \underbrace{\varphi_v(o_v)}_{\text{unary}} \prod_{r \in \mathcal{R}} \underbrace{\psi_r(\mathbf{o}_r, \mathbf{a}_r)}_{\text{ray}}$$

## Unary Potentials:

$$\varphi_v(o_v) = \gamma^{o_v} (1 - \gamma)^{1-o_v}$$

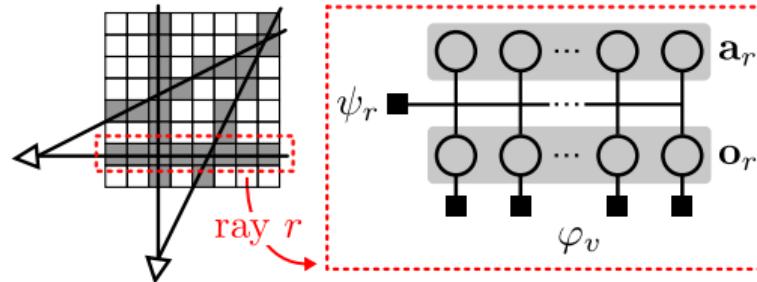
- Most voxels are empty  $\Rightarrow \gamma < 0.5$

# Probabilistic Model

## Joint Distribution:

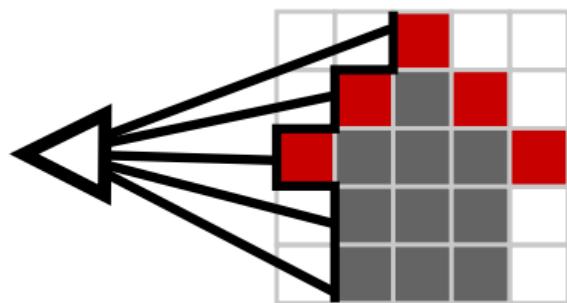
$$p(\mathbf{O}, \mathbf{A}) = \frac{1}{Z} \prod_{v \in \mathcal{V}} \underbrace{\varphi_v(o_v)}_{\text{unary}} \prod_{r \in \mathcal{R}} \underbrace{\psi_r(\mathbf{o}_r, \mathbf{a}_r)}_{\text{ray}}$$

## Ray Potentials:

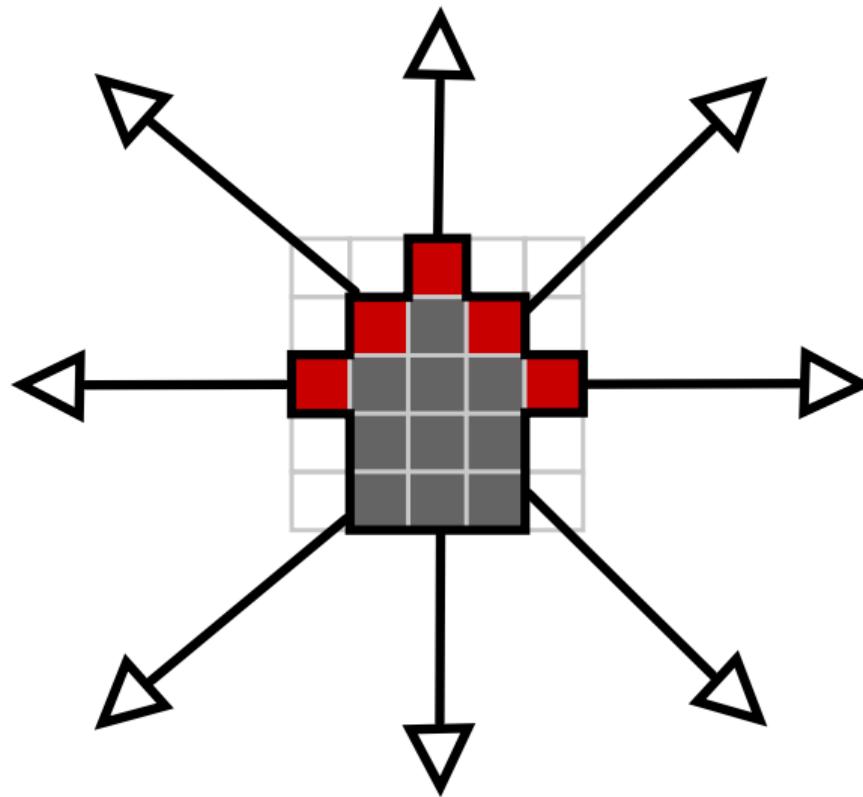


$$\psi_r(\mathbf{o}_r, \mathbf{a}_r) = \sum_{i=1}^{N_r} o_i^r \prod_{j < i} (1 - o_j^r) \underbrace{\mathcal{N}(a_i^r | I_r, \sigma)}_{\text{Gaussian Noise}}$$

# 3D Reconstruction

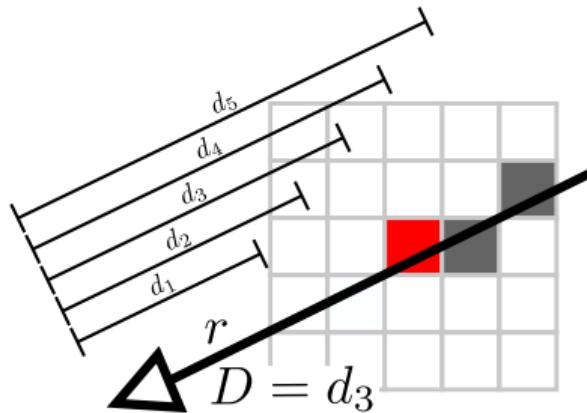


# 3D Reconstruction



# Bayes Optimal Depth Estimation

- ▶ Consider a single ray  $r$  in space
- ▶ Let  $d_k$  be the distance from the camera to voxel  $k$  along ray  $r$
- ▶ Depth  $D \in \{d_1, \dots, d_N\}$ : distance to closest occupied voxel



# Bayes Optimal Depth Estimation

- ▶ Consider a single ray  $r$  in space
- ▶ Let  $d_k$  be the distance from the camera to voxel  $k$  along ray  $r$
- ▶ Depth  $D \in \{d_1, \dots, d_N\}$ : distance to closest occupied voxel
- ▶ **Optimal depth estimate:**

$$\begin{aligned} D^* &= \underset{D'}{\operatorname{argmin}} \text{ Risk}(D') \\ &= \underset{D'}{\operatorname{argmin}} \mathbb{E}_{p(D)} [\Delta(D, D')] \\ &= \begin{cases} \text{mean}(\textcolor{red}{p(D)}) & \text{if } \Delta(D, D') = (D - D')^2 \\ \text{median}(\textcolor{red}{p(D)}) & \text{if } \Delta(D, D') = |D - D'| \end{cases} \end{aligned}$$

- ▶ Requires **marginal depth distribution**  $p(D)$  along each ray

# Depth Distribution for Single Ray

Let:

$$o_1 = 0 \quad \dots \quad o_{k-1} = 0 \quad o_k = 1$$

Then:

$$p(D = d_k) = \underbrace{\sum_{\mathbf{o}_{>k}} \int_{\mathbf{a}} p(\mathbf{o}, \mathbf{a})}_{=p(o_1, \dots, o_k)}$$

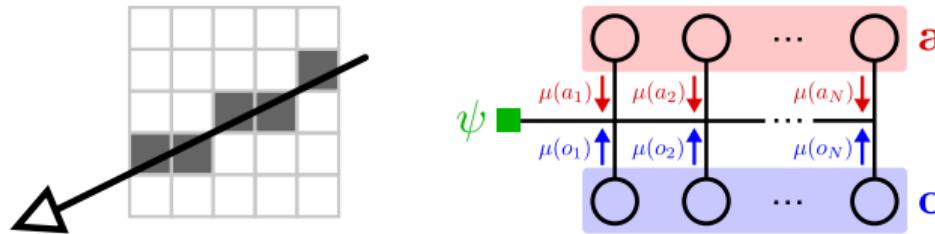
# Depth Distribution for Single Ray

Let:

$$o_1 = 0 \quad \dots \quad o_{k-1} = 0 \quad o_k = 1$$

Then:

$$p(D = d_k) \propto \sum_{\mathbf{o}_{>k}} \int_{\mathbf{a}} \psi(\mathbf{o}, \mathbf{a}) \prod_i \mu(o_i) \prod_i \mu(a_i)$$



Marginal = Product of Factor & Incoming Messages

# Depth Distribution for Single Ray

Let:

$$o_1 = 0 \quad \dots \quad o_{k-1} = 0 \quad o_k = 1$$

Then:

$$p(D = d_k) \propto \sum_{\mathbf{o}_{>k}} \int_{\mathbf{a}} \psi(\mathbf{o}, \mathbf{a}) \prod_i \mu(o_i) \prod_i \mu(a_i)$$

$$\boxed{\psi(\mathbf{o}, \mathbf{a}) = \underbrace{\sum_i o_i \prod_{j < i} (1 - o_j) \mathcal{N}(a_i | I, \sigma)}_{= \mathcal{N}(a_k | I, \sigma)}}$$

# Depth Distribution for Single Ray

Let:

$$o_1 = 0 \quad \dots \quad o_{k-1} = 0 \quad o_k = 1$$

Then:

$$p(D = d_k) \propto \sum_{\mathbf{o}_{>k}} \int_{\mathbf{a}} \mathcal{N}(a_k | I, \sigma) \prod_i \mu(o_i) \prod_i \mu(a_i)$$

# Depth Distribution for Single Ray

Let:

$$o_1 = 0 \quad \dots \quad o_{k-1} = 0 \quad o_k = 1$$

Then:

$$\begin{aligned} p(D = d_k) &\propto \sum_{\mathbf{o}_{>k}} \int_{\mathbf{a}} \mathcal{N}(a_k | I, \sigma) \prod_{i>k} \mu(o_i) \prod_i \mu(a_i) \\ &\quad \times \prod_{i \leq k} \mu(o_i) \end{aligned}$$

# Depth Distribution for Single Ray

Let:

$$o_1 = 0 \quad \dots \quad o_{k-1} = 0 \quad o_k = 1$$

Then:

$$\begin{aligned} p(D = d_k) &\propto \sum_{\substack{\mathbf{o}_{>k} \\ \mathbf{a}_{\neq k}}} \int \prod_{i>k} \mu(o_i) \prod_{i \neq k} \mu(a_i) \\ &\quad \times \prod_{i \leq k} \mu(o_i) \int_{a_k} \mathcal{N}(a_k | I, \sigma) \mu(a_k) \end{aligned}$$

# Depth Distribution for Single Ray

Let:

$$o_1 = 0 \quad \dots \quad o_{k-1} = 0 \quad o_k = 1$$

Then:

$$\begin{aligned} p(D = d_k) &\propto \underbrace{\sum_{\mathbf{o}_{>k}} \int_{\mathbf{a}_{\neq k}} \prod_{i>k} \mu(o_i) \prod_{i\neq k} \mu(a_i)}_{=1} \\ &\times \prod_{i\leq k} \mu(o_i) \int_{a_k} \mathcal{N}(a_k | I, \sigma) \mu(a_k) \end{aligned}$$

# Depth Distribution for Single Ray

Let:

$$o_1 = 0 \quad \dots \quad o_{k-1} = 0 \quad o_k = 1$$

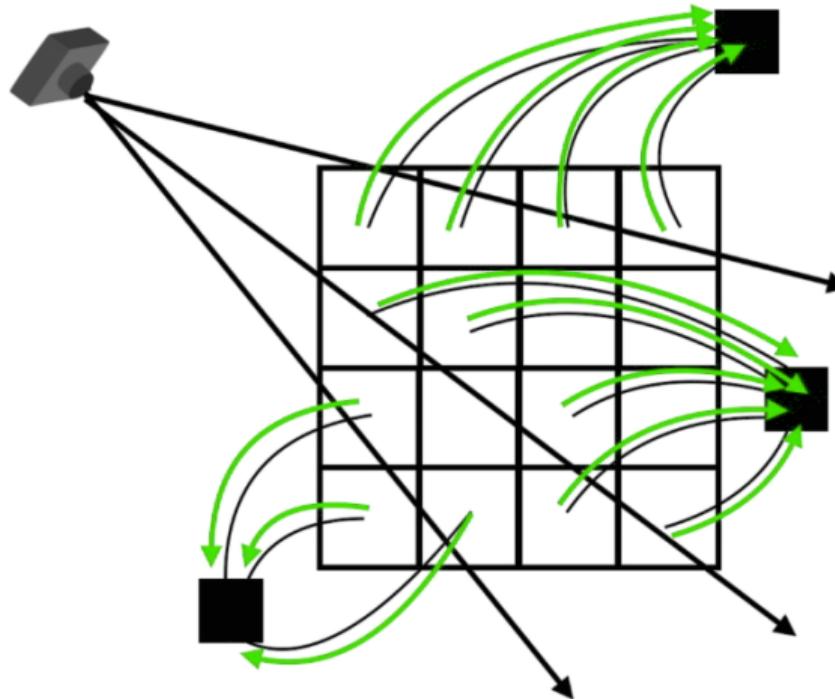
Then:

$$p(D = d_k) \propto \prod_{i \leq k} \mu(o_i) \int_{a_k} \mathcal{N}(a_k | I, \sigma) \mu(a_k)$$

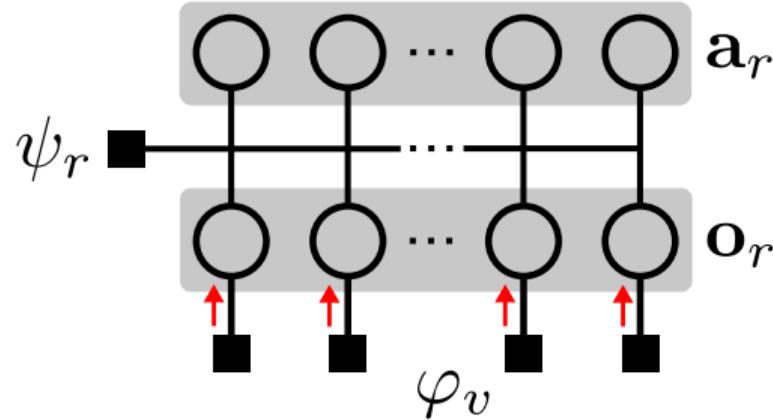
**Intuition:** Depth  $D = d_k \Leftrightarrow$  Voxel  $k$  is **occupied and visible** and **explains the observed pixel value.**

- ▶ How can we obtain  $\mu(o_i)$  and  $\mu(a_k)$ ?

# Message Passing

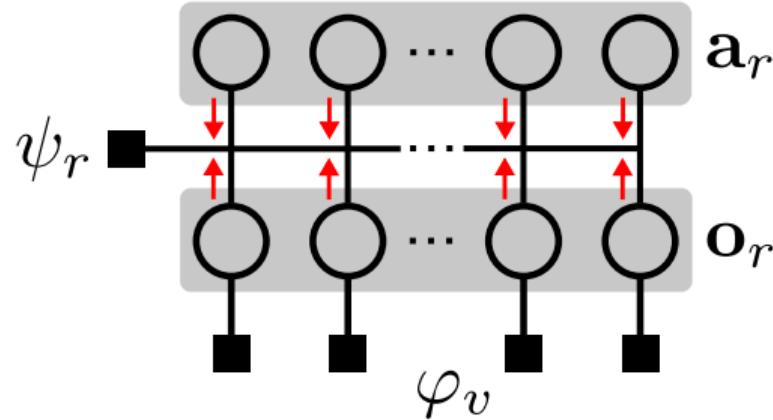


# Message Passing



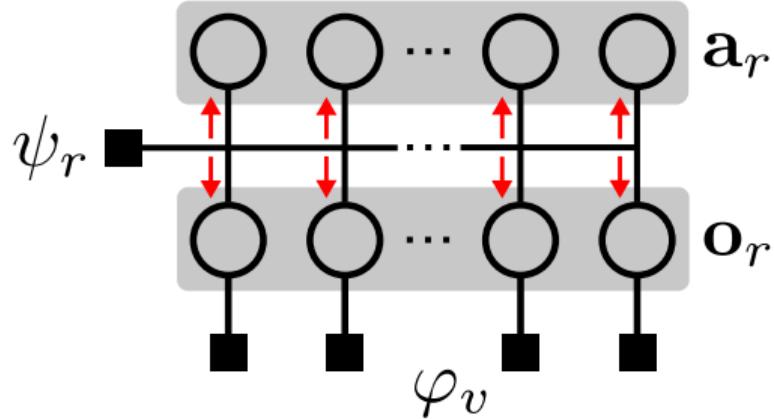
$$\mu_{\varphi_v \rightarrow o_v}(o_v) = \varphi_v(o_v)$$

# Message Passing



$$\mu_{x \rightarrow \psi_r}(x) = \prod_{f \in \mathcal{F} \setminus \psi_r} \mu_{f \rightarrow x}(x)$$

# Message Passing



$$\mu_{\psi_r \rightarrow x}(x) = ?$$

- ▶ Can be computed in **linear time** (see paper for technical derivation)

# Inference

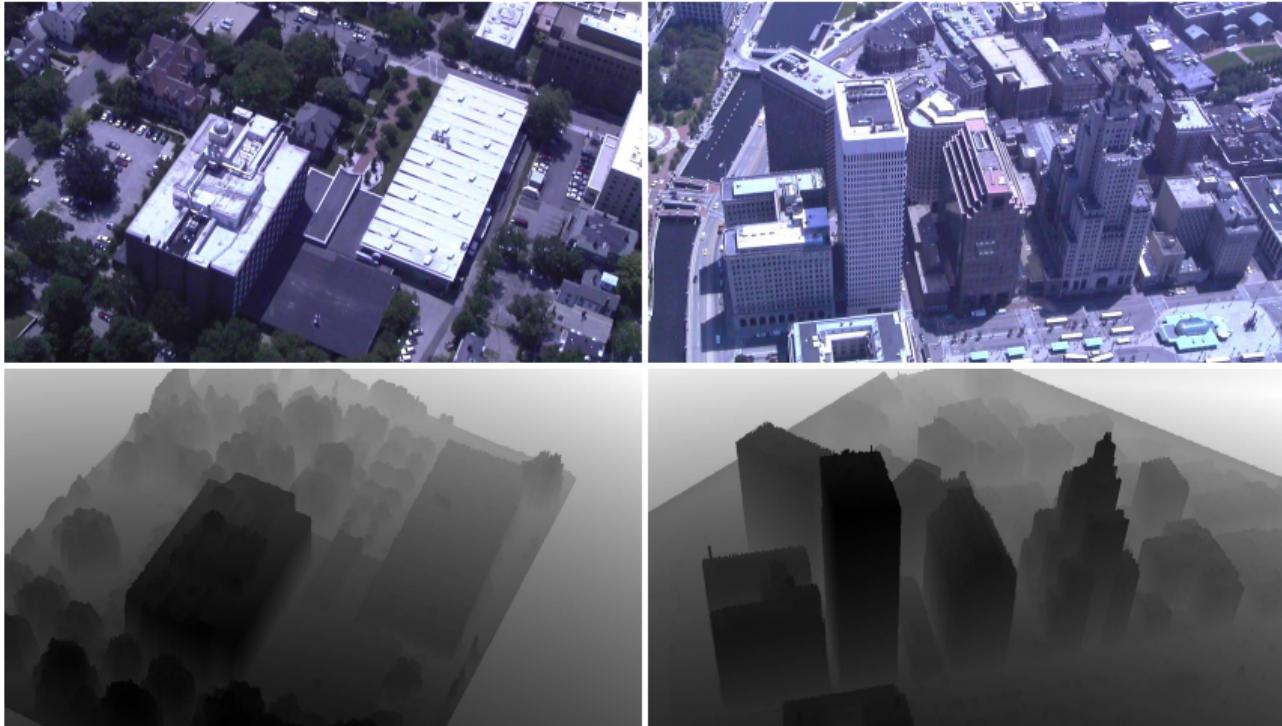
## Challenges:

1. MRF comprises **discrete and continuous variables**
2. Ray potentials are **high-order**
3. **Many factors:** each pixel defines a factor

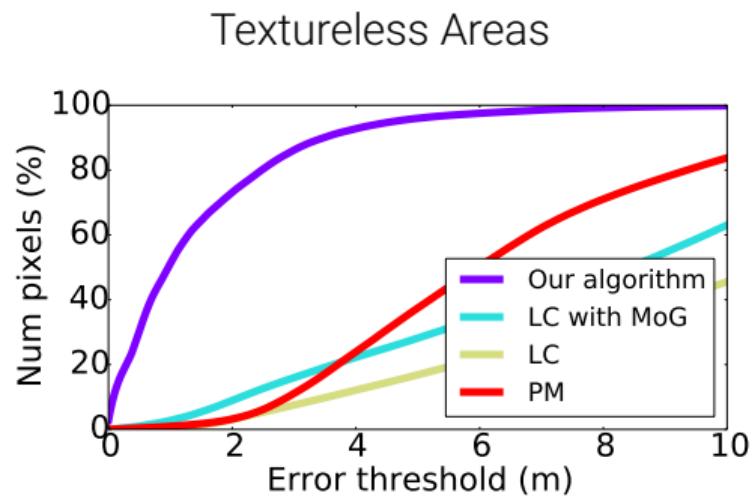
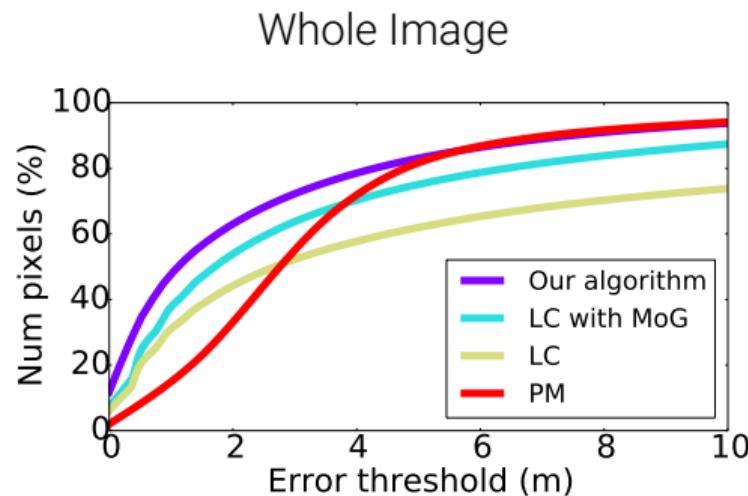
## Solution:

1. Approximate continuous belief propagation  
    ⇒ Update MoG's via importance sampling
2. Messages can be calculated in linear time  
    ⇒ Exact (but technical) derivation of messages
3. Octree implementation & GPGPU parallelization

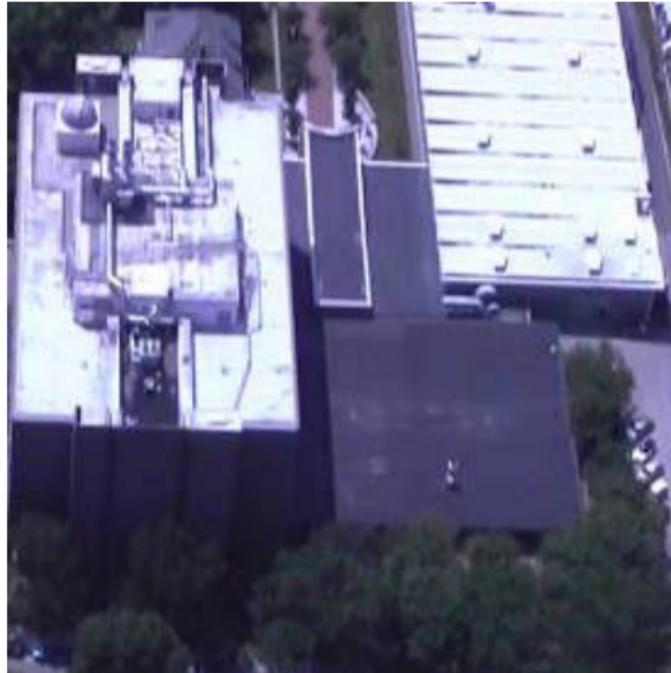
# Experimental Results



# Quantitative Results

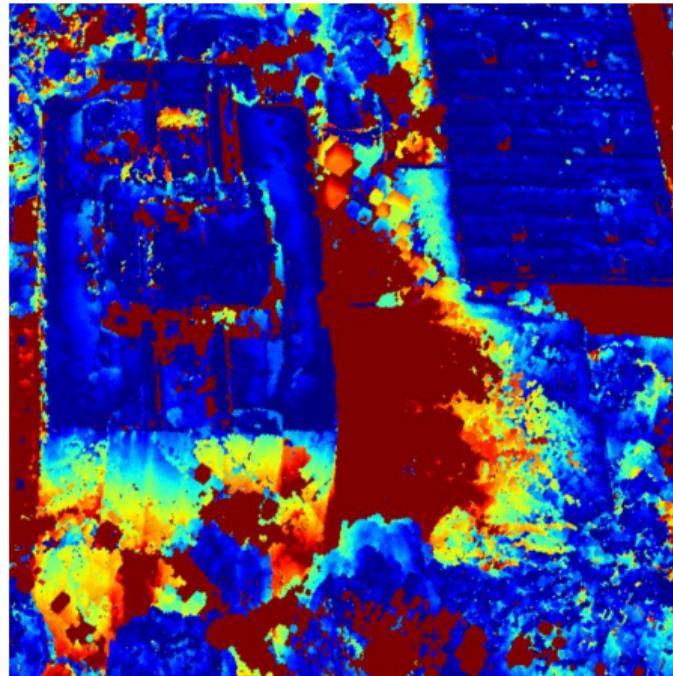


# Qualitative Results



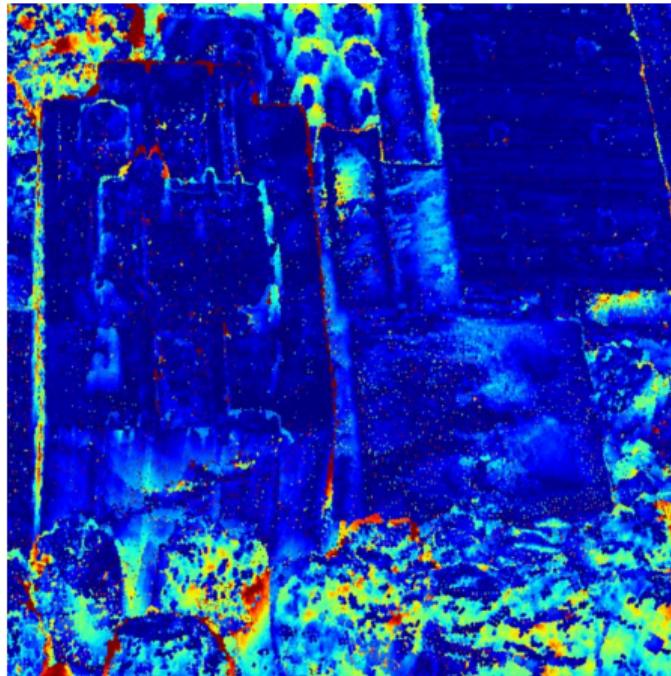
Input Image

# Qualitative Results



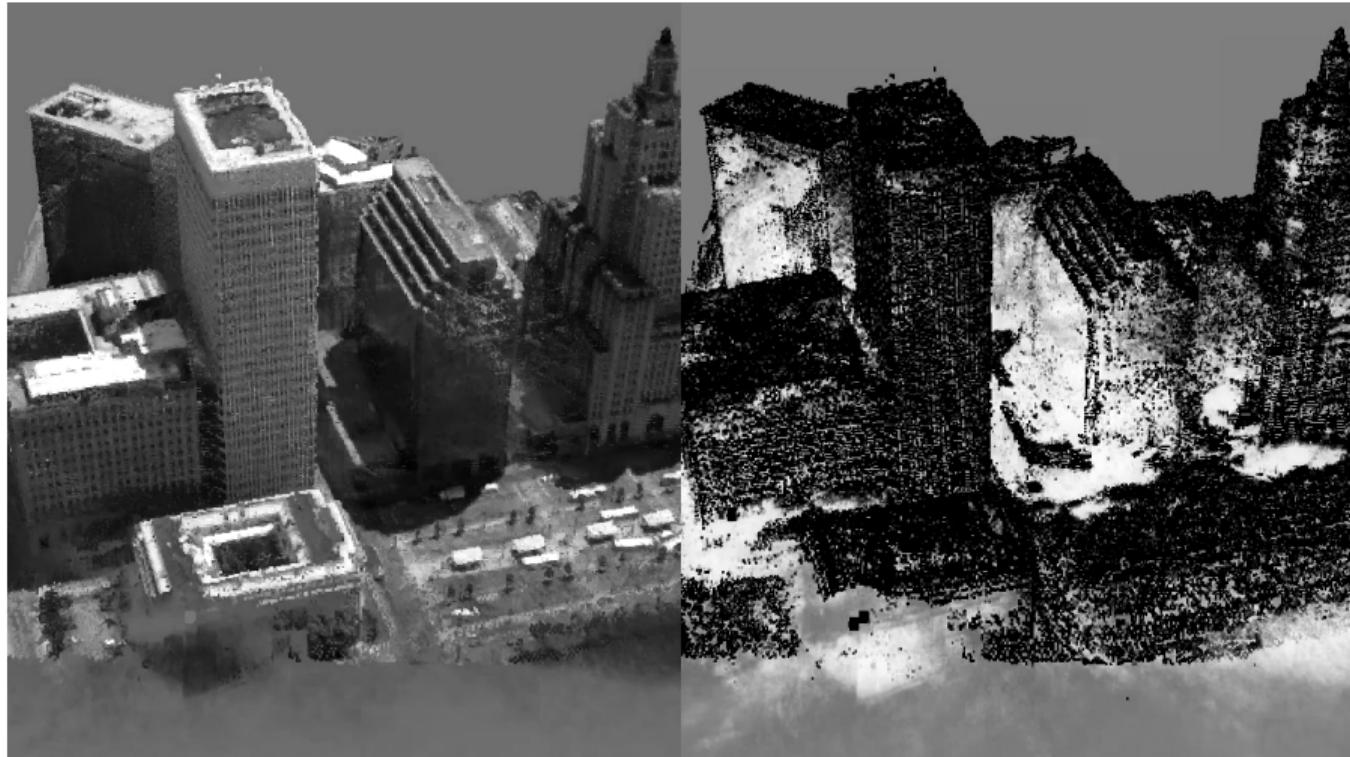
Error Map (Liu & Cooper, 2014)

# Qualitative Results



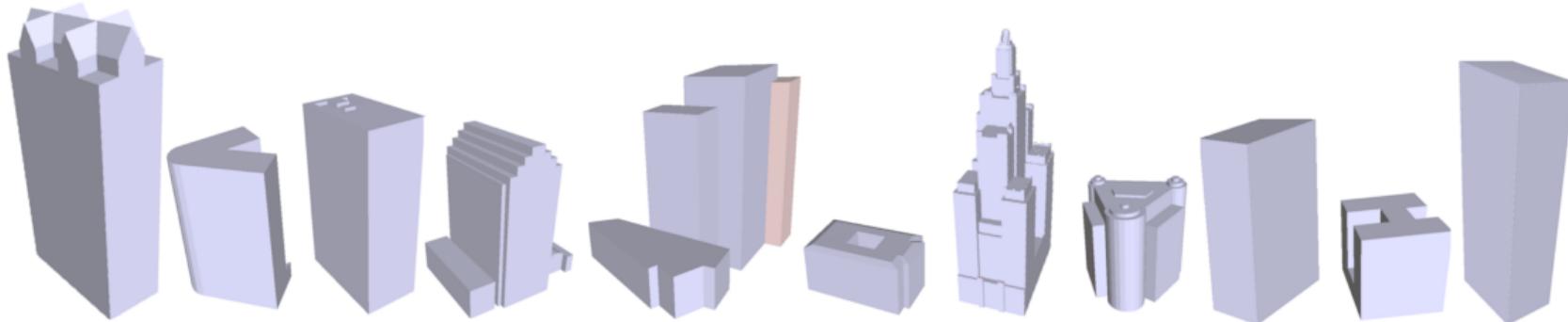
Error Map (Bayes Optimal)

# Qualitative Results: Our Results



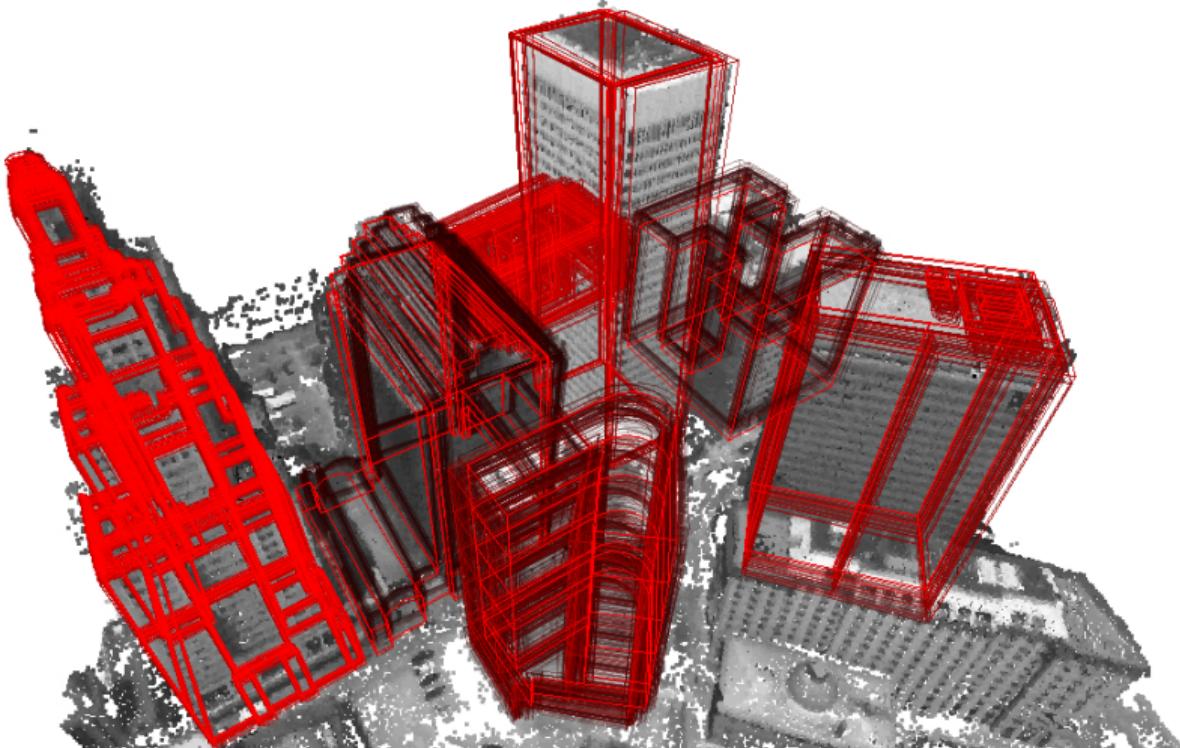
# Integrating 3D Shape Priors

# 3D Shape Priors



- ▶ For many scenes **exclusive prior knowledge** is available
  - ▶ GPS tags can be used to retrieve 3D Warehouse models
  - ▶ 3D models of IKEA furniture for indoor scenes
- ▶ Challenges:
  - ▶ Often only coarse and inaccurate models
  - ▶ Unknown orientation and approximate location
  - ▶ Occlusions and object size

# Probabilistic Model Fitting and 3D Reconstruction



# Probabilistic Multi-View Reconstruction Summary

## Pros:

- ▶ Probabilistic formulation is tractable as ray factors decompose
- ▶ Non-local constraints via joint inference in 2D and 3D
- ▶ CAD priors can help disambiguate textureless regions
- ▶ Using octrees, reconstructions up to  $1024^3$  voxels are possible

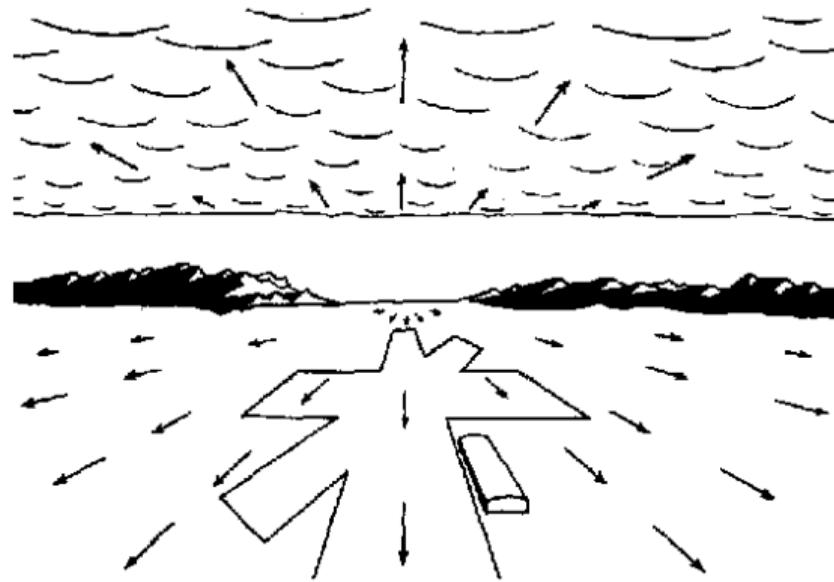
## Cons:

- ▶ Only approximate inference possible (highly loopy)
- ▶ Relatively slow: several minutes per scene on a GPU
- ▶ Appearance term very simplistic and not robust
- ▶ Resolution limited by discrete voxels (as opposed to meshes)

# 6.3

## Optical Flow

# Optical Flow

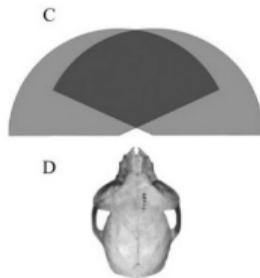


- ▶ Optical flow is the **apparent motion** of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and a scene.

# Stereo vs. Optical Flow

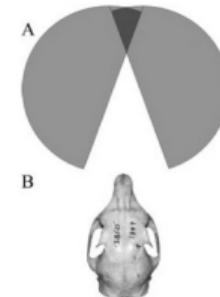
## Stereo

- ▶ 2 images at the same time
- ▶ Only camera motion
- ▶ 1D estimation problem
- ▶ Monkeys

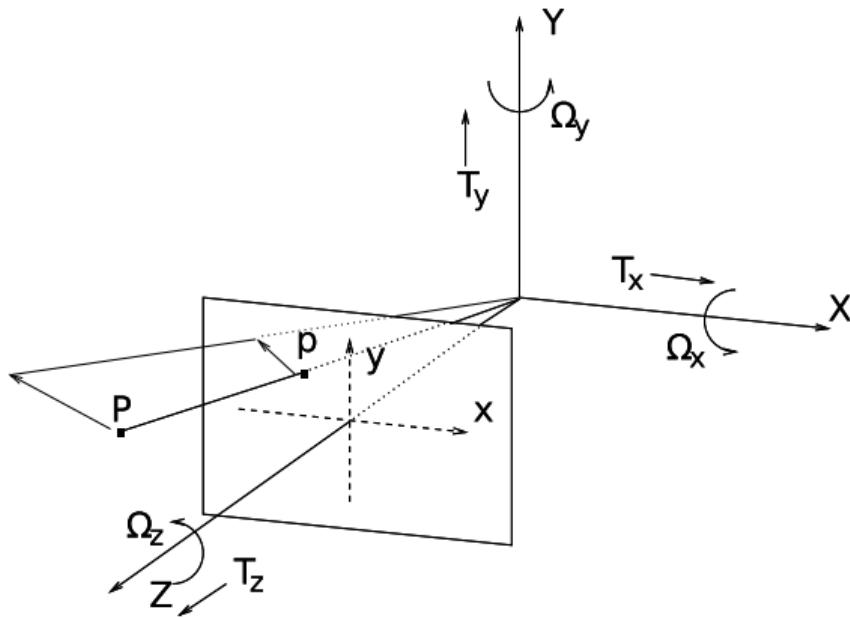


## Optical Flow

- ▶ 2 images at 2 time steps
- ▶ Camera and object motion
- ▶ 2D estimation problem
- ▶ Squirrels



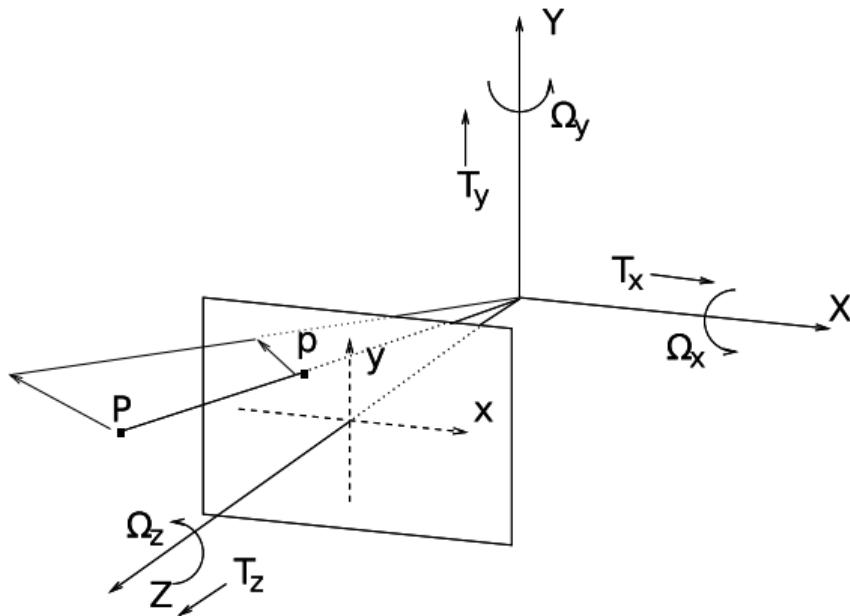
# Optical Flow



## Motion field:

- ▶ 2D motion field representing the **projection of the 3D motion** of points in the scene onto the image plane
- ▶ Can be the result of camera motion or object motion (or both)

# Optical Flow

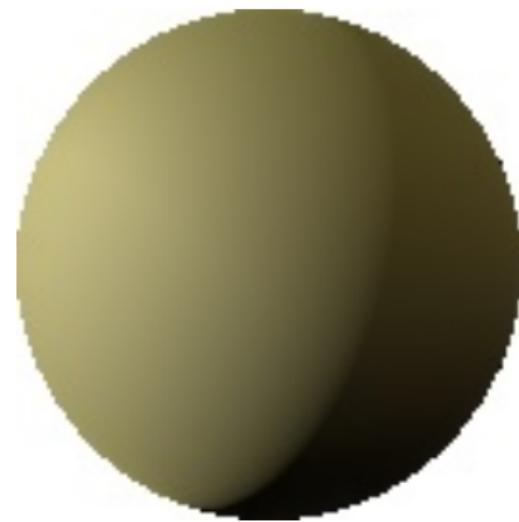


## Optical flow:

- ▶ 2D velocity field describing the **apparent motion** in the image  
(i.e., the displacement of pixels looking “similar”)
- ▶ Optical flow  $\neq$  motion field! Why?

# Thought Experiment

- ▶ Lambertian ball  
**rotating in 3D**
- ▶ What does the 2D  
**motion field** look like?
- ▶ What does the 2D  
**optical flow field** look like?

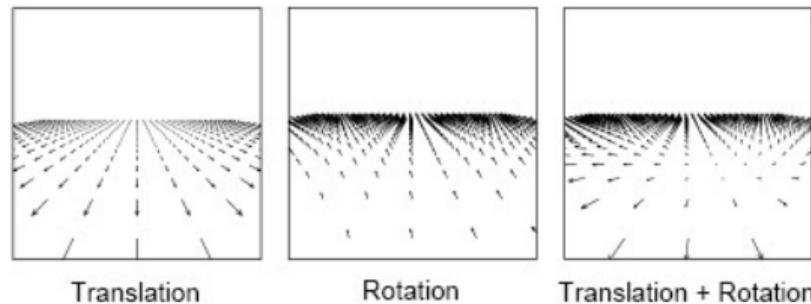


# Thought Experiment

- ▶ Stationary specular ball  
**moving light source**
- ▶ What does the 2D  
**motion field** look like?
- ▶ What does the 2D  
**optical flow field** look like?



# Optical Flow Field



The optical flow fields tell us something (maybe ambiguous) about:

- ▶ The **3D structure** of the world
- ▶ The **motion of objects** in the viewing area
- ▶ The **motion of the observer** (if any)

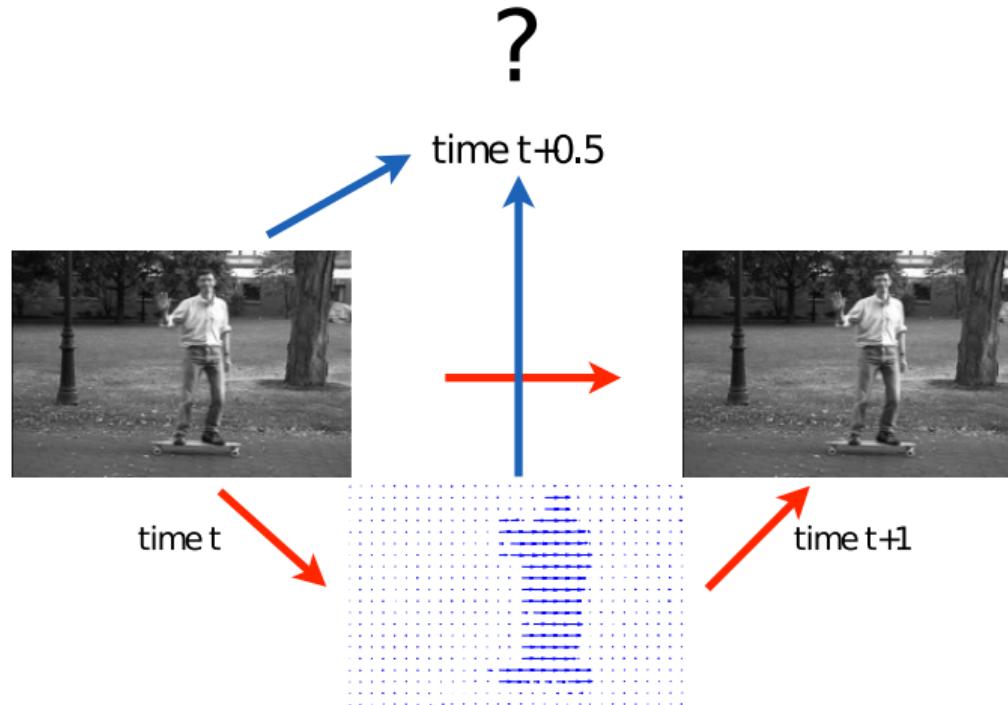
In contrast to stereo:

- ▶ No epipolar geometry  $\Rightarrow$  **2D estimation problem!**

# The Northern Gannet



# Applications: Video Interpolation / Frame Rate Adaption



- If we know the image motion we can compute images at intermediate frames

## Applications: Video Interpolation / Frame Rate Adaption



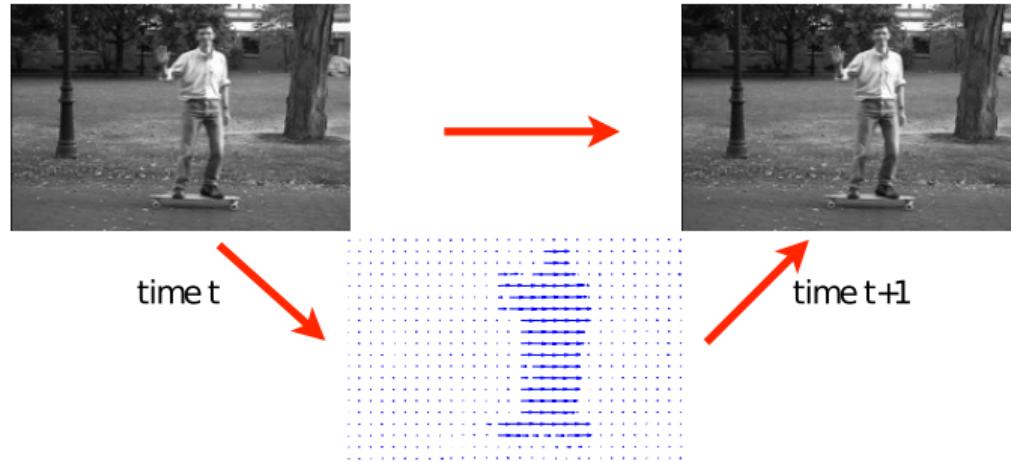
- If we know the image motion we can compute images at intermediate frames

## Applications: Video Interpolation / Frame Rate Adaption



- If we know the image motion we can compute images at intermediate frames

# Applications: Video Compression



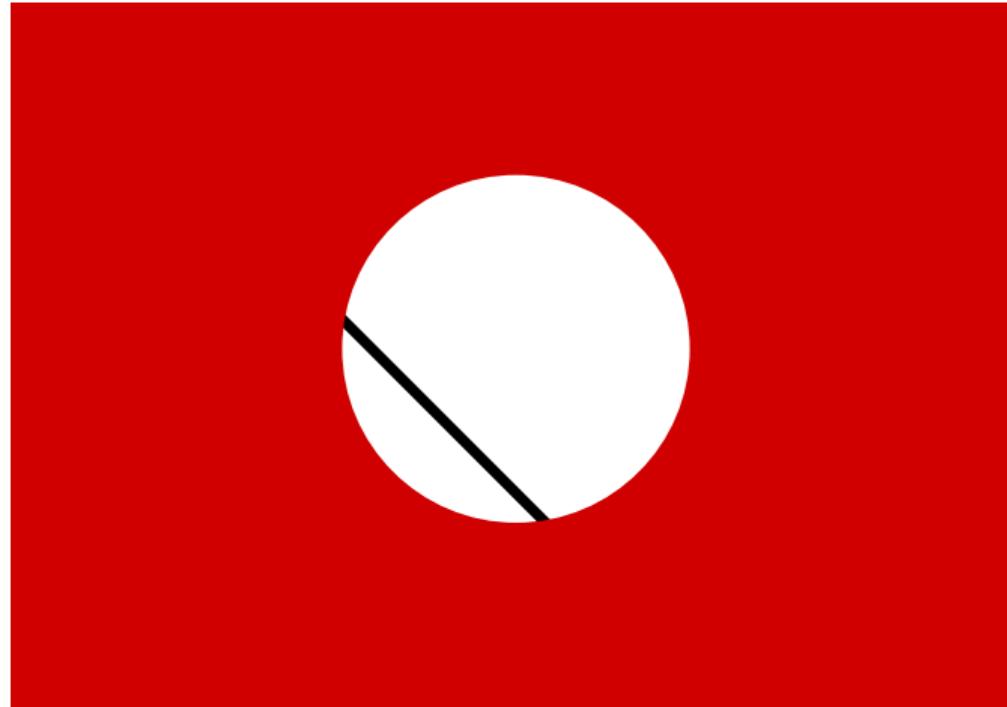
- ▶ To compress an image sequence, we can predict new frames using the optical flow field and only store how to “fix” the prediction
- ▶ Flow fields are smooth, thus easier to compress/store than images!

# Applications: Autonomous Driving



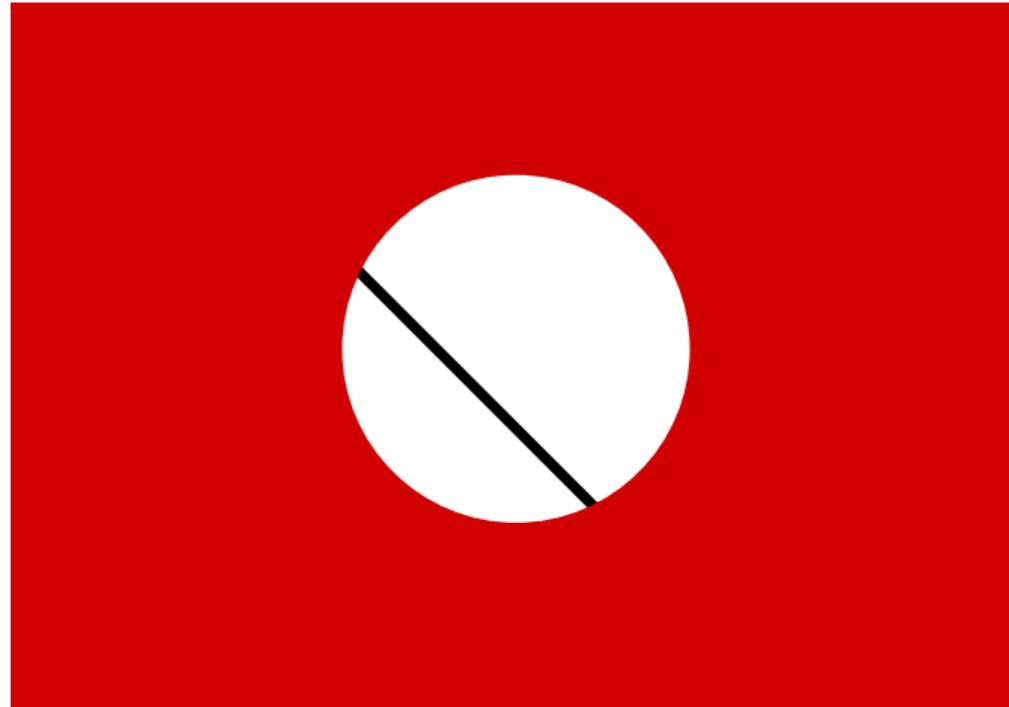
# Aperture Problem

In which direction does the line move?



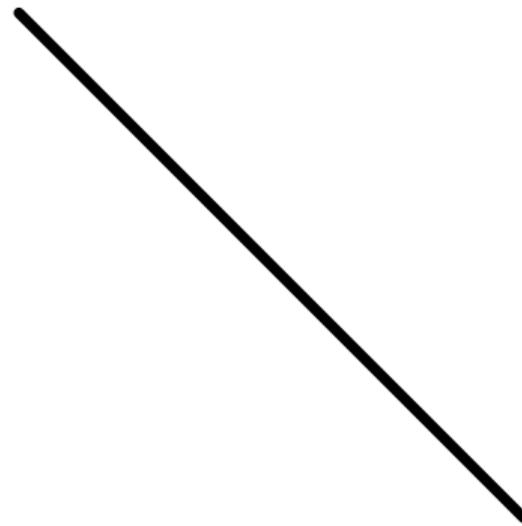
# Aperture Problem

In which direction does the line move?



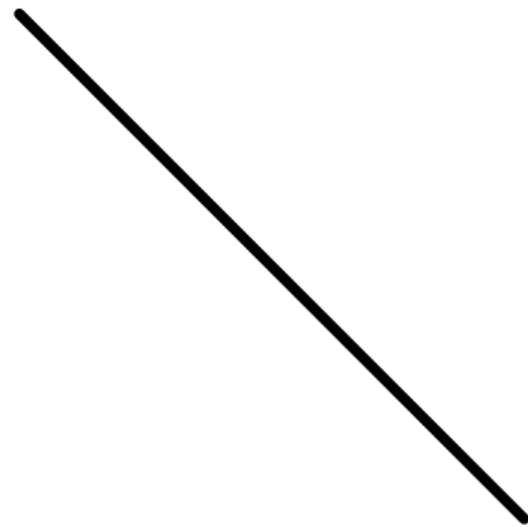
# Aperture Problem

Now the full picture ...



# Aperture Problem

Now the full picture ...



# Aperture Problem



- ▶ Aperture problem: A single observation is not enough to determine flow
- ▶ Barber Pole: What is the motion field? What is the optic flow field?

# Determining Optical Flow

# Horn-Schunck Optical Flow

- ▶ Consider the image  $I$  as a function of continuous variables  $x, y, t$
- ▶ Consider  $u(x, y)$  and  $v(x, y)$  as continuous flow fields (functions)
- ▶ Minimize the following **energy functional**

$$\begin{aligned} E(u, v) = & \iint \underbrace{(I(x + u(x, y), y + v(x, y), t + 1) - I(x, y, t))^2}_{\text{quadratic penalty for brightness change}} \\ & + \lambda \cdot \underbrace{\left( \|\nabla u(x, y)\|^2 + \|\nabla v(x, y)\|^2 \right)}_{\text{quadratic penalty for flow change}} dx dy \end{aligned}$$

with regularization parameter  $\lambda$  and gradient  $\nabla = \left( \frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right)$ .

# Horn-Schunck Optical Flow

$$\begin{aligned} E(u, v) &= \iint (I(x + u(x, y), y + v(x, y), t + 1) - I(x, y, t))^2 \\ &\quad + \lambda \cdot \left( \|\nabla u(x, y)\|^2 + \|\nabla v(x, y)\|^2 \right) dx dy \end{aligned}$$

- ▶ Minimizing this directly is a hard problem because the energy is highly **non-convex** and has many local optima
- ▶ Solution: **linearize** the brightness constancy assumption

# Horn-Schunck Optical Flow

First-Order Multivariable Taylor Series:

$$f(x, y) \stackrel{a,b}{\approx} f(a, b) + \frac{\partial f(a, b)}{\partial x}(x - a) + \frac{\partial f(a, b)}{\partial y}(y - b)$$

Therefore, we have:

$$\begin{aligned} & I(x + u(x, y), y + v(x, y), t + 1) \\ & \stackrel{x,y,t}{\approx} I(x, y, t) + I_x(x, y, t)(x + u(x, y) - x) \\ & \quad + I_y(x, y, t)(y + v(x, y) - y) + I_t(x, y, t)(t + 1 - t) \\ = & I(x, y, t) + I_x(x, y, t) u(x, y) + I_y(x, y, t) v(x, y) + I_t(x, y, t) \end{aligned}$$

# Horn-Schunck Optical Flow

Thus, the optical flow energy functional

$$\begin{aligned} E(u, v) &= \iint (I(x + u(x, y), y + v(x, y), t + 1) - I(x, y, t))^2 \\ &\quad + \lambda \cdot (\|\nabla u(x, y)\|^2 + \|\nabla v(x, y)\|^2) dx dy \end{aligned}$$

is approximated by the following **linearized equation:**

$$\begin{aligned} E(u, v) &= \iint (I_x(x, y, t)u(x, y) + I_y(x, y, t)v(x, y) + I_t(x, y, t))^2 \\ &\quad + \lambda \cdot (\|\nabla u(x, y)\|^2 + \|\nabla v(x, y)\|^2) dx dy \end{aligned}$$

# Horn-Schunck Optical Flow

Spatial discretization of the equation

$$\begin{aligned} E(u, v) = & \iint (I_x(x, y, t)u(x, y) + I_y(x, y, t)v(x, y) + I_t(x, y, t))^2 \\ & + \lambda \cdot (\|\nabla u(x, y)\|^2 + \|\nabla v(x, y)\|^2) dx dy \end{aligned}$$

leads to the following **discretized objective:**

$$\begin{aligned} E(\mathbf{U}, \mathbf{V}) = & \sum_{x,y} (I_x(x, y) u_{x,y} + I_y(x, y) v_{x,y} + I_t(x, y))^2 \\ & + \lambda \cdot ((u_{x,y} - u_{x+1,y})^2 + (u_{x,y} - u_{x,y+1})^2 + (v_{x,y} - v_{x+1,y})^2 + (v_{x,y} - v_{x,y+1})^2) \end{aligned}$$

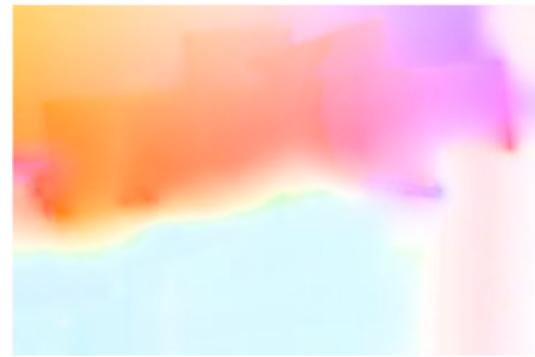
This objective is quadratic in the flow maps  $\mathbf{U}, \mathbf{V}$  and thus has a unique optimum.

# Horn-Schunck Optical Flow

$$\begin{aligned} E(\mathbf{U}, \mathbf{V}) = & \sum_{x,y} (I_x(x,y) u_{x,y} + I_y(x,y) v_{x,y} + I_t(x,y))^2 \\ & + \lambda \cdot ((u_{x,y} - u_{x+1,y})^2 + (u_{x,y} - u_{x,y+1})^2 + \\ & (v_{x,y} - v_{x+1,y})^2 + (v_{x,y} - v_{x,y+1})^2) \end{aligned}$$

- ▶ Differentiate wrt.  $\mathbf{U}$ ,  $\mathbf{V}$  and set the gradient to 0
- ▶ Results in a huge but **sparse linear system**
- ▶ Can be solved using **standard techniques** (e.g., Gauss-Seidel, SOR)
- ▶ However, linearization works only for **small motions**
- ▶ Solution: Iterative estimation & warping, coarse-to-fine estimation

# Results of Horn & Schunck



# Horn & Schunck Optical Flow

- ▶ The HS results are quite plausible already
- ▶ However, the flow is very smooth, i.e., to overcome ambiguities we need to set  $\lambda$  to a high value which **oversmooths** flow discontinuities. Why?
- ▶ We use a **quadratic penalty** for penalizing changes in the flow
- ▶ This does not allow for **discontinuities** in the flow field
- ▶ In other words, it penalizes large changes too much and causes oversmoothing

# Robust Estimation of Optical Flow

# Probabilistic Interpretation

The HS optimization problem can be interpreted as **MAP inference in a MRF**:

$$p(\mathbf{U}, \mathbf{V}) = \frac{1}{Z} \exp \{-E(\mathbf{U}, \mathbf{V})\}$$

with the following **Gibbs energy**:

$$\begin{aligned} E(\mathbf{U}, \mathbf{V}) &= \sum_{x,y} (I_x(x, y) u_{x,y} + I_y(x, y) v_{x,y} + I_t(x, y))^2 \\ &+ \lambda \cdot ((u_{x,y} - u_{x+1,y})^2 + (u_{x,y} - u_{x,y+1})^2 + \\ &\quad (v_{x,y} - v_{x+1,y})^2 + (v_{x,y} - v_{x,y+1})^2) \end{aligned}$$

Remark: As  $\mathbf{U}, \mathbf{V}$  are continuous, we solve inference with gradient descent (not BP)

# Probabilistic Interpretation

The HS optimization problem can be interpreted as **MAP inference in a MRF**:

$$p(\mathbf{U}, \mathbf{V}) = \frac{1}{Z} \exp \{-E(\mathbf{U}, \mathbf{V})\}$$

The corresponding **Gibbs distribution** is Gaussian:

$$\begin{aligned} p(\mathbf{U}, \mathbf{V}) &\propto \prod_{x,y} \exp \left\{ - (I_x(x, y) u_{x,y} + I_y(x, y) v_{x,y} + I_t(x, y))^2 \right\} \\ &\times \exp \left\{ -\lambda (u_{x,y} - u_{x+1,y})^2 \right\} \times \exp \left\{ -\lambda (u_{x,y} - u_{x,y+1})^2 \right\} \\ &\times \exp \left\{ -\lambda (v_{x,y} - v_{x+1,y})^2 \right\} \times \exp \left\{ -\lambda (v_{x,y} - v_{x,y+1})^2 \right\} \end{aligned}$$

Remark: As  $\mathbf{U}, \mathbf{V}$  are continuous, we solve inference with gradient descent (not BP)

# Robust Regularization

**Quadratic penalties translate to Gaussian distributions:**

$$\begin{aligned} p(\mathbf{U}, \mathbf{V}) &\propto \prod_{x,y} \exp \left\{ - (I_x(x,y) u_{x,y} + I_y(x,y) v_{x,y} + I_t(x,y))^2 \right\} \\ &\times \exp \left\{ -\lambda (u_{x,y} - u_{x+1,y})^2 \right\} \times \exp \left\{ -\lambda (u_{x,y} - u_{x,y+1})^2 \right\} \\ &\times \exp \left\{ -\lambda (v_{x,y} - v_{x+1,y})^2 \right\} \times \exp \left\{ -\lambda (v_{x,y} - v_{x,y+1})^2 \right\} \end{aligned}$$

- ▶ Both assumptions are invalid (e.g., discontinuities at object boundaries). Why?
- ▶ Gaussian distributions correspond to squared loss functions
- ▶ Squared loss functions are not robust to outliers!
- ▶ Outliers occur at object boundaries (violation of smoothness/regularizer)
- ▶ Outliers occur at specular highlights (violation of photoconsistency/data term)

# Robust Regularization

Solution: Use **robust data term** and smoothness penalties  $\rho(\cdot)$ :

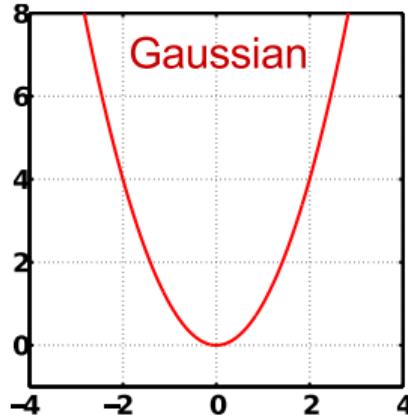
$$\begin{aligned} p(\mathbf{U}, \mathbf{V}) &\propto \prod_{x,y} \exp \left\{ -\rho_D (I_x(x,y) u_{x,y} + I_y(x,y) v_{x,y} + I_t(x,y)) \right\} \\ &\times \exp \left\{ -\lambda \rho_S(u_{x,y} - u_{x+1,y}) \right\} \times \exp \left\{ -\lambda \rho_S(u_{x,y} - u_{x,y+1}) \right\} \\ &\times \exp \left\{ -\lambda \rho_S(v_{x,y} - v_{x+1,y}) \right\} \times \exp \left\{ -\lambda \rho_S(v_{x,y} - v_{x,y+1}) \right\} \end{aligned}$$

- ▶ How to choose  $\rho_D(\cdot)$  and  $\rho_S(\cdot)$ ? We want a prior that allows for discontinuities in the optical flow and a likelihood that allows for outliers and occlusions
- ▶ Replace Gaussian with (heavy-tailed) **Student-t distribution** (=Lorentzian penalty):

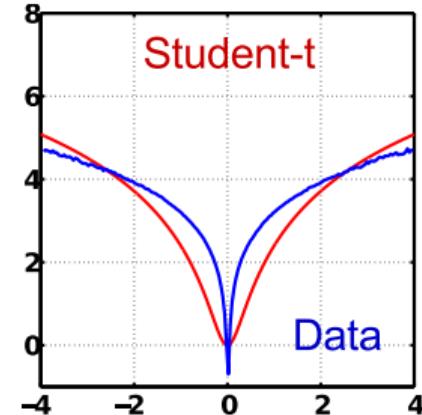
$$p(x) \propto \left( 1 + \frac{x^2}{2\sigma^2} \right)^{-\alpha} \Rightarrow \rho(x) = -\log(p(x)) = \alpha \log \left( 1 + \frac{x^2}{2\sigma^2} \right)$$

# Robust Regularization

Gaussian penalty (squared loss) vs. robust Student-t penalty:



negative  
log-density  
(i.e. energy)

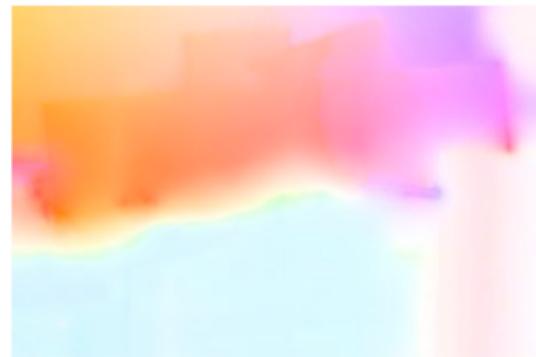


$$p(x) \propto \left(1 + \frac{x^2}{2\sigma^2}\right)^{-\alpha}$$

$$\rho(x) = \alpha \log \left(1 + \frac{x^2}{2\sigma^2}\right)$$

- Proposed by [Black-Anandan, ICCV 1993], [Sun et al., CVPR 2010] and others

# Result of Horn & Schunck

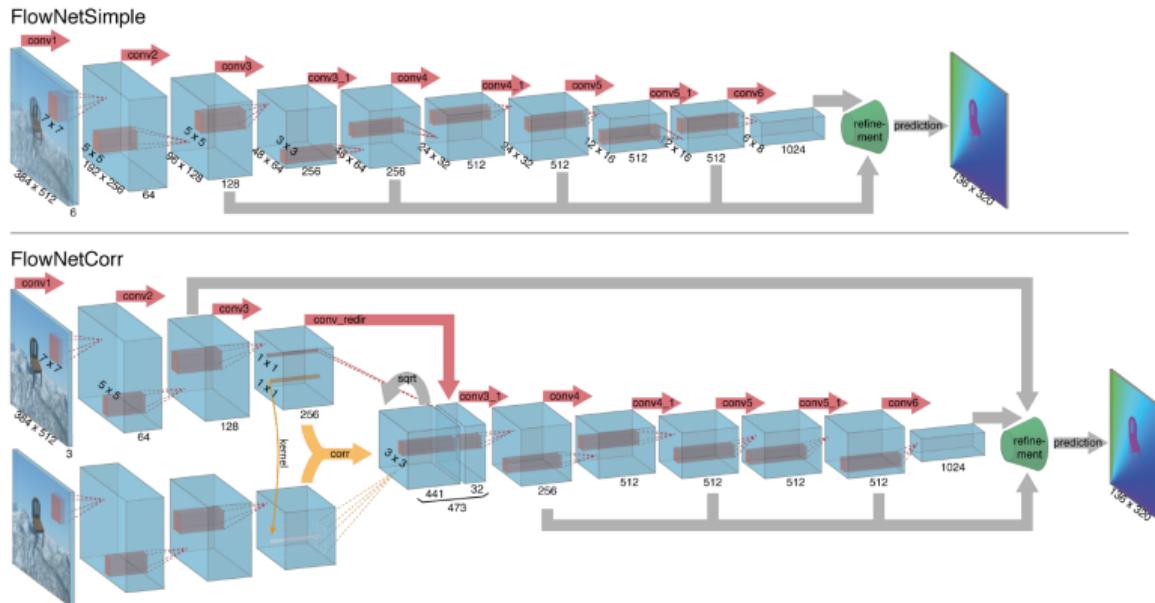


# Results of Sun / Black & Anandan



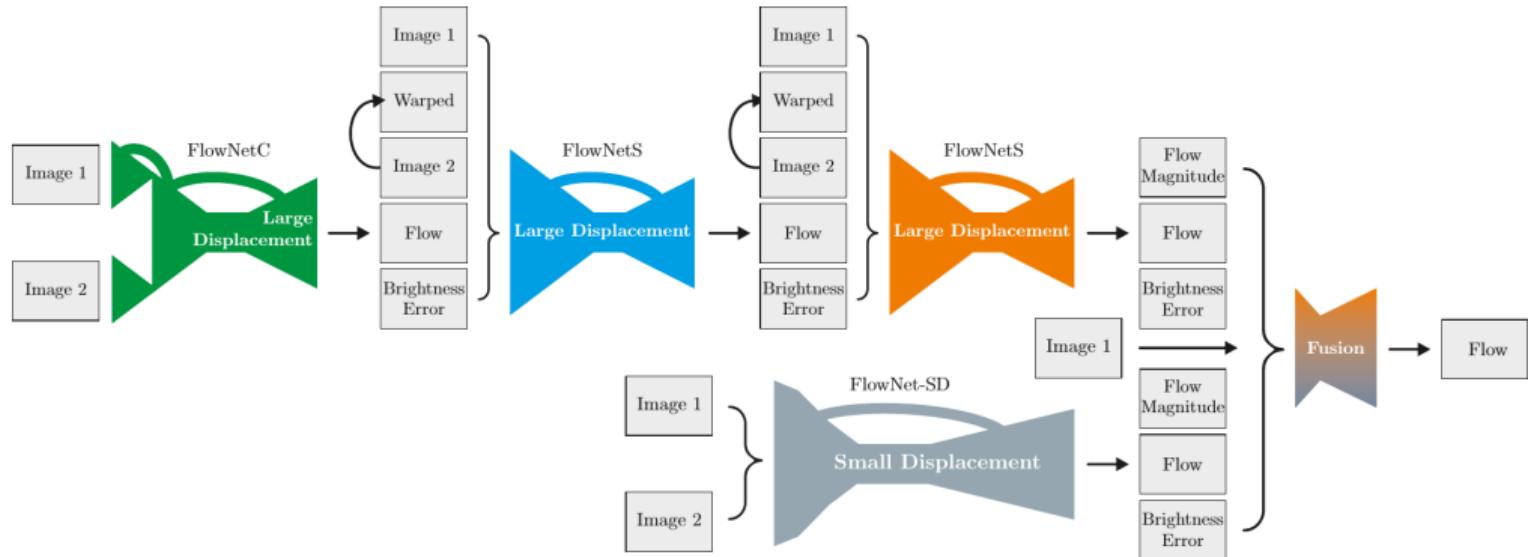
# End-to-End Deep Learning

# FlowNet



- ▶ Encoder: Convolution with stride, decoder: Upconvolution, skip-connections
- ▶ Multi-scale loss (EPE in pixels), curriculum learning, synth. training data

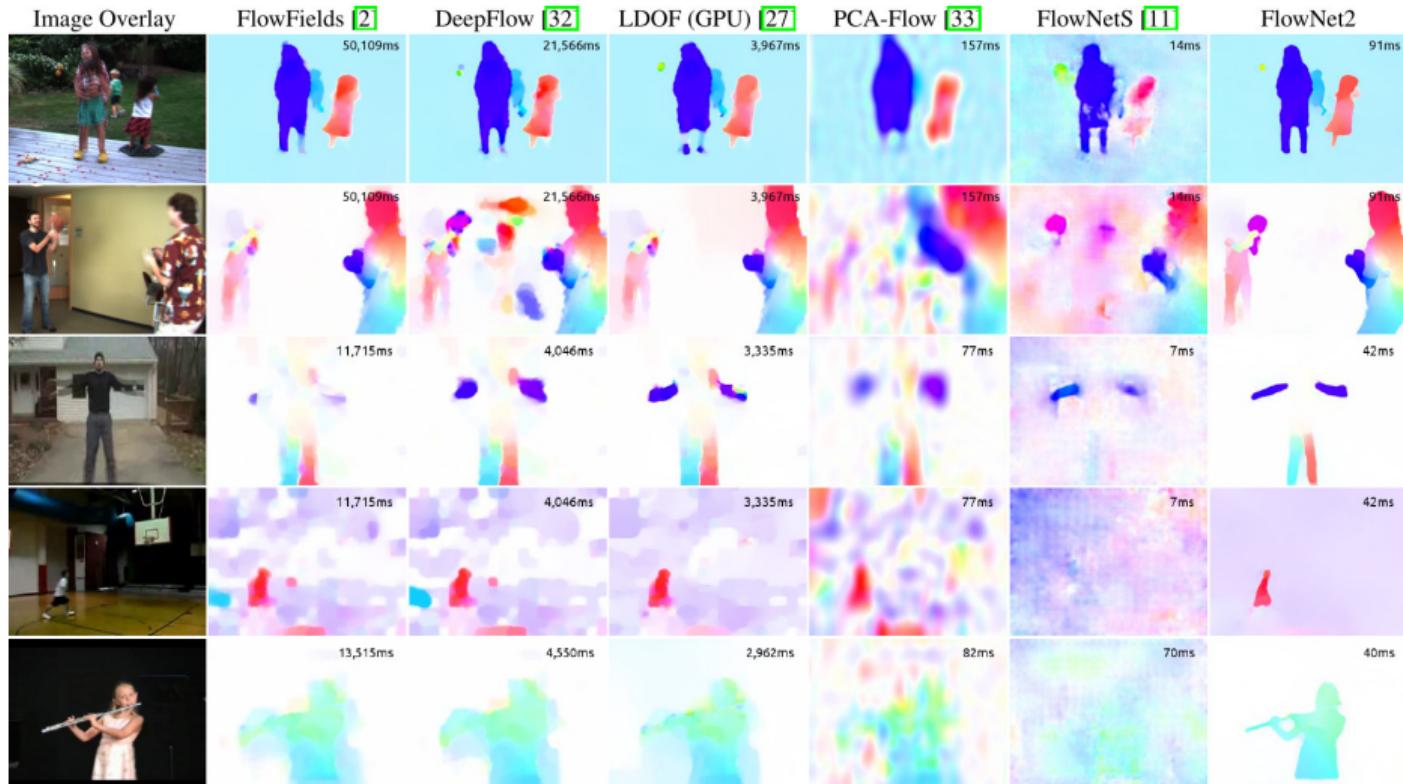
# FlowNet2



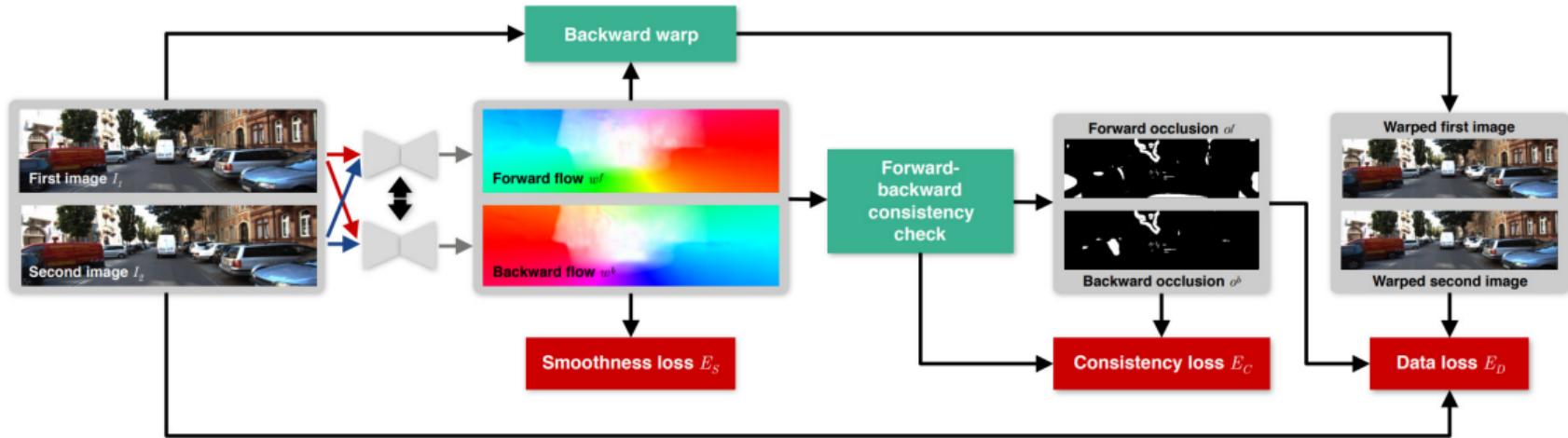
[Ilg et al., CVPR 2017]

- Combination (stacking) of multiple FlowNets with warping

# FlowNet2 – Results



# UnFlow: Unsupervised Optical Flow



- ▶ Learn optical flow without supervision by warping images into the other frame
- ▶ Uses photometric/smoothness terms as loss functions for training the neural net
- ▶ More about self-supervised learning in Lecture 12

# Optical Flow Summary

- ▶ Classical OF approaches state-of-the-art until 2016
- ▶ DL based methods on par or better since 2017
- ▶ But require
  - ▶ Big models
  - ▶ Enormous amount of (synthetic) training data
  - ▶ GPU compute time
  - ▶ Sophisticated curriculum learning schedules
- ▶ Top performing DL methods borrow many elements from classical methods  
(e.g., warping, cost volume, coarse-to-fine estimation, loss functions)