Graphical Models in Computer Vision

Andreas Geiger

Max Planck Institute for Intelligent Systems
Perceiving Systems

July 4, 2016

Syllabus

11.04.2016	Introduction
18.04.2016	Graphical Models 1
25.04.2016	Graphical Models 2 (Sand 6/7)
02.05.2016	Graphical Models 3
09.05.2016	Graphical Models 4
23.05.2016	Body Models 1
30.05.2016	Body Models 2
06.06.2016	Body Models 3
13.06.2016	Body Models 4
20.06.2016	Object Detection 1
27.06.2016	Object Detection 2
04.07.2016	Stereo
11.07.2016	Optical Flow
18.07.2016	Segmentation

Todays topic

What have we learned so far?

- Graphical Models
  - Belief Networks
  - Markov Networks
  - Factor Graphs
- Inference (Marginals, MAP)
  - Belief Propagation
  - Sampling
- Parameter Learning
  - Maximum-Likelihood
  - Max-Margin Methods
- Running Example
  - Human Pose Estimation

What's up for today?

Stereo

Slide credits: Robert Collins, Rob Fergus, Stefan Roth, Antonio Torralba

What is Vision?

"Vision is the act of knowing what is where by looking"
– Aristotle

Special emphasis:

- Relationship between 3D world and a 2D image
- Location and identity of objects.

Why is Computer Vision Hard?

- Knowing the geometry, material and lighting conditions it is well-understood how to generate the value at each pixel (computer graphics)
- However, this confluence of factors contributing to each pixel can not be easily decomposed. The process can not be inverted!

Experiment

- I am going to show you several images from the same scene/viewpoint
- I am going to change the light conditions
- Tell me what you see!

Why is vision hard?

- Congratulations! You just did something mathematically impossible!
- How? You used assumptions based on prior knowledge / experience about the way the world works.

Why is vision hard?

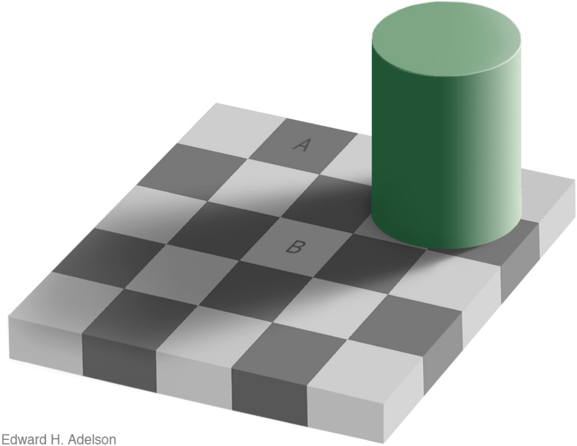
Let's make it a bit harder ...

Why is vision hard?

What about this one?

Why is vision hard?

Which square is brighter? A or B?



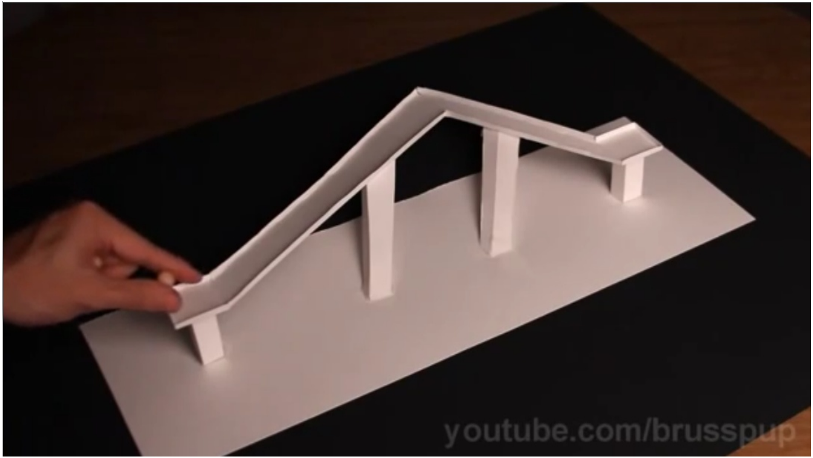
Edward H. Adelson

Why is vision hard?

Do you know this guy?

Why is vision hard?

Here is another one ...



Why is vision hard?

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Artificial Intelligence Group July 7, 1966

Vision Memo. No. 100.

THE SUMMER VISION PROJECT

Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

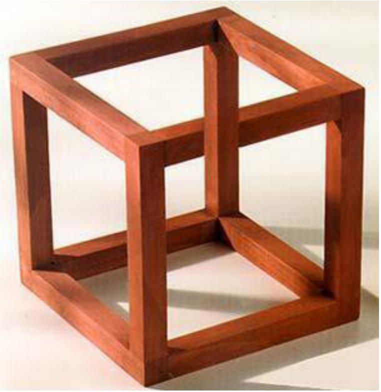
How to recover 3D from an image?

Which cues do we as humans have available
in order to recover depth from images of the 3D world?

How to recover 3D from an image?

In general, this is an ill-posed problem, but there are several cues:

- Occlusion
- Parallax
- Perspective
- Accomodation
- Stereopsis



How to recover 3D from an image?

In general, this is an ill-posed problem, but there are several cues:

- Occlusion
- Parallax
- Perspective
- Accomodation
- Stereopsis

How to recover 3D from an image?

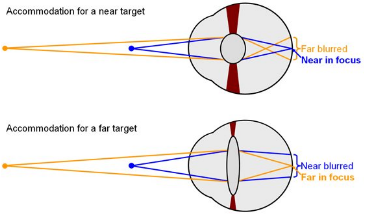
In general, this is an ill-posed problem, but there are several cues:

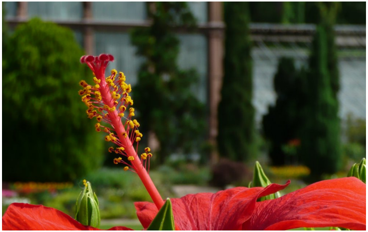
- Occlusion
- Parallax
- Perspective
- Accomodation
- Stereopsis

How to recover 3D from an image?

In general, this is an ill-posed problem, but there are several cues:

- Occlusion
- Parallax
- Perspective
- Accomodation
- Stereopsis

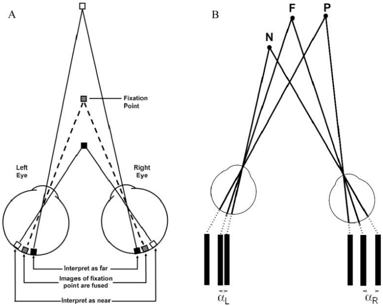




How to recover 3D from an image?

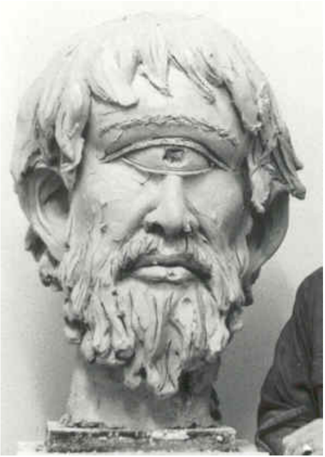
In general, this is an ill-posed problem, but there are several cues:

- Occlusion
- Parallax
- Perspective
- Accomodation
- Stereopsis

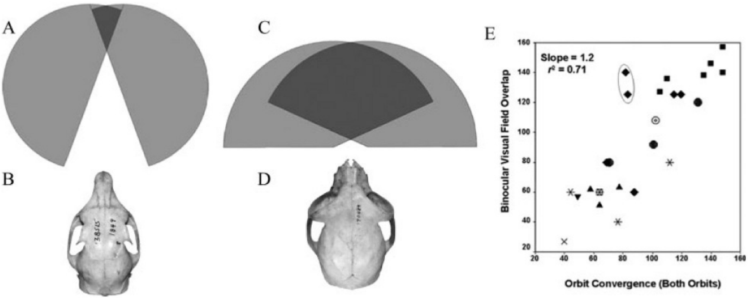


Our topic for today!

Why Binocular Stereopsis?

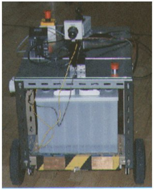


Why Binocular Stereopsis?



Computational Stereo

Computational Stereo

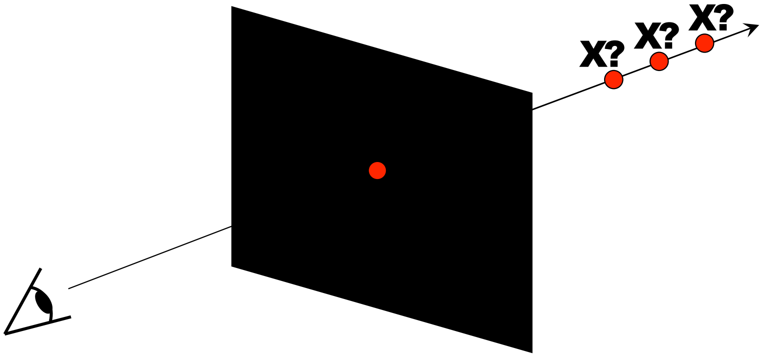


Multi-View Reconstruction

[Building Rome in a Day – Agarwal et al. 2011]

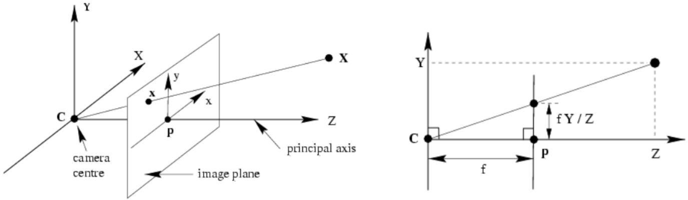
Projective Geometry

- Recovery of structure from one image is ambiguous



Pinhole Camera Model

- Assumption: Undistorted images (no lens distortion)
- 2D projection: Intersection of viewing ray with the image plane



- X, Y, Z: 3D coordinates
- x, y: 2D image coordinates
- f: focal length
- 3D to 2D projection π:

$$\pi : \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \mapsto \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f\,X/Z \\ f\,Y/Z \end{pmatrix}$$

Pinhole Camera Model

- Is this projection a linear mapping?
- No (division by Z)! Can we somehow "make it" linear?
- Yes: **Homogeneous coordinates** (add one more coordinate)
- Conversion to homogeneous coordinates:

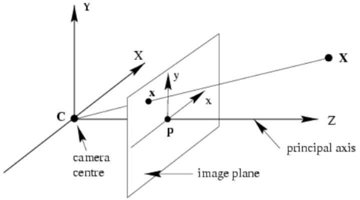
$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \Rightarrow \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \qquad \begin{pmatrix} x \\ y \end{pmatrix} \Rightarrow \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

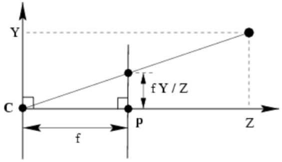
- Conversion *from* homogeneous coordinates:

$$\begin{pmatrix} X \\ Y \\ Z \\ W \end{pmatrix} \Rightarrow \begin{pmatrix} X/W \\ Y/W \\ Z/W \end{pmatrix} \qquad \begin{pmatrix} x \\ y \\ w \end{pmatrix} \Rightarrow \begin{pmatrix} x/w \\ y/w \end{pmatrix}$$

- Which homogeneous coordinates are equivalent?

Pinhole Camera Model



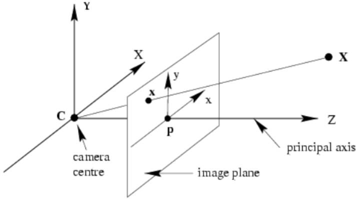


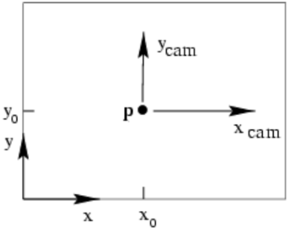
$$\pi : \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \mapsto \begin{pmatrix} f\,X/Z \\ f\,Y/Z \end{pmatrix} \quad \Rightarrow \quad \pi_H : \mathbf{X} = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} f\,X \\ f\,Y \\ Z \end{pmatrix} = \mathbf{x}$$

- π_H can be written as a linear function of the projection matrix $\mathbf{P}$:

$$\underbrace{\begin{pmatrix} f\,X \\ f\,Y \\ Z \end{pmatrix}}_{\mathbf{x}\in\mathbb{R}^3} = \underbrace{\begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\mathbf{P}\in\mathbb{R}^{3\times 4}} \cdot \underbrace{\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}}_{\mathbf{X}\in\mathbb{R}^4}$$

Pinhole Camera Model

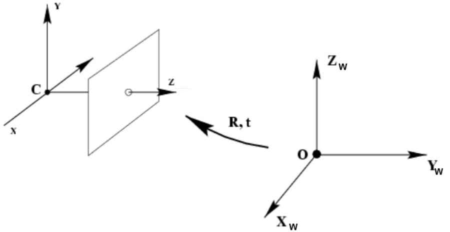




- Principal point $\mathbf{p} = (c_x, c_y)^T$: Point where the principal axis intersects the image plane (origin of normalized coordinate system)

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \mapsto \begin{pmatrix} f\,X/Z + c_x \\ f\,Y/Z + c_y \end{pmatrix} \Rightarrow \underbrace{\begin{pmatrix} f\,X + c_x Z \\ f\,Y + c_y Z \\ Z \end{pmatrix}}_{\mathbf{x}\in\mathbb{R}^3} = \underbrace{\begin{pmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\mathbf{P}\in\mathbb{R}^{3\times 4}} \cdot \underbrace{\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}}_{\mathbf{X}\in\mathbb{R}^4}$$

Pinhole Camera Model

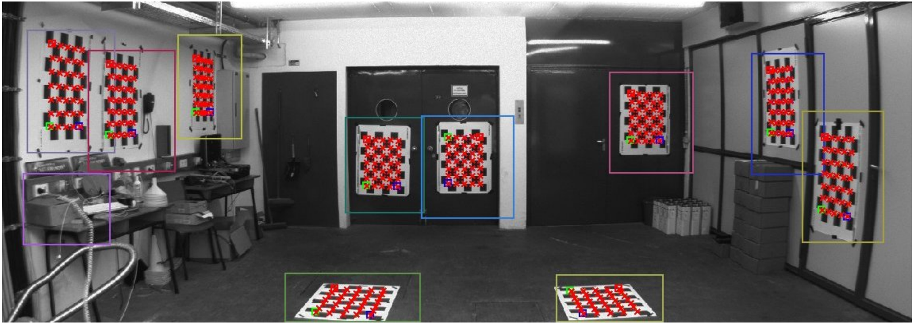


- Considering also the pose of the camera wrt. world coordinates:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} \underbrace{\mathbf{R}}_{\in\mathbb{R}^{3\times3}} & \big| & \underbrace{\mathbf{t}}_{\in\mathbb{R}^{3\times1}} \\ 0\;0\;0 & & 1 \end{pmatrix} \cdot \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} f\,X + c_x Z \\ f\,Y + c_y Z \\ Z \end{pmatrix} = \underbrace{\begin{pmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \underbrace{\mathbf{R}}_{\in\mathbb{R}^{3\times3}} & \big| & \underbrace{\mathbf{t}}_{\in\mathbb{R}^{3\times1}} \\ 0\;0\;0 & & 1 \end{pmatrix}}_{\mathbf{P}'} \cdot \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$

Camera Calibration



How to obtain these parameters for a given camera?

- Closed form approximation of intrinsics (f, c_x, c_y)
- Non-linear optimization of intrinsics (f, c_x, c_y) and extrinsics $(\mathbf{R}, \mathbf{t})$
- Several algorithms available (Zhang, Bouget, OpenCV)
- Online toolbox: `www.cvlibs.net/software/calibration`

Stereo Reconstruction

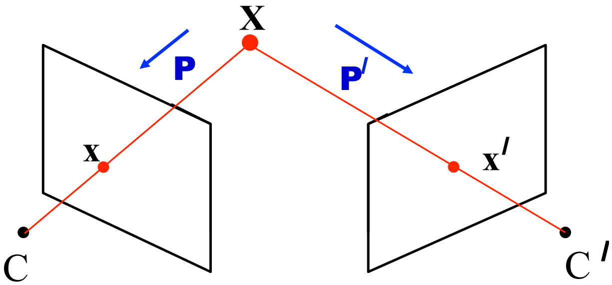
Task

- Construct a 3D model from 2 images of a calibrated camera

Pipeline:

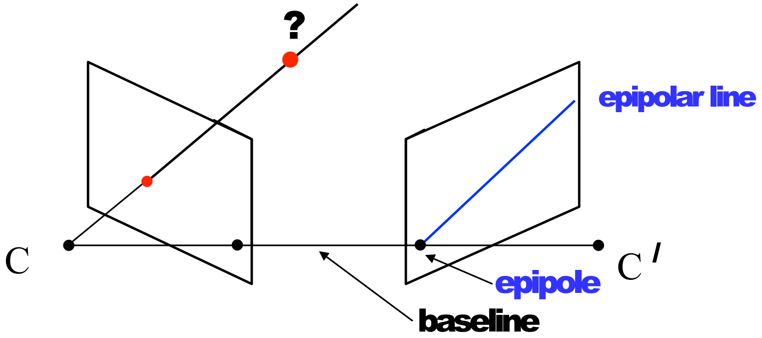
1. Find a set of corresponding points
2. Estimate the **epipolar geometry**
3. Rectify both images
4. Dense feature matching
5. 3D reconstruction

Epipolar Geometry



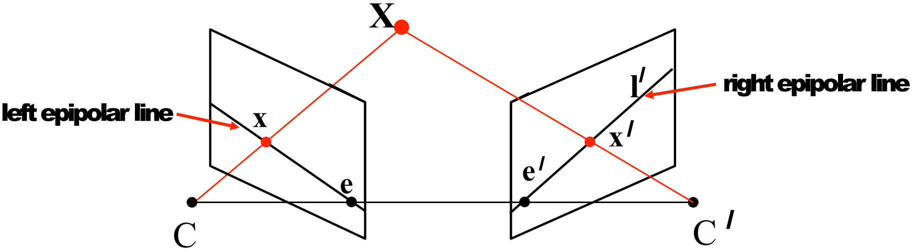
- $\mathbf{X} \in \mathbb{R}^4$: Homogeneous point in the 3D world
- $\mathbf{P}, \mathbf{P}' \in \mathbb{R}^{3\times4}$: Projection matrices ($\mathbf{x} = \mathbf{P}\mathbf{X}$, $\mathbf{x}' = \mathbf{P}'\mathbf{X}$)
- $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^3$: Homogeneous 2D pixel coordinates
- C, C': Camera center / focal point

Epipolar Geometry



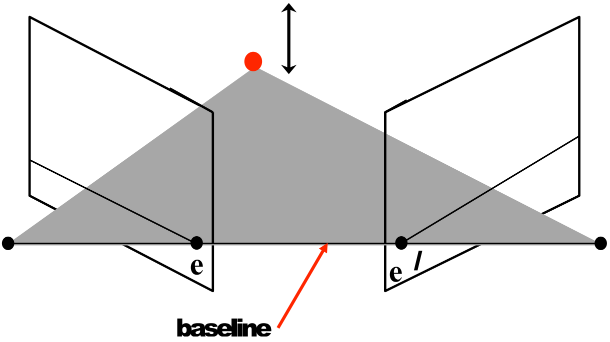
- Lets assume the camera parameters and geometry is known!
- Given a projection of a 3D point in the left image
- Where is it located in 3D?
- On the epipolar line defined by this point and the camera centers
- Reduces the search problem to 1D!

Epipolar Geometry



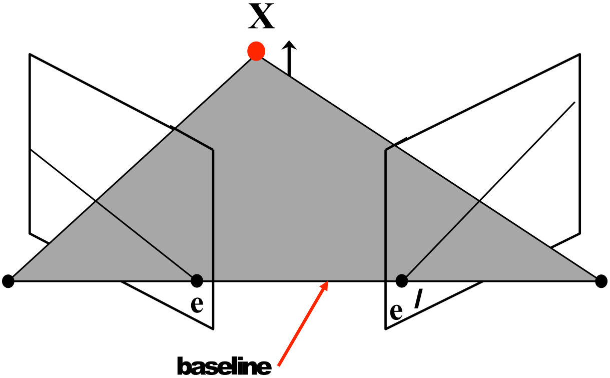
- $\overline{\mathbf{CC'}}$: Baseline (translation between cameras)
- $\mathbf{e}, \mathbf{e}'$: Epipole (intersection of image plane with baseline)
- $\mathbf{l}, \mathbf{l}'$: Epipolar line (intersection of image plane with epipolar plane)

Epipolar Geometry



- Each 3D point defines an epipolar plane (in combination with both camera centers). The set of planes is called “epipolar pencil”.
- All epipolar lines pass through the epipole

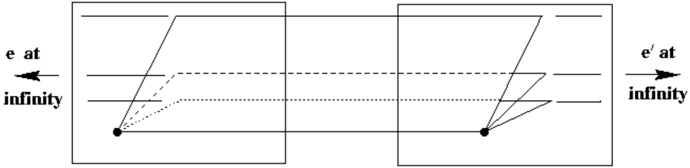
Epipolar Geometry

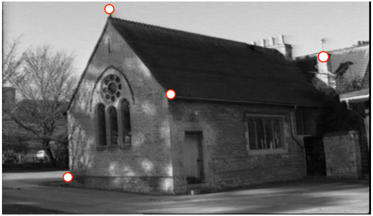


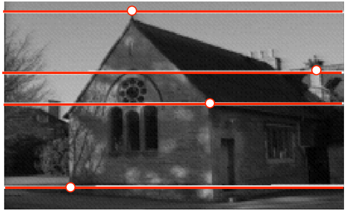
- Each 3D point defines an epipolar plane (in combination with both camera centers). The set of planes is called "epipolar pencil".
- All epipolar lines pass through the epipole

Epipolar Geometry

What if both cameras face the same direction?

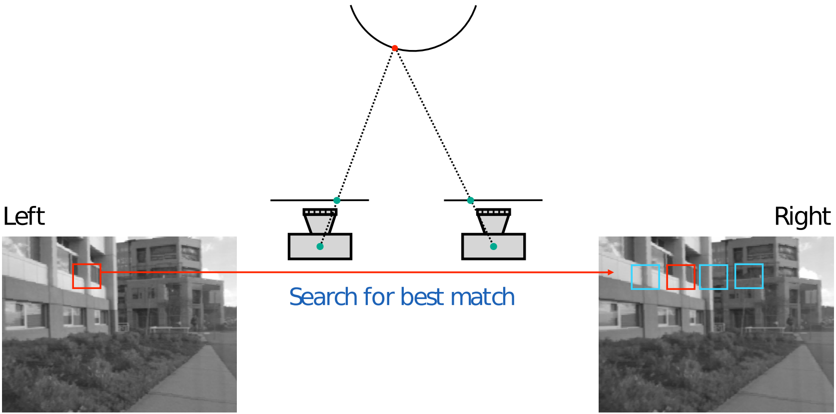






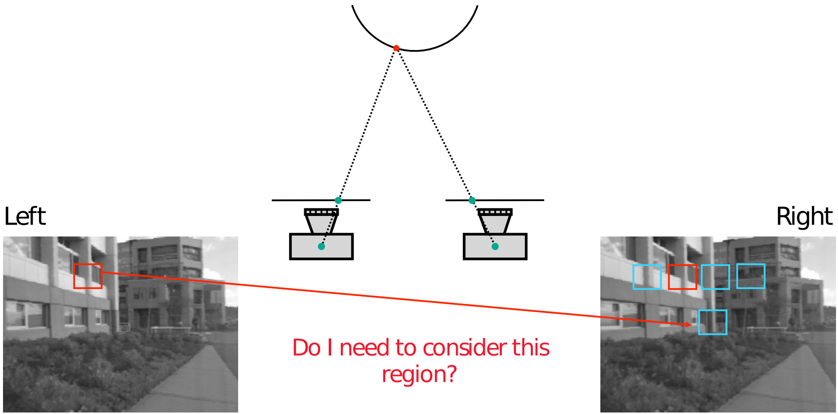
- Epipoles are at infinity, epipolar lines are parallel
- Correspondences along "scanlines" (simplifies computation)

Image Disparity



- The displacement between pixels is called "disparity": $d = x - x'$

Image Disparity



- The displacement between pixels is called "disparity": $d = x - x'$

Triangulation

How to recover a 3D point from two corresponding image points?

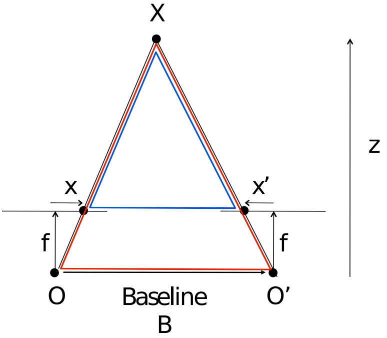
- Equal triangles (only when image planes are parallel)
- Using the definition $d = x - x'$:

$$\frac{Z - f}{B - (x - x')} = \frac{Z}{B}$$

$$ZB - fB = ZB - Z(x - x')$$

$$Z = \frac{fB}{x - x'} = \frac{fB}{d}$$

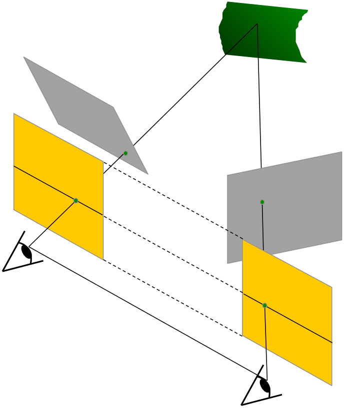
$$d = \frac{fB}{Z}$$



Rectification

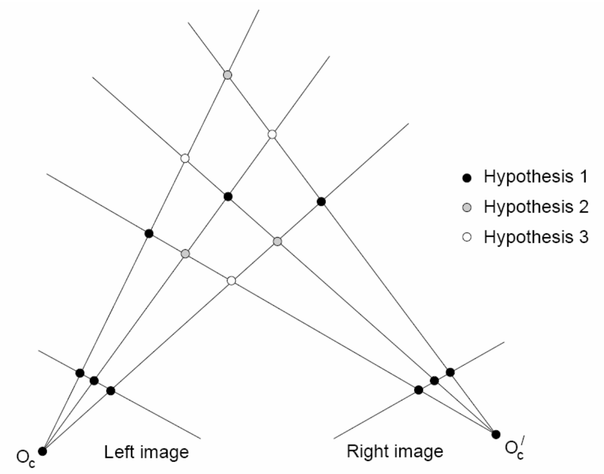
What if the images are not in the required setup?

- Rewarp them such that they are! ("Rectification")
- Map both image planes to a common plane parallel to the baseline using a "homography" (using homogeneous coordinates!)



Correspondences Unrectified vs. Rectified

Correspondence Ambiguity

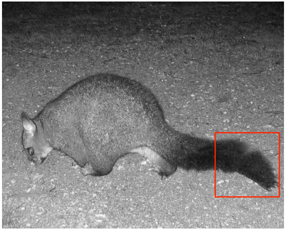


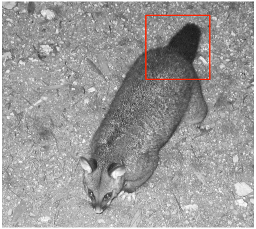
- Even when constrained to 1D many matching hypotheses exist
- Which one is correct?

Correspondence Ambiguity

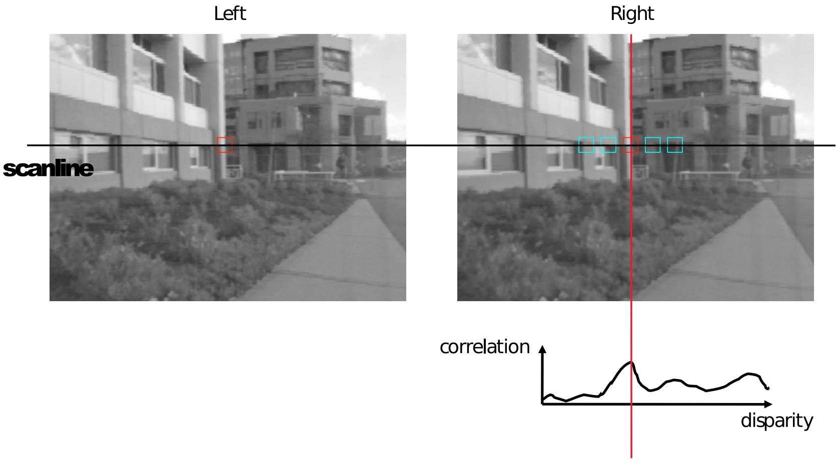
How to determine if two image points correspond?

- Assume that they look "similar"
- A single pixel does not reveal the local structure (ambiguities)
- Compare a small image region/patch!
- But even then the task is difficult:

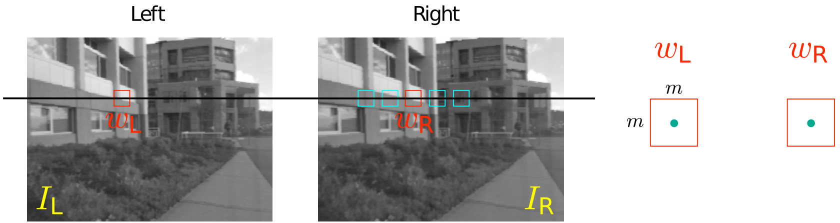




Normalized Correlation



Normalized Correlation



- w_L and w_R are corresponding $m \times m$ windows of pixels
- We can write them as vectors: $\mathbf{w}_L, \mathbf{w}_R \in \mathbb{R}^{m^2}$
- Normalized correlation (cosine of the enclosed angle):

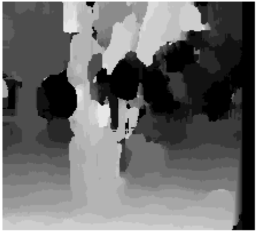
$$\text{NC}(x, y, d) = \frac{(\mathbf{w}_L(x, y) - \bar{\mathbf{w}}_L(x, y))^T (\mathbf{w}_R(x - d, y) - \bar{\mathbf{w}}_R(x - d, y))}{\|\mathbf{w}_L(x, y) - \bar{\mathbf{w}}_L(x, y)\|_2 \, \|\mathbf{w}_R(x - d, y) - \bar{\mathbf{w}}_R(x - d, y)\|_2}$$

Sum of squared differences (SSD):

$$\text{SSD}(x, y, d) = \|\mathbf{w}_L(x, y) - \mathbf{w}_R(x - d, y)\|_2^2$$

Block Matching: Window Size

$m = 3$



$m = 20$

Block Matching:

- Choose some disparity range $[0, d_{max}]$
- For all pixels $\mathbf{x} = (x, y)$ try all disparities and choose the one that maximizes the normalized correlation or minimizes the SSD
- This strategy is called: Winner-takes-all (WTA)
- Do this for both images, apply left-right consistency check

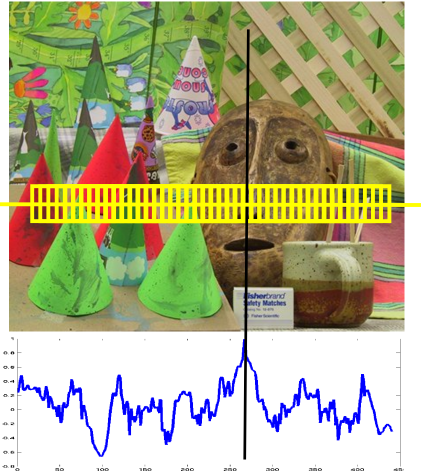
Challenges:

- Which window size to choose? Tradeoff: Ambiguity $\leftrightarrow$ Bleeding!
- Block matching = fronto-parallel assumption (often invalid!)

Another Example: Middlebury Cones Dataset

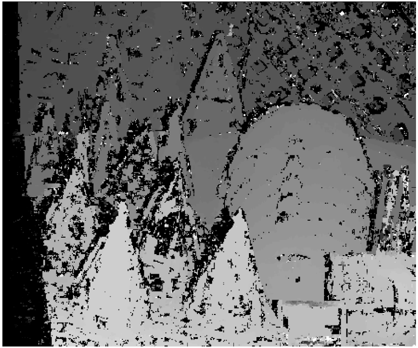
Left Image

Right Image

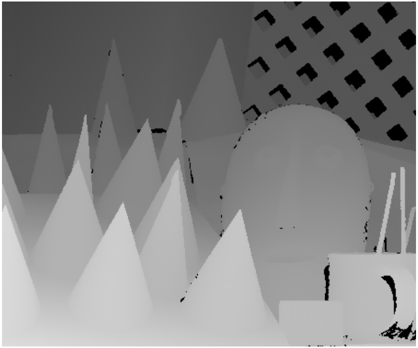


Match Score Values

Middlebury Cones Dataset: Block Matching Result

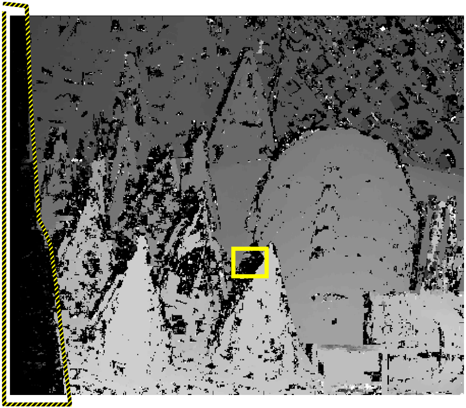


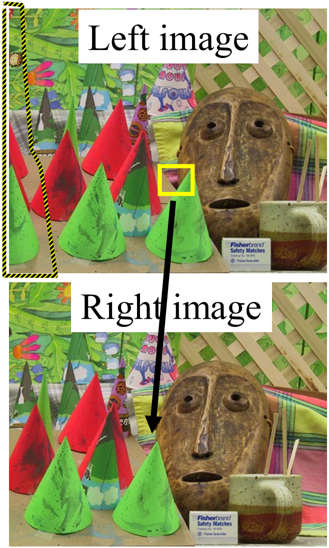
Computed disparities



Ground truth

Middlebury Cones Dataset: Half-Occlusions

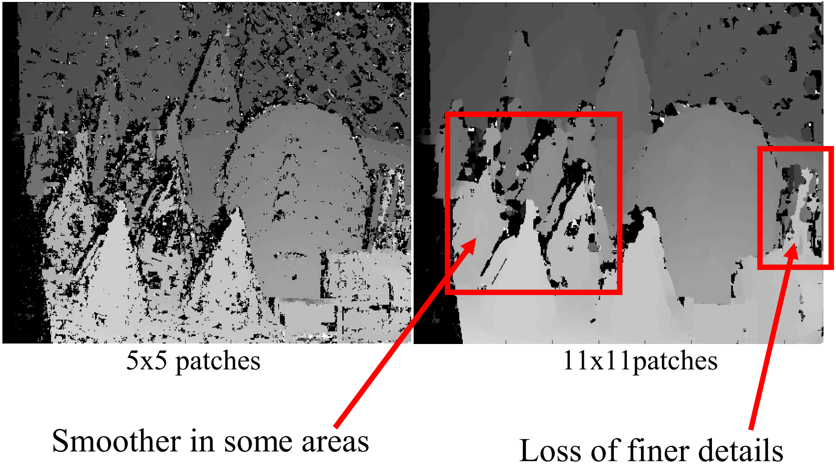




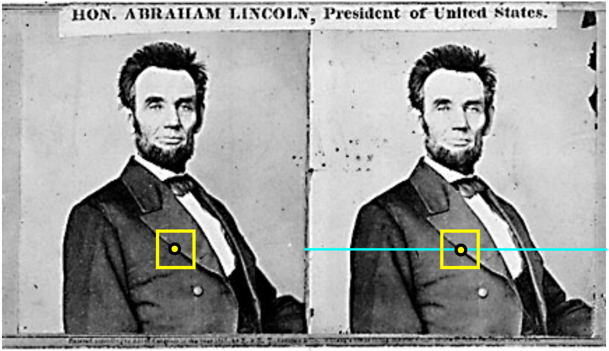
Middlebury Cones Dataset: Half-Occlusions

Middlebury Cones Dataset: Half-Occlusions

Middlebury Cones Dataset: Window Size

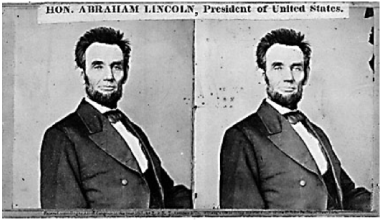


The Underlying Assumption: Similarity Constraint



- Corresponding regions in both images should look similar ...
- ... and non-corresponding regions should look different.
- When will the similarity constraint fail?

Similarity Constraint: Failure Cases



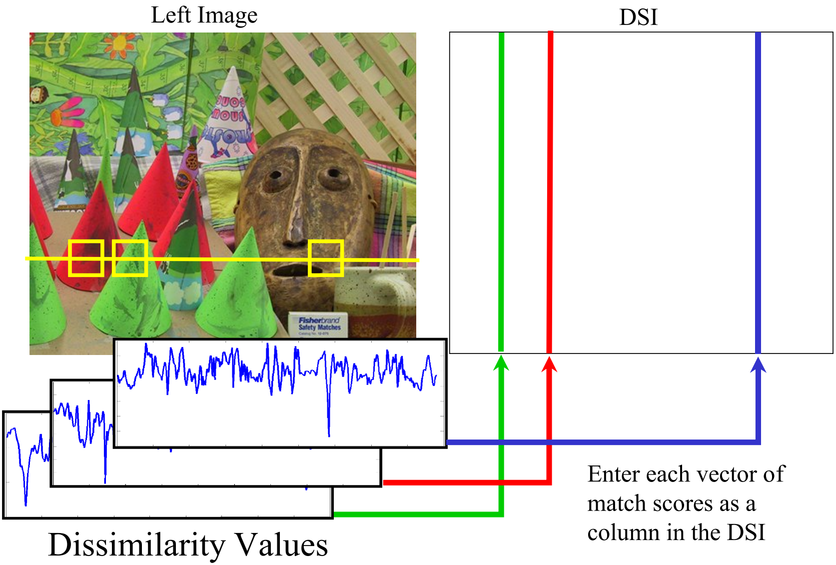
Textureless surfaces

Occlusions, repetition

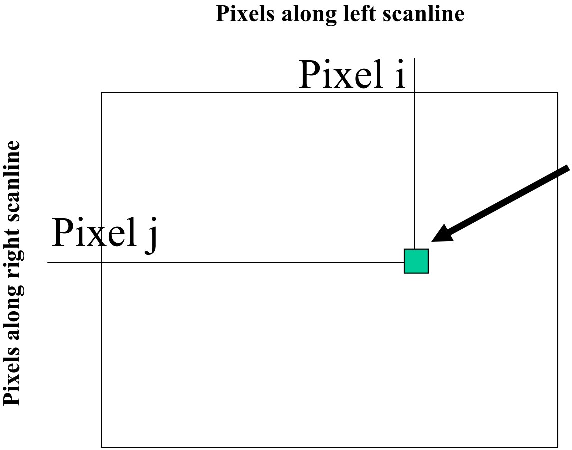
Non-Lambertian surfaces, specularities

To overcome these difficulties we need to incorporate our prior knowledge about the world!

Disparity Space Image (DSI)

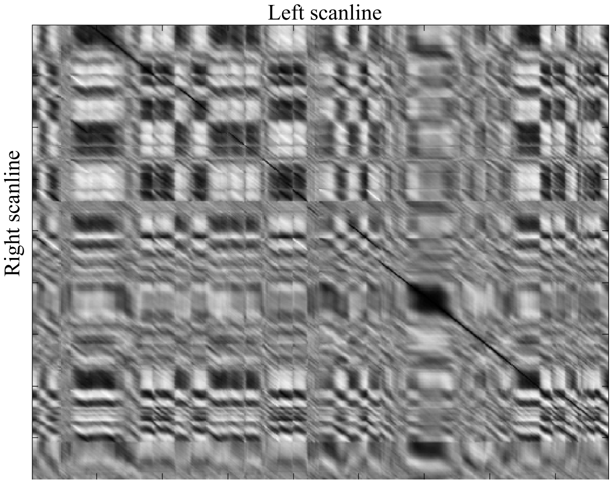


Disparity Space Image (DSI)

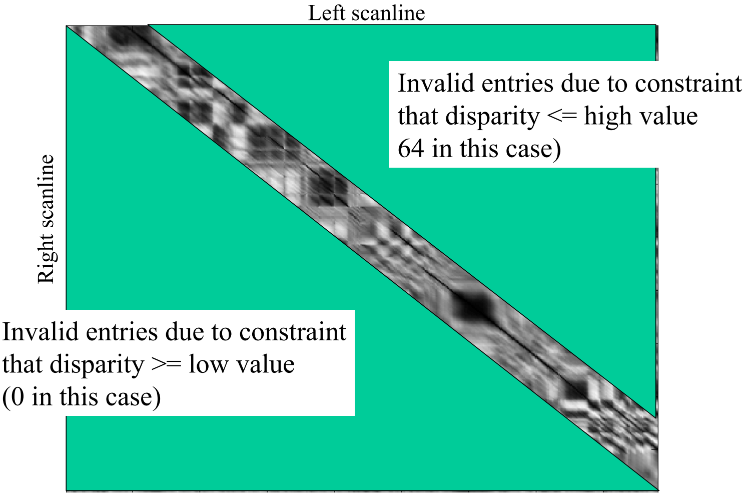


C(i,j) = Match score for patch centered at left pixel i with patch centered at right pixel j.

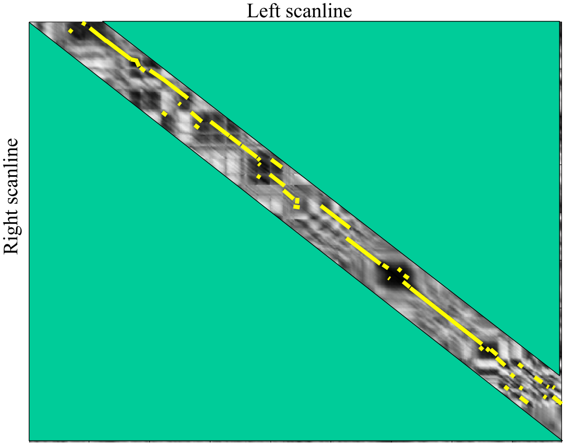
Disparity Space Image (DSI)



Disparity Space Image (DSI)

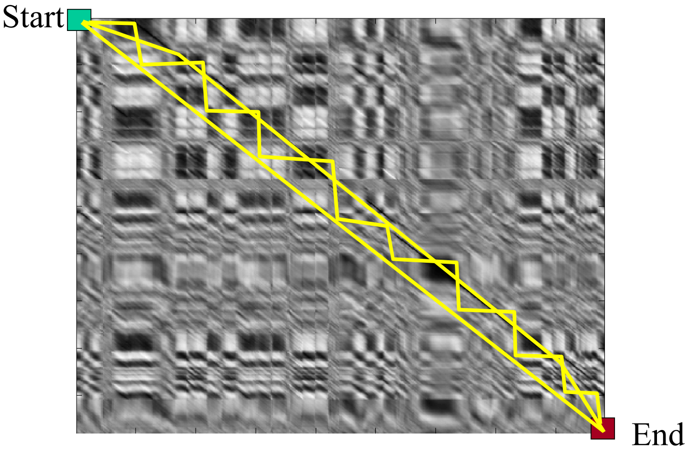


Winner-Takes-All Solution (Block Matching)



- Assigns each pixel in left scanline the “best” match in right scanline
- Is this the optimal solution?

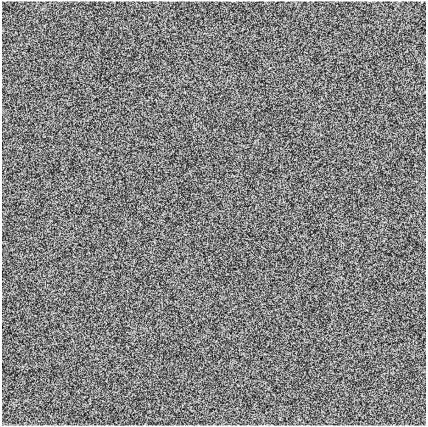
Alternative Solutions

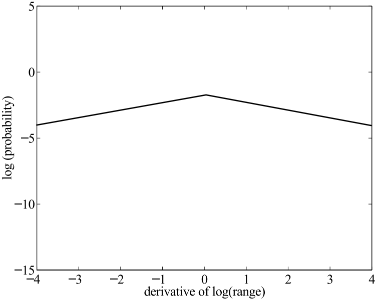


- Which solution should we choose?
- Do we have prior knowledge about this problem?

Thought Example

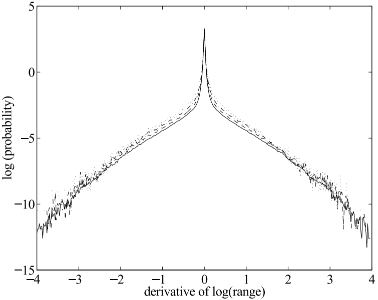
- Let's consider the block matching term as a likelihood
- What do we assume about neighboring disparities?





Thought Example

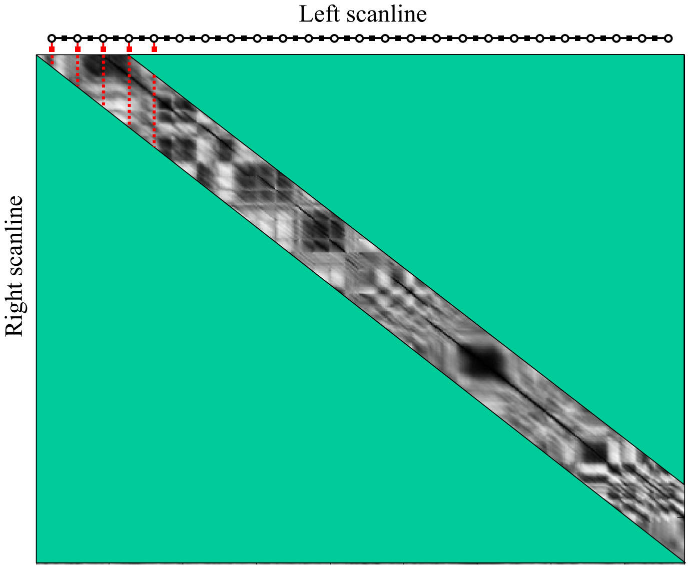
- How does the real world look like?
- The Brown range image database [Mumford et al.]



Question

Can MRFs help us to incorporate these statistics?

Chain MRF for Scanline Stereo



Chain MRF for Scanline Stereo

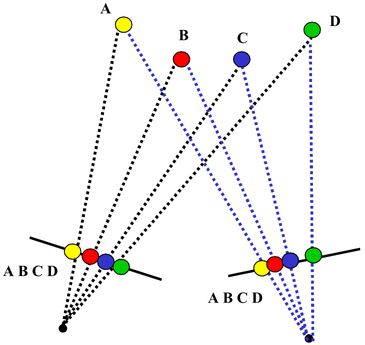
For each image row independently do:

- Setup a pairwise MRF *energy* (d_i=disparity at column i):

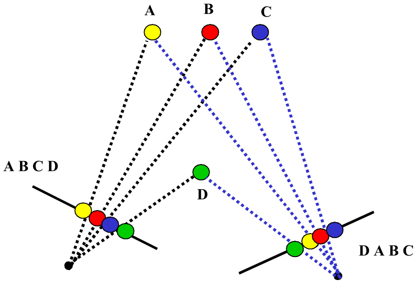
$$p(\mathbf{d}) \propto \exp\left\{-\sum_{i=1}^{N} \psi_{data}(d_i) - \lambda \sum_{i=1}^{N-1} \psi_{smooth}(d_i, d_{i+1})\right\}$$

- Disparities in scanline: $\mathbf{d} = \{d_1, \ldots, d_N\}$
- Unary terms: Matching cost $\psi_{data}(d)$
- Pairwise terms: Smoothness between adjacent pixels, e.g.:
  - Potts: $\psi_{smooth}(d, d') = [d \neq d']$
  - Truncated l_1: $\psi_{smooth}(d, d') = \min(|d - d'|, \tau)$
  - Truncated l_2: $\psi_{smooth}(d, d') = \min((d - d')^2, \tau)$
- Solve MRF using max-product BP, graph cuts, etc.
- This can be done in an optimal way since it is a chain!

More constraints you can use: Ordering

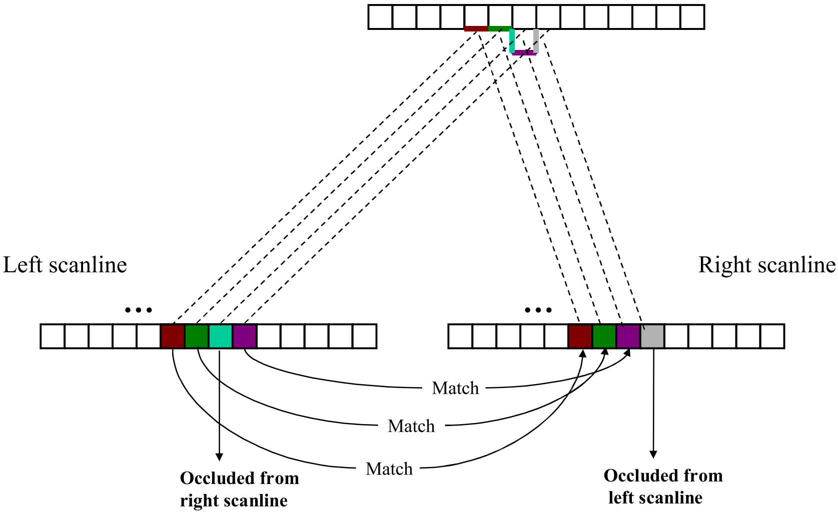


Ordering constraint…

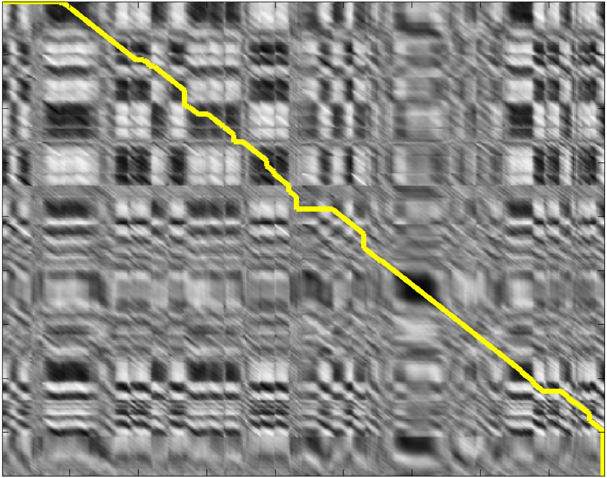


…and its failure

More constraints you can use: Occlusion



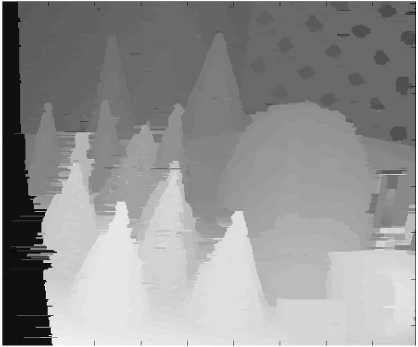
Chain MRF – MAP Solution



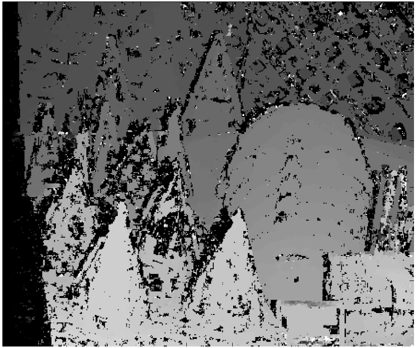
Chain MRF – MAP Solution

- Optimal solution – sounds great, right?
- Question: What is the catch?
- Independent processing of scanlines leads to streaking artifacts:

Result of DP alg with occlusion filling.



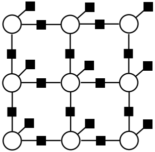
Result without DP (independent pixels)



Stereo MRF

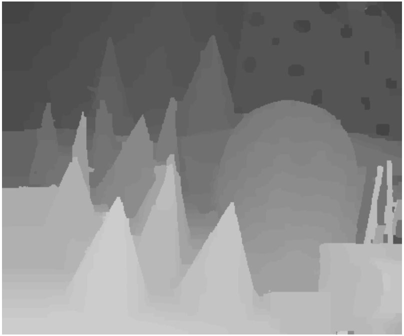
- What can we do to preserve inter-scanline consistency?
- Specify a loopy MRF on a grid instead of a chain MRF on individual scanlines and solve for the whole disparity map at once!

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

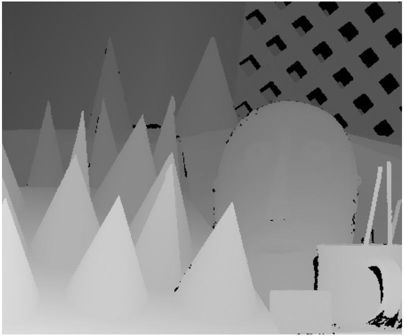


- Disparity image: $\mathbf{D}$
- $i \sim j$ indicates neighboring pixels on a 4-connected grid
- Unary terms: Matching cost $\psi_{data}(d)$
- Pairwise terms: Smoothness between adjacent pixels, e.g.:
  - Potts: $\psi_{smooth}(d, d') = [d \neq d']$
  - Truncated l_1: $\psi_{smooth}(d, d') = \min(|d - d'|, \tau)$
  - Truncated l_2: $\psi_{smooth}(d, d') = \min((d - d')^2, \tau)$
- Solve MRF approximately using max-product BP, graph cuts, etc.

Stereo MRF – Results



Inference Results



Ground Truth

Semiglobal Matching (SGM)

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

- Unary terms: Matching cost $\psi_{data}(d)$
- Pairwise terms ($0 < \lambda_1 < \lambda_2$):

$$\psi_{smooth}(d, d') = \begin{cases} 0 & \text{if } d = d' \\ \lambda_1 & \text{if } |d - d'| = 1 \\ \lambda_2 & \text{otherwise} \end{cases}$$

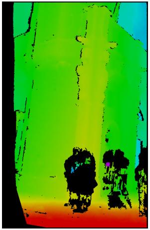
- Aggregates cost in each image direction (4/8) individually
- Afterwards: Winner-takes-all
- SGM can be interpreted as one iteration of message passing (TRW)
- It is extremely fast and produces good results!

Programming Exercise

Using Python, NumPy, OpenCV and MeshLab ...

... you will create your own 3D model from rectified images!


$\Rightarrow$



$\Rightarrow$