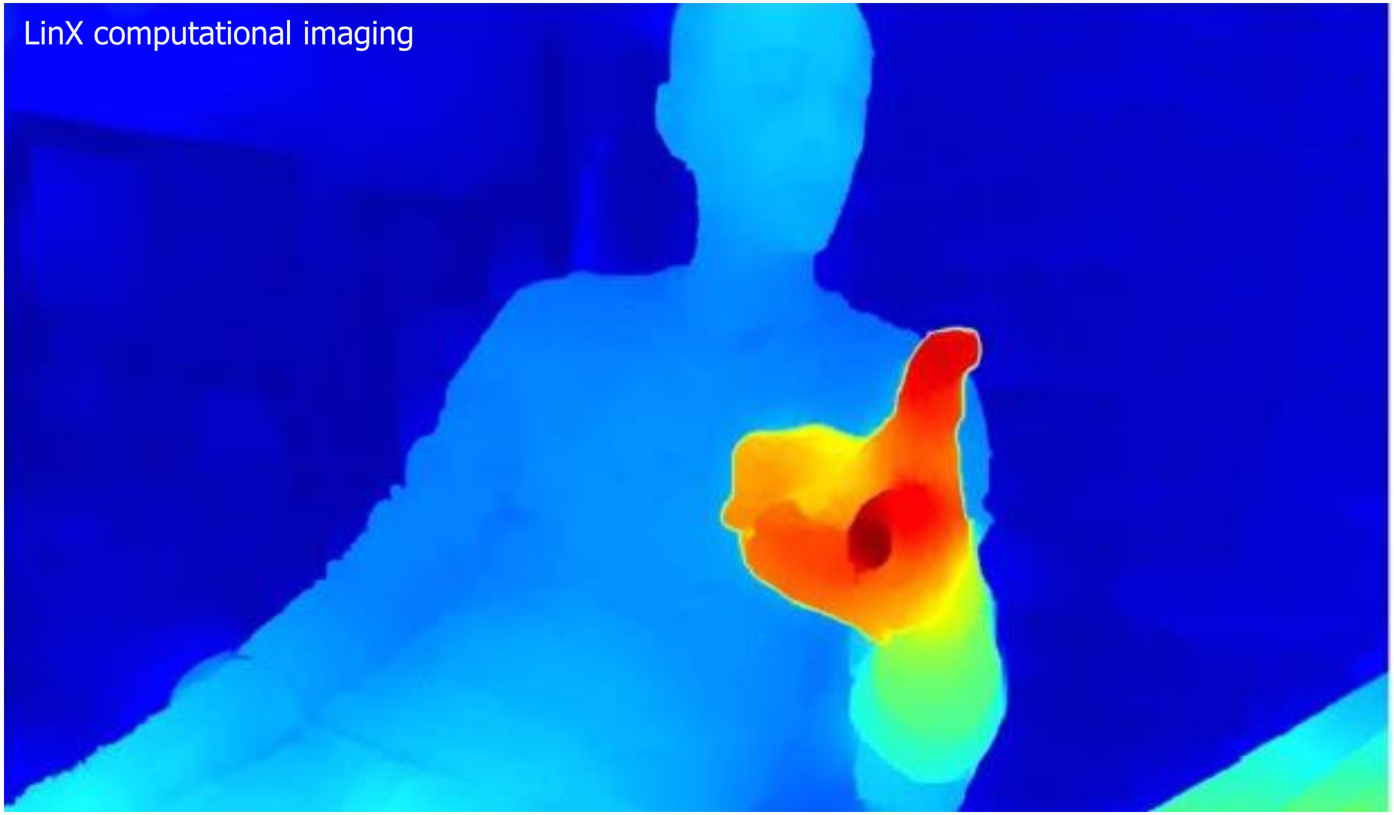
CS484/684 Computational Vision

Dense Stereo



Dense Stereo

towards **dense** 3D reconstruction

- (dense) stereo is an example of **dense correspondence**
- another example is dense motion estimation (*optical flow*)

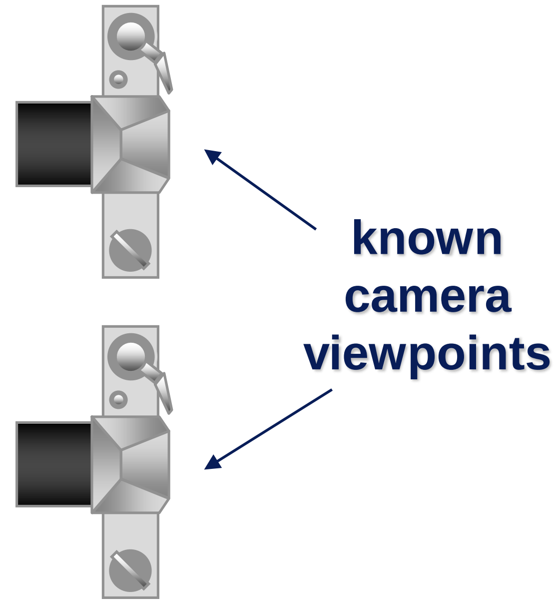
But, **stereo is simpler** since the search for correspondences
is restricted to 1D epipolar lines (versus 2D search for non-rigid motion)

Dense Stereo

- camera rectification for stereo pairs
- local stereo methods (windows)
- scan-line stereo correspondence
  - optimization via DP, Viterbi, Dijkstra
- global stereo
  - optimization via multi-layered graph cuts

Szeliski, Chapter 11

Stereo vision



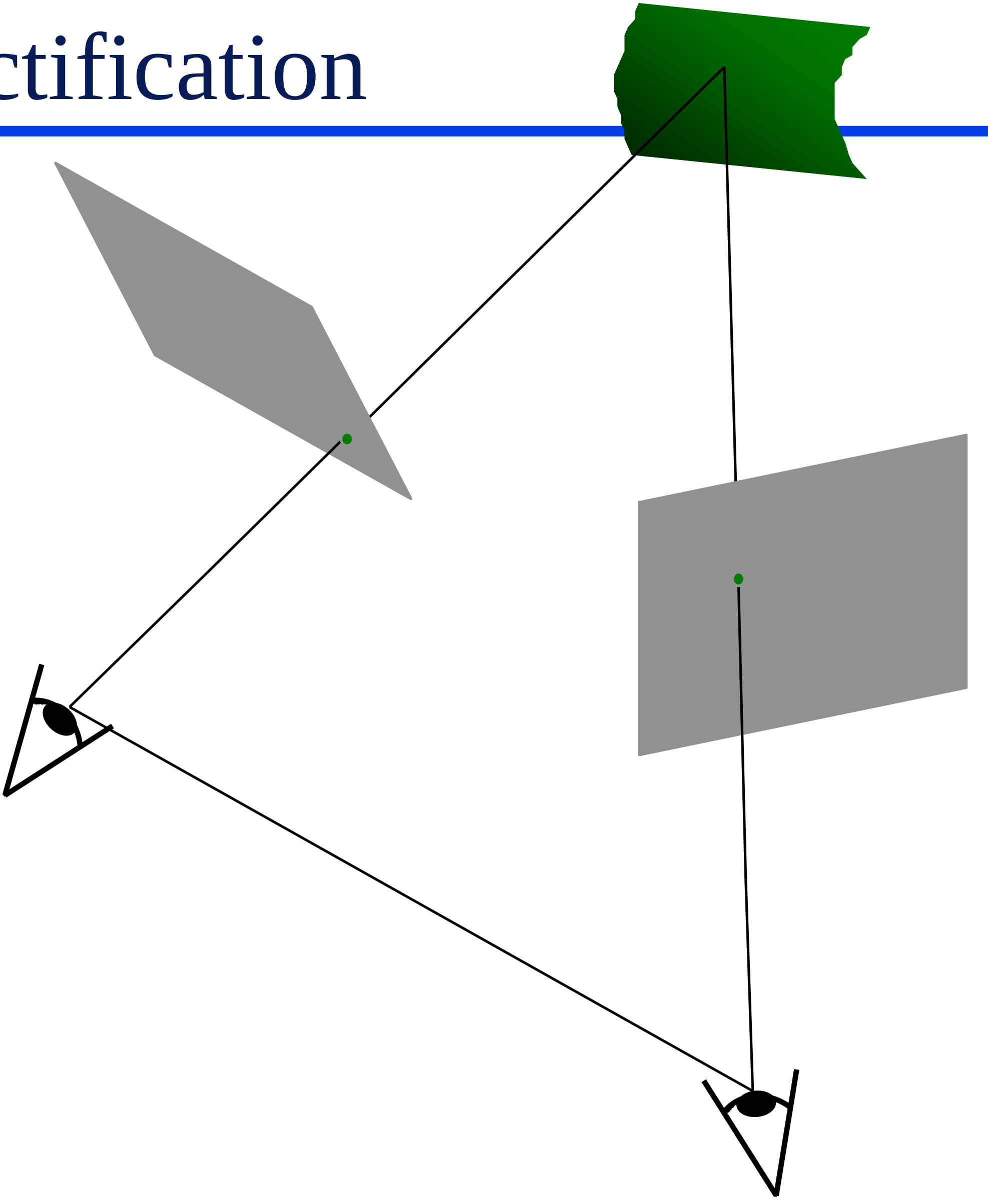
Two views of the same scene from slightly different point of view

Also called, narrow baseline stereo.

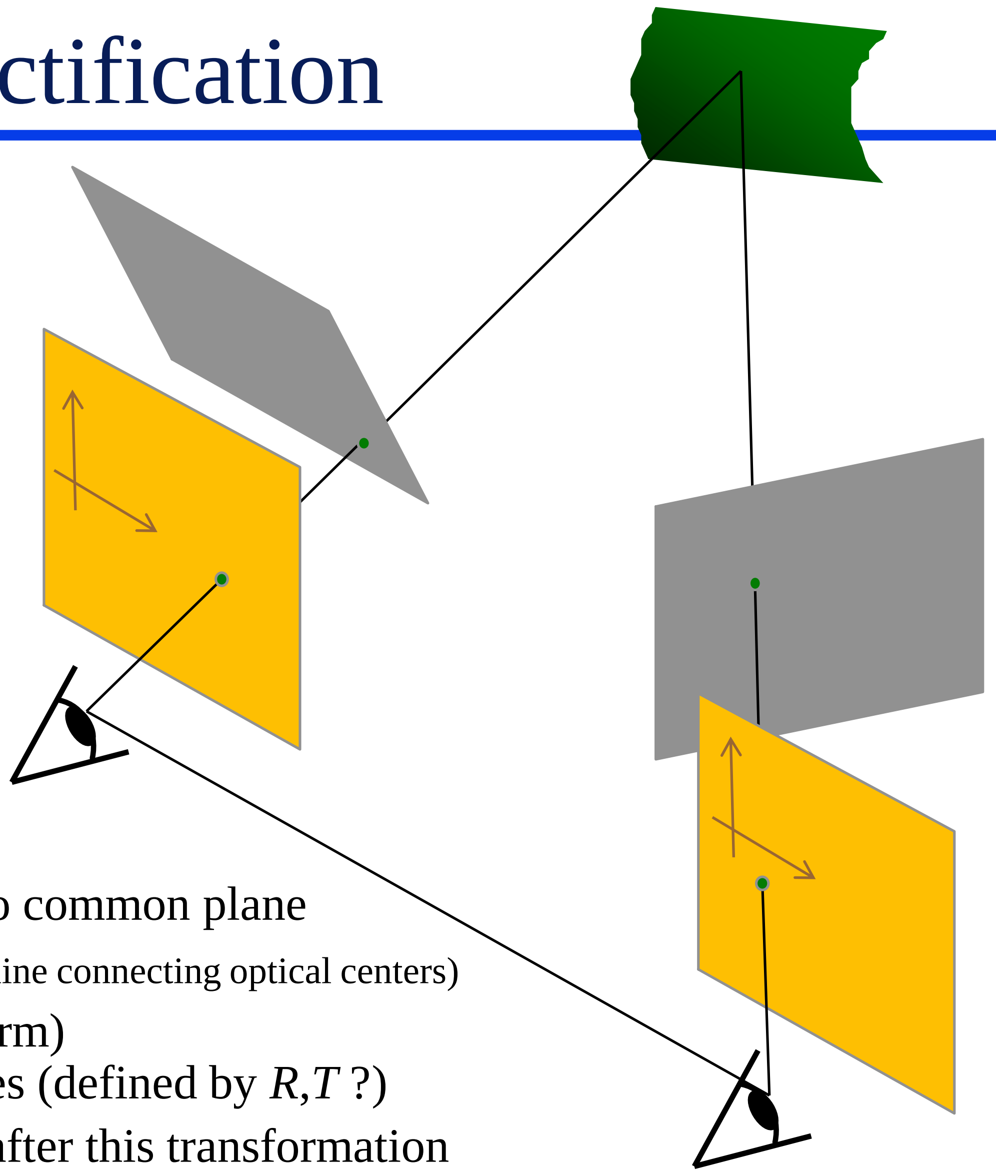
Motivation: - smaller difference in views allows to find **more matches** (Why?)

- scene reconstruction can be formulated via simple **depth map**

Stereo image rectification

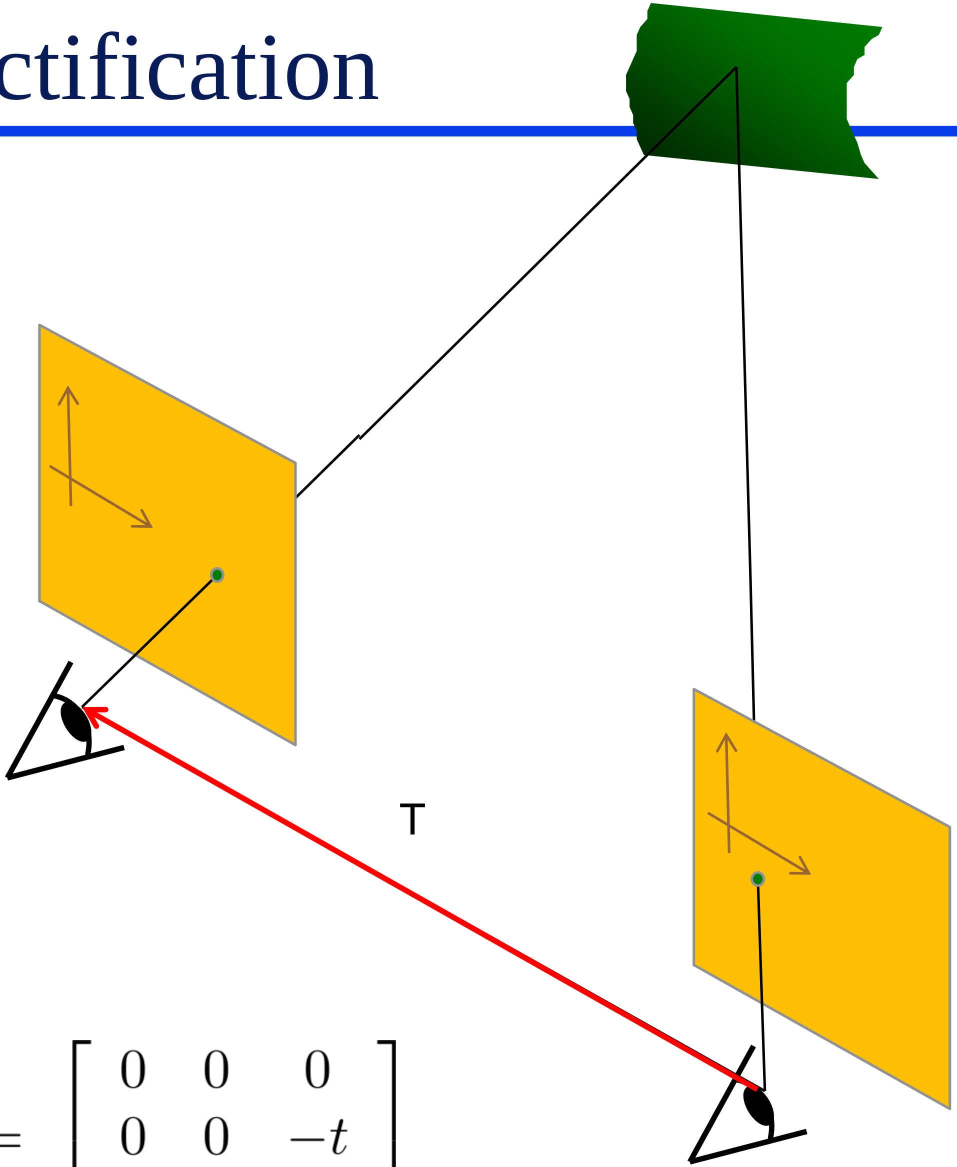


Stereo image rectification



- Image Reprojection
  - reproject image planes onto common plane
    parallel to the baseline (i.e. line connecting optical centers)
  - homographies (3x3 transform)
    applied to both input images (defined by R,T ?)
  - pixel motion is horizontal after this transformation
  - C. Loop and Z. Zhang. [Computing Rectifying Homographies for Stereo Vision](). IEEE Conf. Computer Vision and Pattern Recognition, 1999.

Stereo image rectification



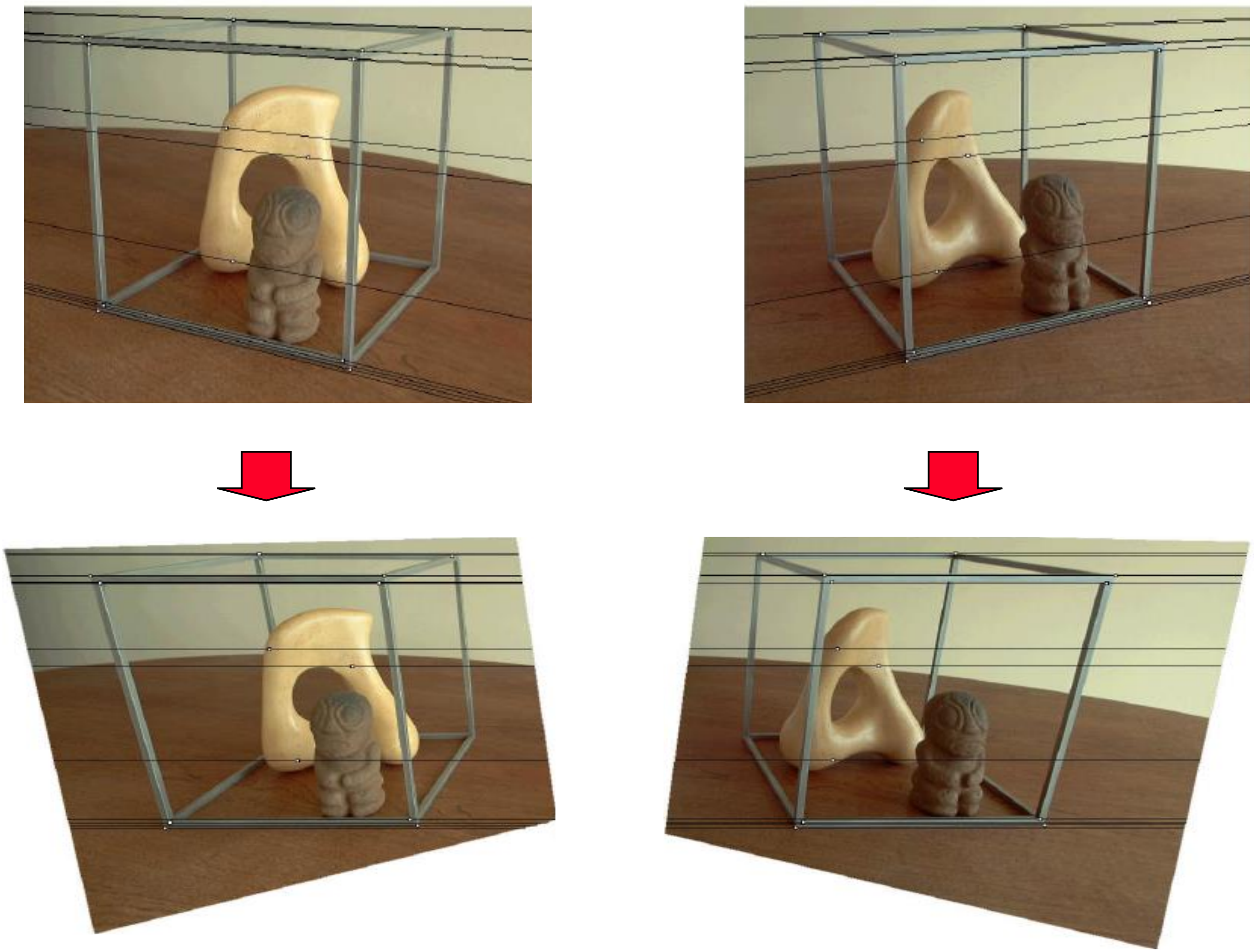
- Epipolar constraint:

$$R = I$$

$$T = \begin{bmatrix} t \\ 0 \\ 0 \end{bmatrix}$$

$$\Rightarrow E = [T]_\times R = [T]_\times = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -t \\ 0 & t & 0 \end{bmatrix}$$

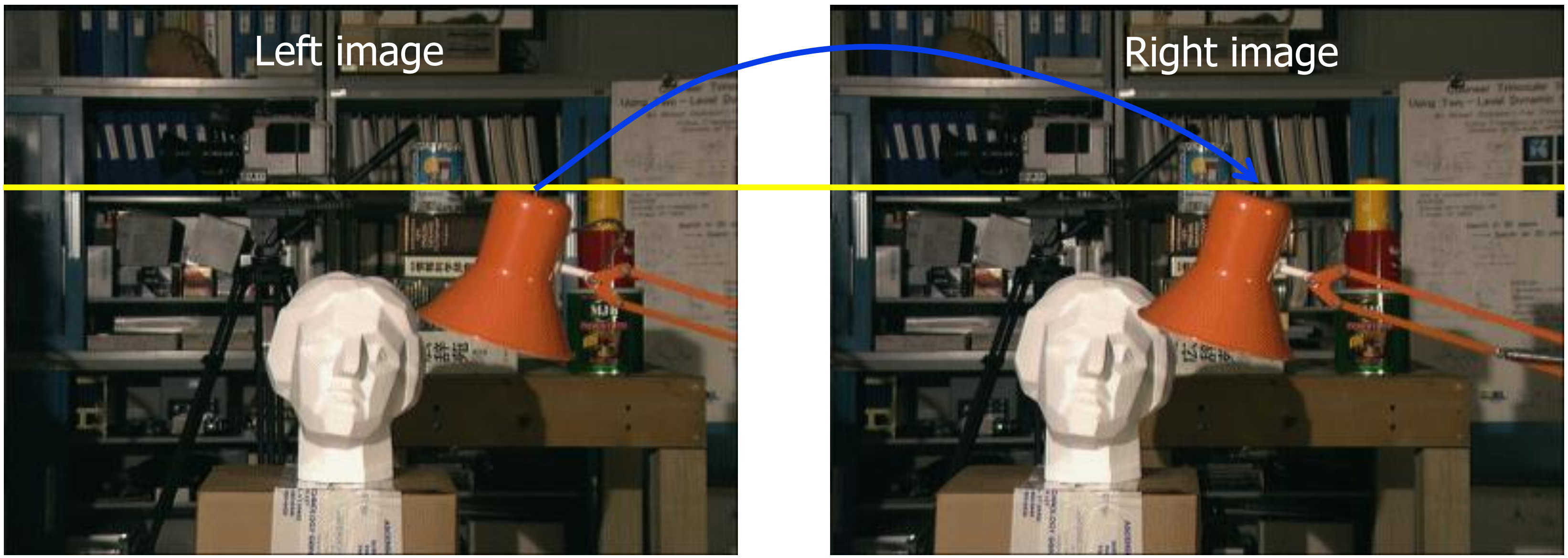
Stereo Rectification



Note projective distortion.
It will be much bigger
if images are taken from
very different view points
(large baseline).

in this example the base line C_1C_2 is parallel to cube edges.

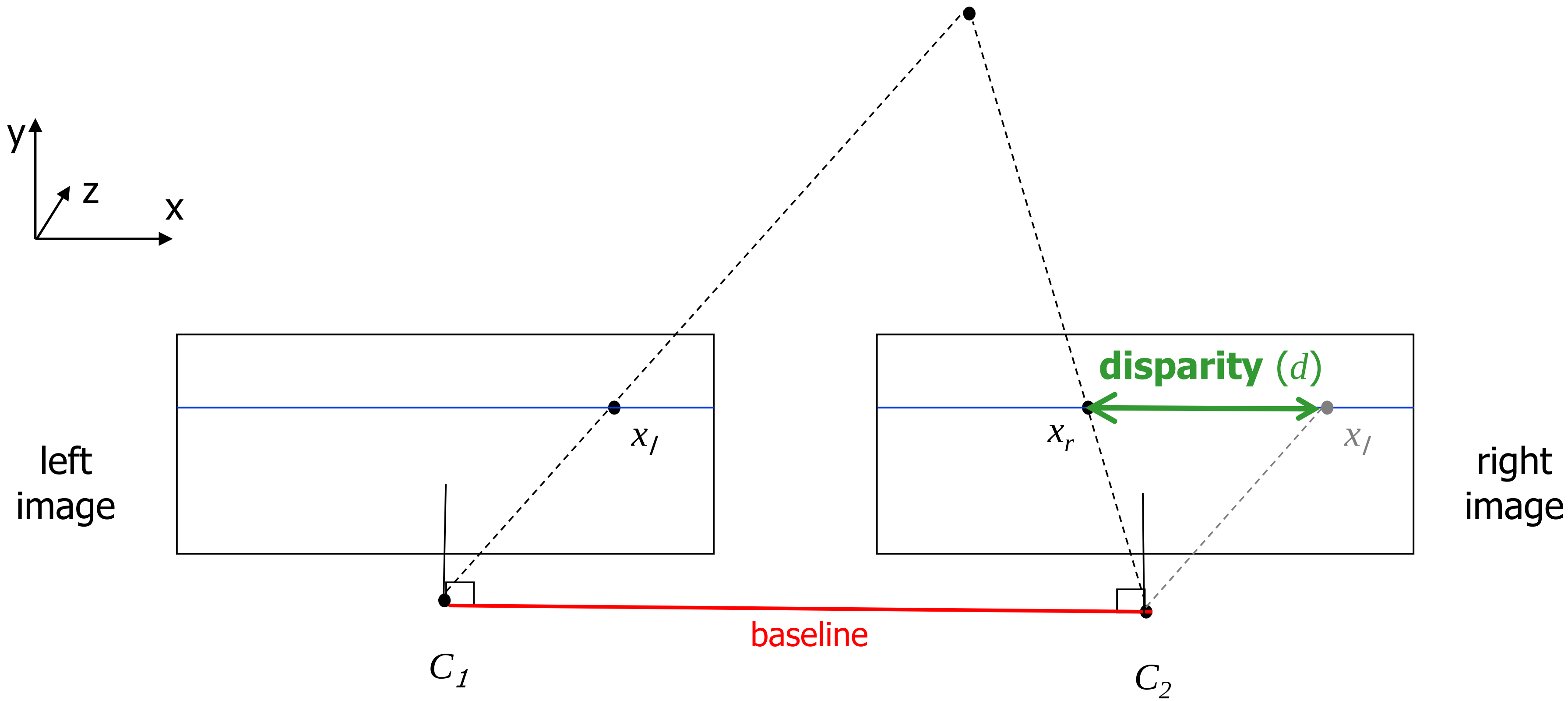
Stereo as a *correspondence* problem



(After rectification) all correspondences are along the same <u>horizontal scan lines</u>

(epipolar lines)

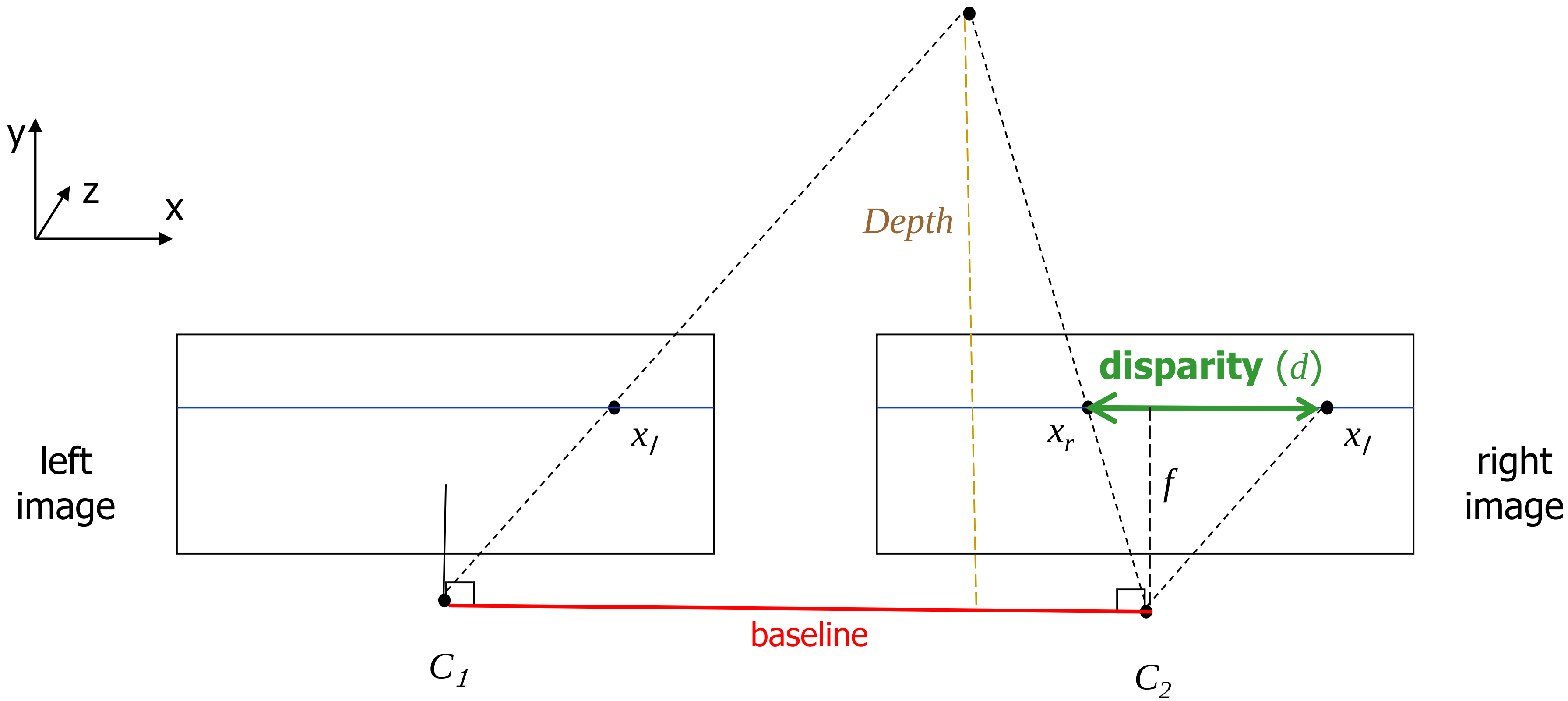
Rectified Cameras



epipolar lines are parallel to the x axis

difference between the x-coordinates of x_l and x_r is called the disparity

Rectified Cameras



$$\text{Depth} = |C_1C_2| \cdot f \,/\, d$$

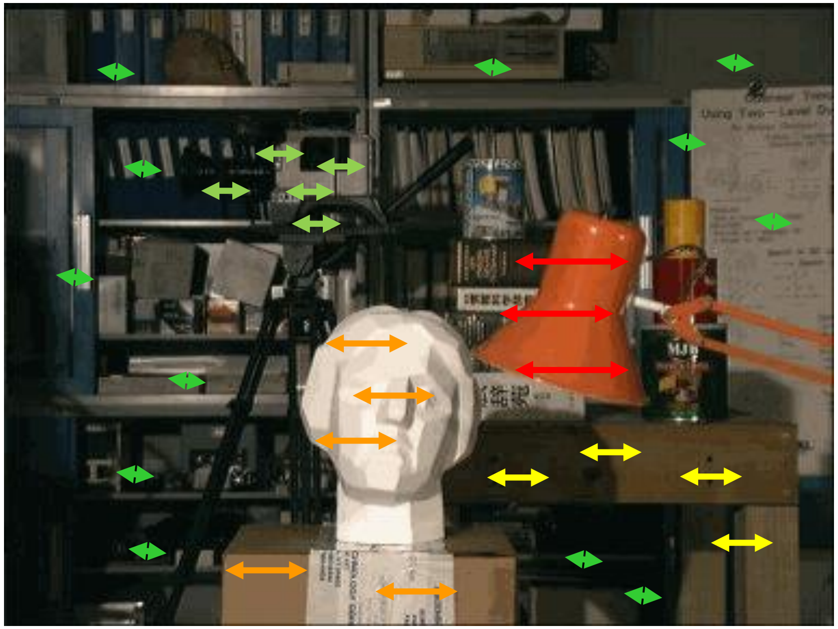
Stereo

Correspondences are described by shifts along horizontal scan lines (epipolar lines)

which can be represented by scalars (**disparities)**

Stereo

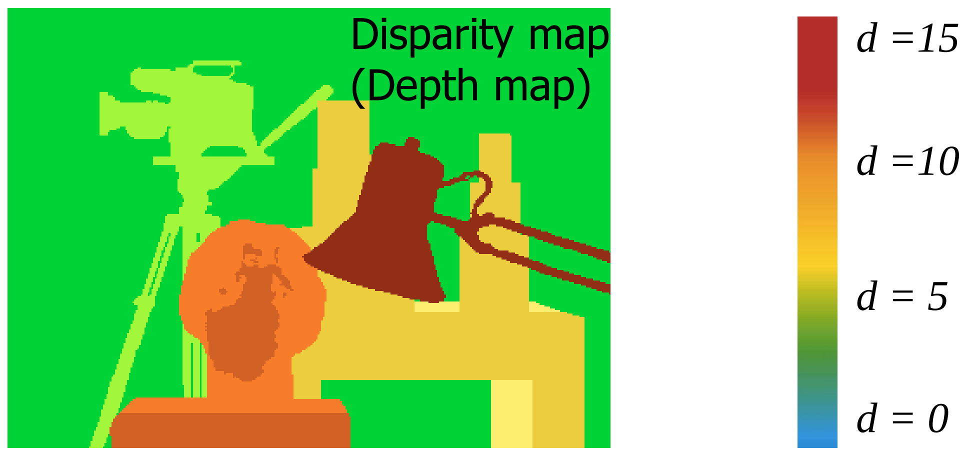
closer objects (smaller depths) correspond to **larger disparities**



Correspondences are described by shifts along horizontal scan lines (epipolar lines)

which can be represented by scalars (**disparities)**

Stereo

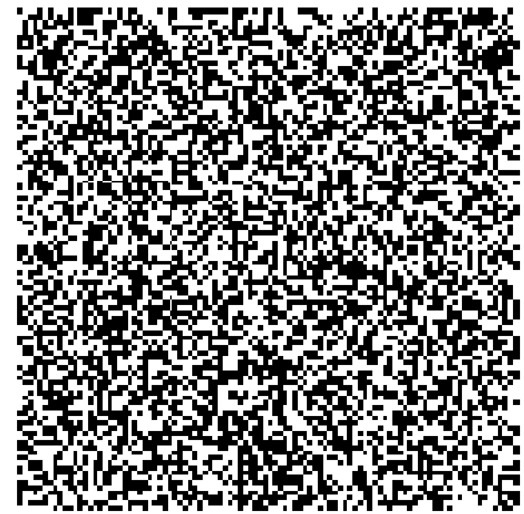


- If x-shifts (disparities) are known for all pixels in the left (or right) image then we can visualize them as a **disparity map** – scalar valued function $d(p)$
- larger disparities correspond to closer objects

Stereo Correspondence problem

- Human vision can solve it

  (even for "random dot" stereograms)



- Can computer vision solve it?

Maybe

see *Middlebury Stereo Database*
for the state-of-the art results
http://cat.middlebury.edu/stereo/

Stereo

- Window based
  - Matching rigid windows around each pixel
  - Each window is matched independently
- Scan-line based approach
  - Finding coherent correspondences for each scan-line
  - Scan-lines are independent
    - DP, shortest paths
- Muti-scan-line approach
  - Finding coherent correspondences for all pixels
    - Graph cuts

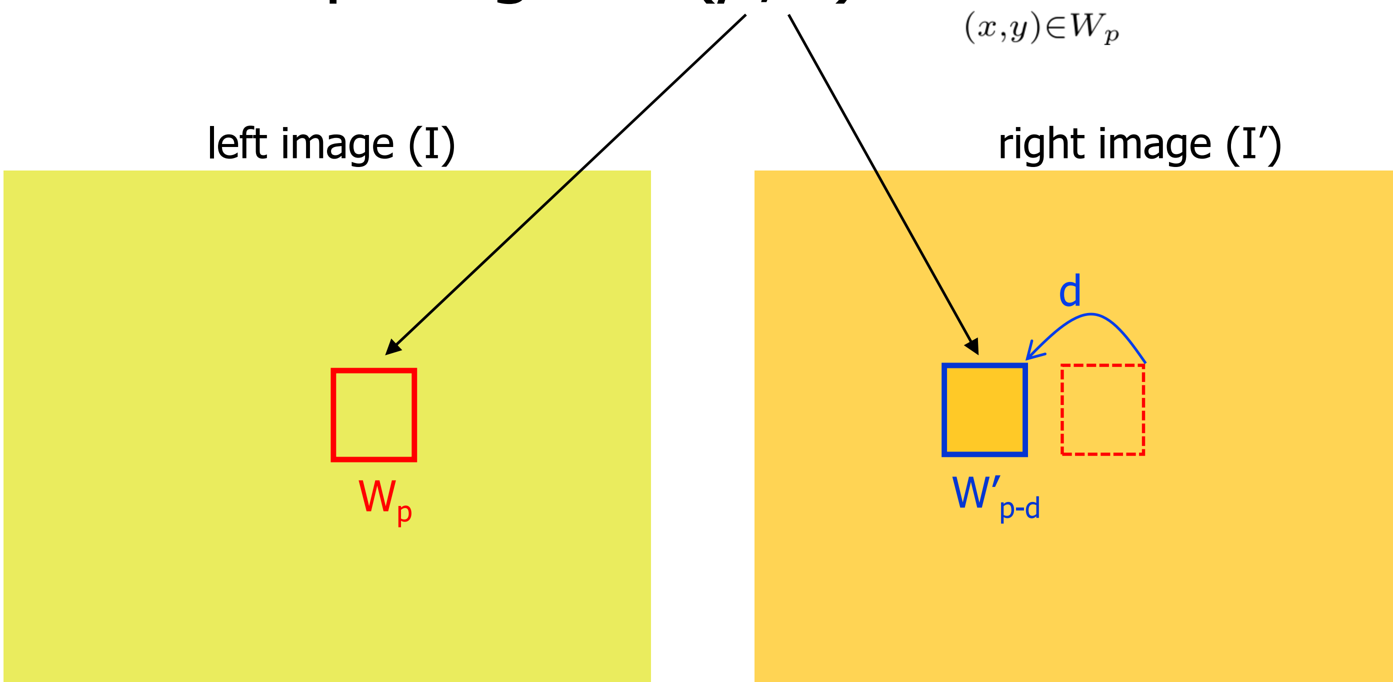
Stereo Correspondence problem
Window based approach

- For any given point p in left image consider window (or image patch) W_p around it
- Find matching window W_q on the same scan line in the right image that looks most similar to W_p

naïve implementation
has |I|*|d|*|W| operations

SSD (sum of squared differences) approach

computing SSD(*p*,*d*) $= \sum_{(x,y)\in W_p} (I(x,y) - I'(x-d,y))^2$

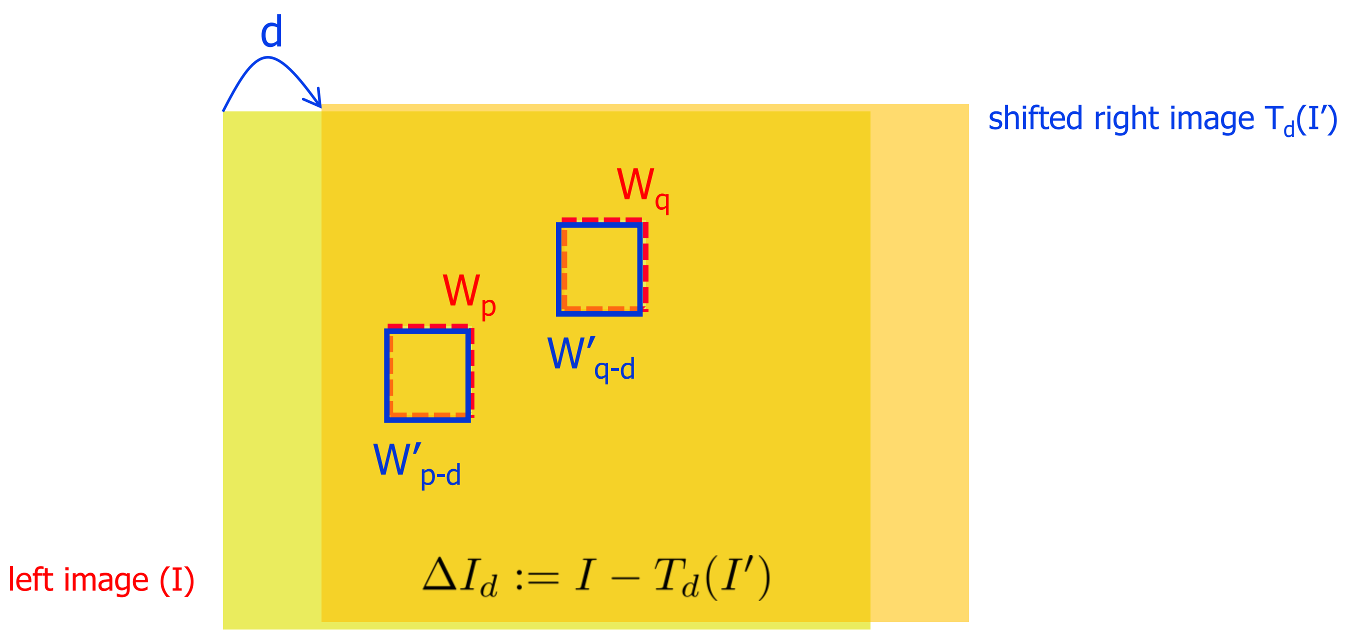


for any pixel *p* compute SSD between windows W_p and W'_{p-d}
for all disparities *d* (in some interval [*min_d, max_d*])

then $\boxed{\hat{d}_p = \arg\min_d SSD(p,d)}$

computing SSD

- For each fixed *d* can get SSD(*p*,*d*) at all points *p*



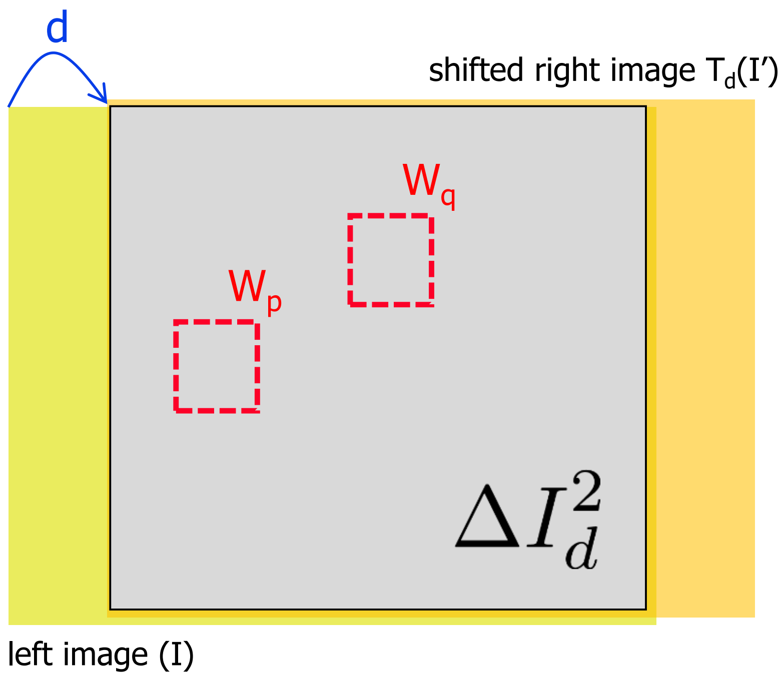
Compute the difference between the left image I and the shifted right image $T_d(I')$

$$\Delta I_d(x,y) := I(x,y) - I'(x-d,y)$$

Then, SSD(p,d) between W_p and W'_{p-d} is equivalent to $\sum_{(x,y)\in W_p} \Delta I_d^2(x,y)$

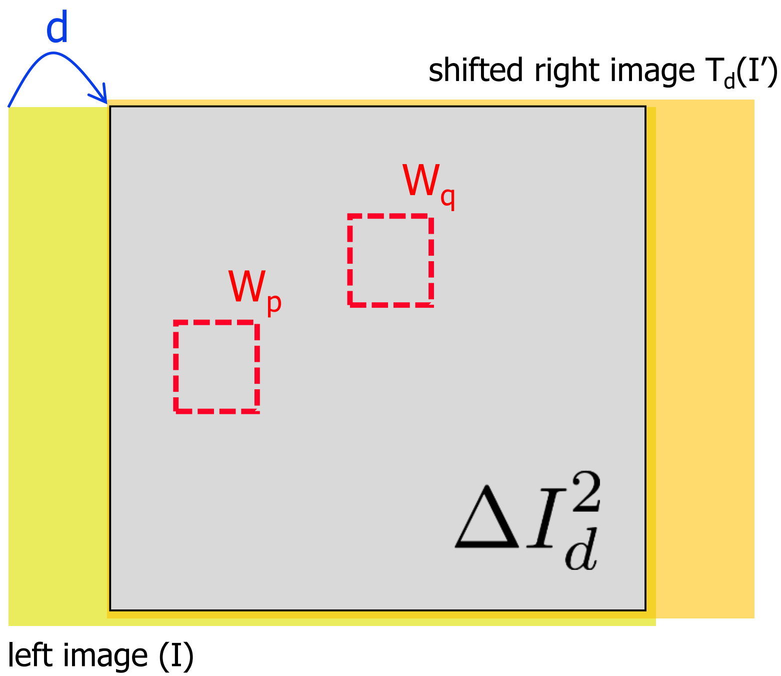
computing SSD

- For each fixed disparity d $\quad \mathrm{SSD}(p,d) = \sum_{(x,y)\in W_p} \Delta I_d^2(x,y)$



computing SSD

- For each fixed disparity d $\quad \text{SSD}(p,d) = \sum_{(x,y) \in W_p} \Delta I_d^2(x,y)$



Need to sum pixel values $\quad f(x,y) \equiv \Delta I_d^2(x,y)$

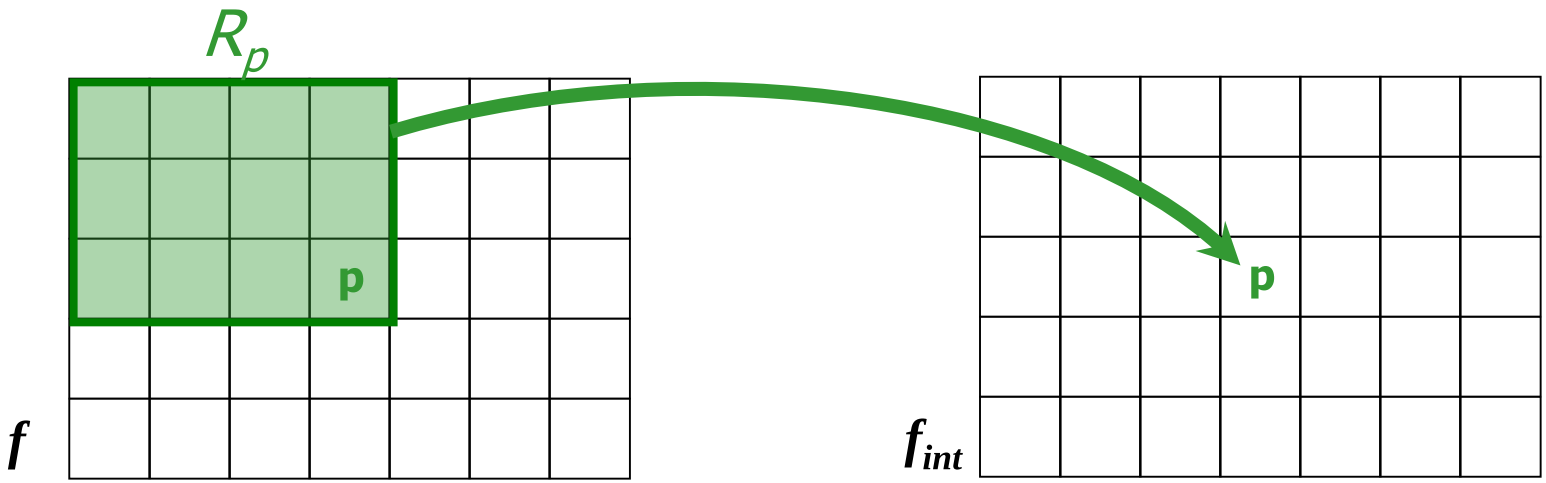
at all possible windows

general trick:

"Integral Images"

$$f_{int}(p) := \sum_{q \in R_p} f(q)$$

- Define integral image $f_{int}(p)$ as the sum (integral) of image f over pixels in rectangle $R_p := \{q \mid \text{“}q \leq p\text{”}\}$



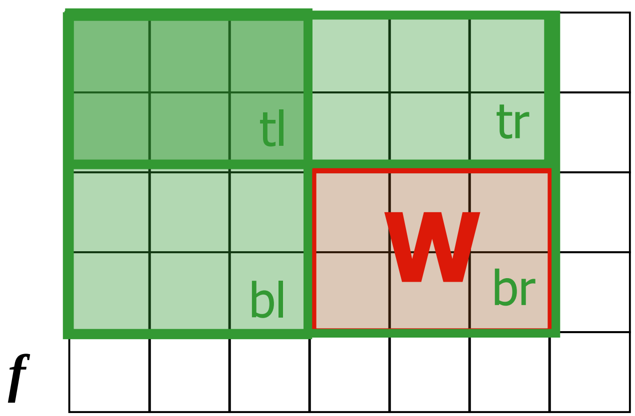
- Can compute $f_{int}(p)$ for all p in two passes over image f (How?)

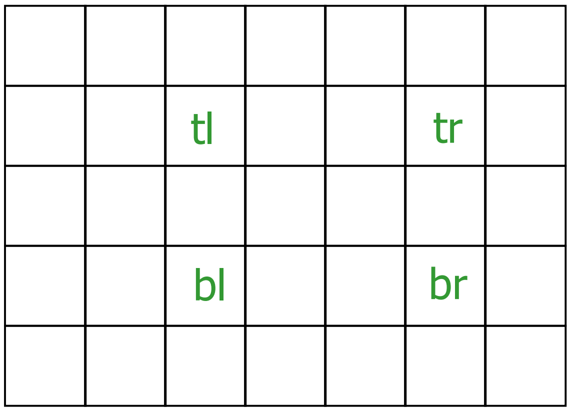
general trick:

"Integral Images"

$$f_{int}(p) := \sum_{q \in R_p} f(q)$$

- Define integral image $f_{int}(p)$ as the sum (integral) of image f over pixels in rectangle $R_p := \{q \mid "q \leq p"\}$





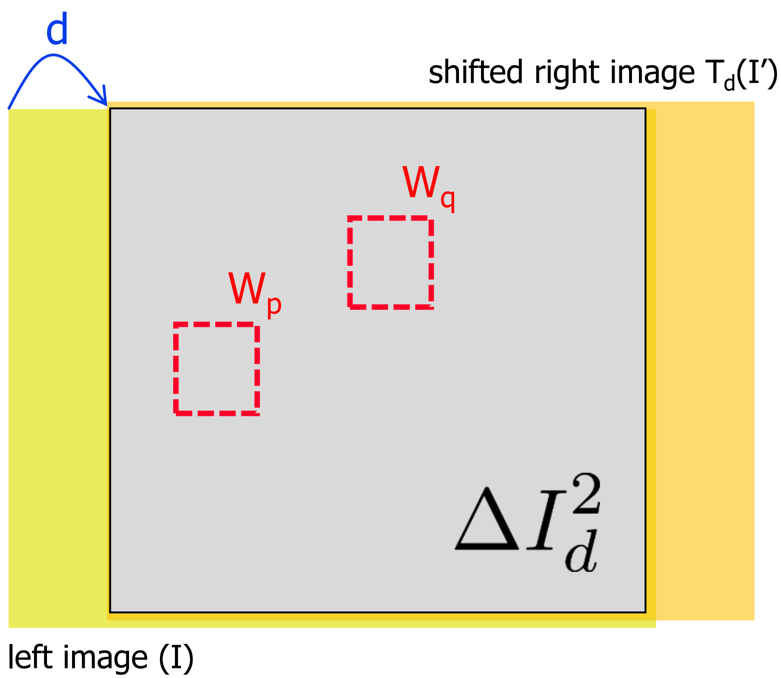
f_{int}

- Now, for any W the sum (integral) of f inside that window can be computed as $\sum_{q \in W} f(q) = f_{int}(\text{br}) - f_{int}(\text{bl}) - f_{int}(\text{tr}) + f_{int}(\text{tl})$

computing SSD

- For each fixed disparity d $\quad \mathrm{SSD}(p,d) = \sum_{(x,y)\in W_p} \Delta I_d^2(x,y)$

$$\sum_{x,y\in\Box} f(x,y)$$



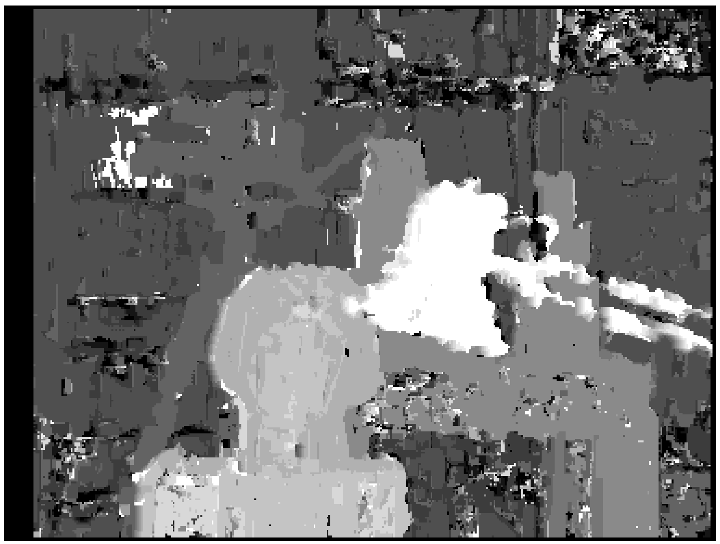
Now, the sum of ΔI_d^2 at any window $\Box$
takes 4 operations <u>independently of window size</u>

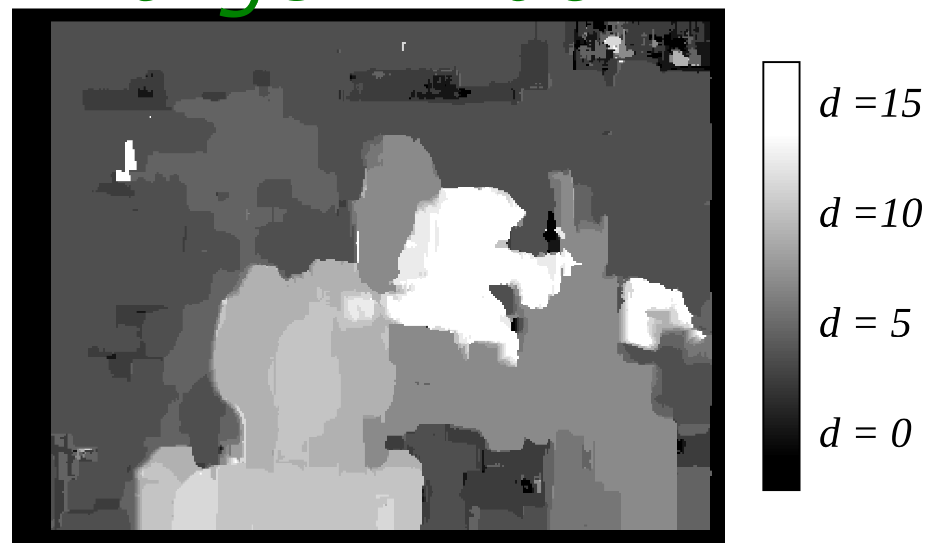
=> O(|I|*|d|) window-based stereo algorithm

Problems with Fixed Windows

disparity maps $\hat{d}_p = \arg\min_d SSD(p,d)$ for:

small window | *large window*





- better at boundaries
- noisy in low texture areas

- better in low texture areas
- blurred boundaries

Q: what do we implicitly assume when using low SSD(d,p) at a window around pixel p as a criteria for “good” disparity d ?

window algorithms

- Maybe variable window size (pixel specific)?
  - What is the right window size?
  - Correspondences are still found <u>independently</u> at each pixel (no coherence)

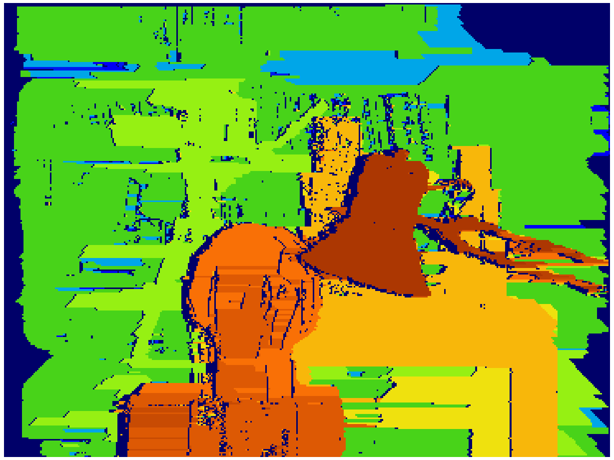
- All window-based solutions can be though of as "local" solutions - but very fast!

need priors
to compensate
for local data ambiguity

- How to go to "global" solutions?
  - use *objectives* (a.k.a. *energy* or *loss* functions)
    - *regularization* (e.g. spatial coherence)
  - optimization

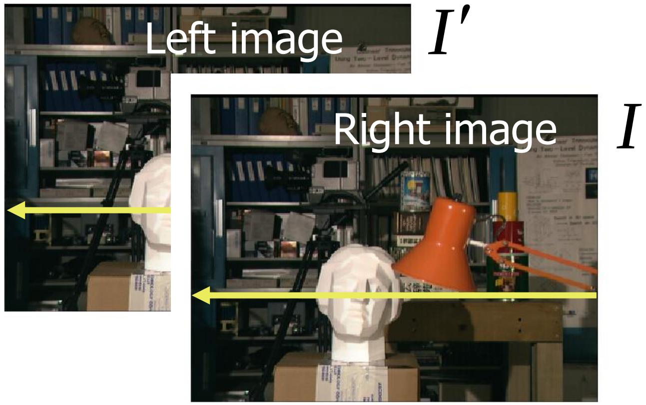
Stereo Correspondence problem
Scan-line approach

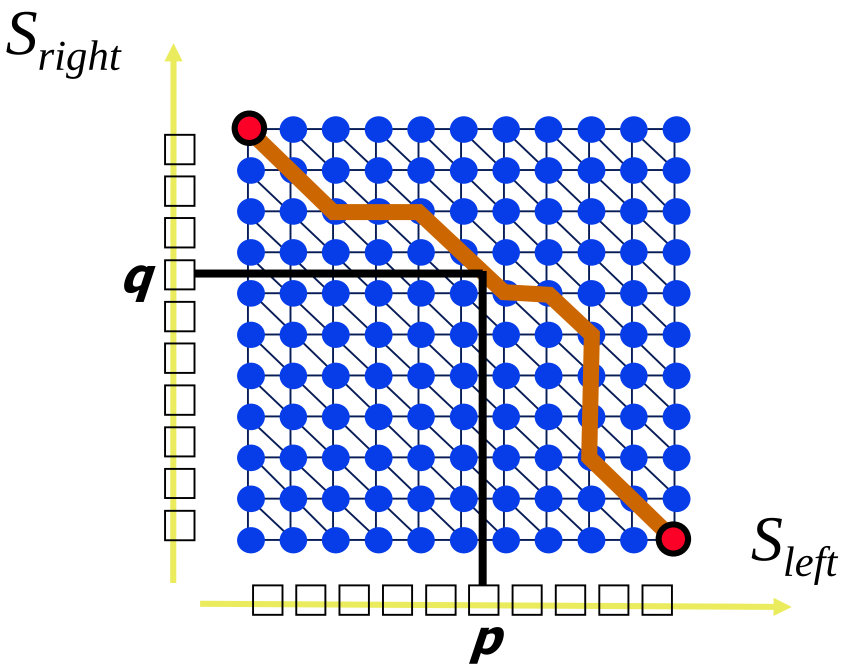
- Scan-line stereo
  - coherently match pixels in each scan line
  - DP or shortest paths work (easy 1D optimization)
  - Note: scan lines are still matched <u>independently</u>
    - – streaking artifacts



“Shortest paths” for Scan-line stereo

e.g. Ohta&Kanade’85, Cox at.al.’96

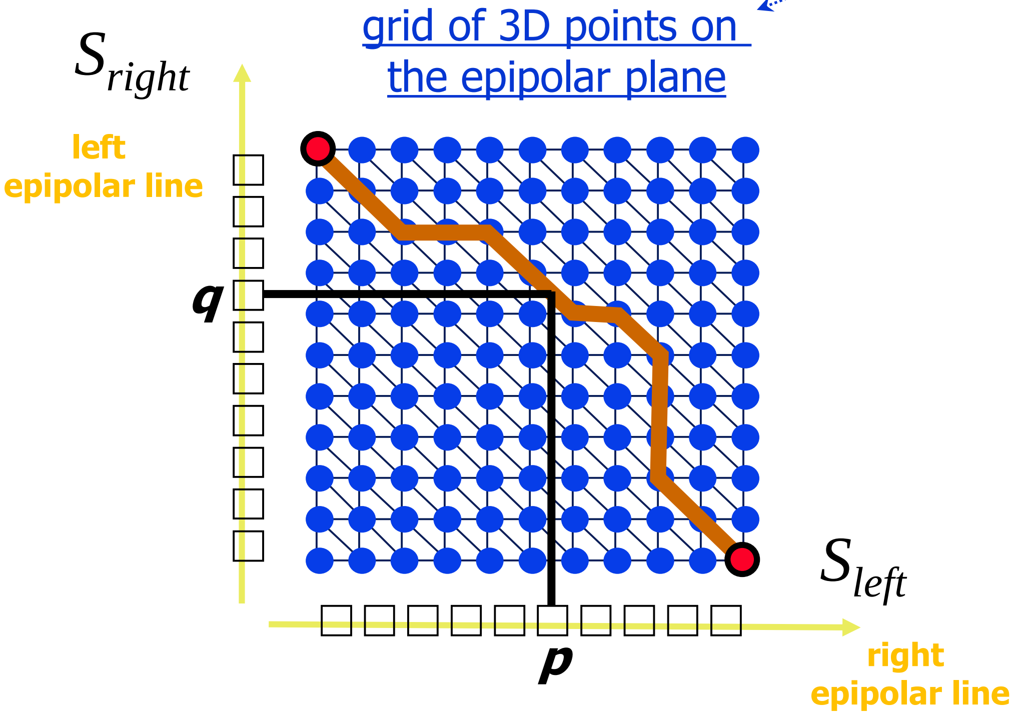


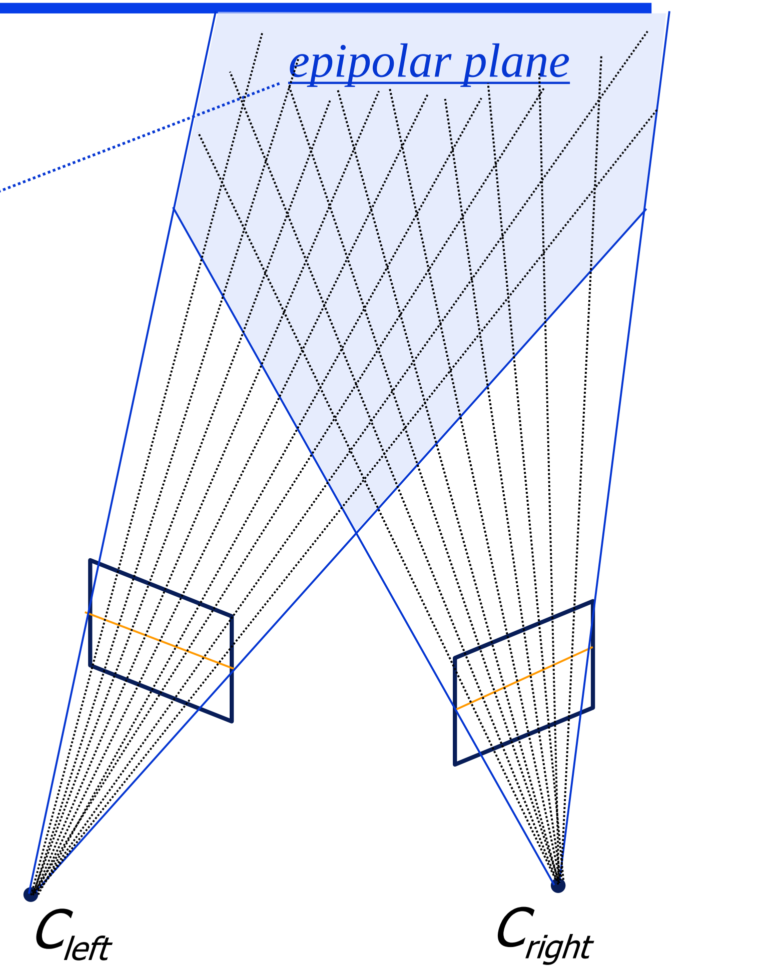


a **path** on this graph represents a matching function

"Shortest paths" for Scan-line stereo

3D interpretation:

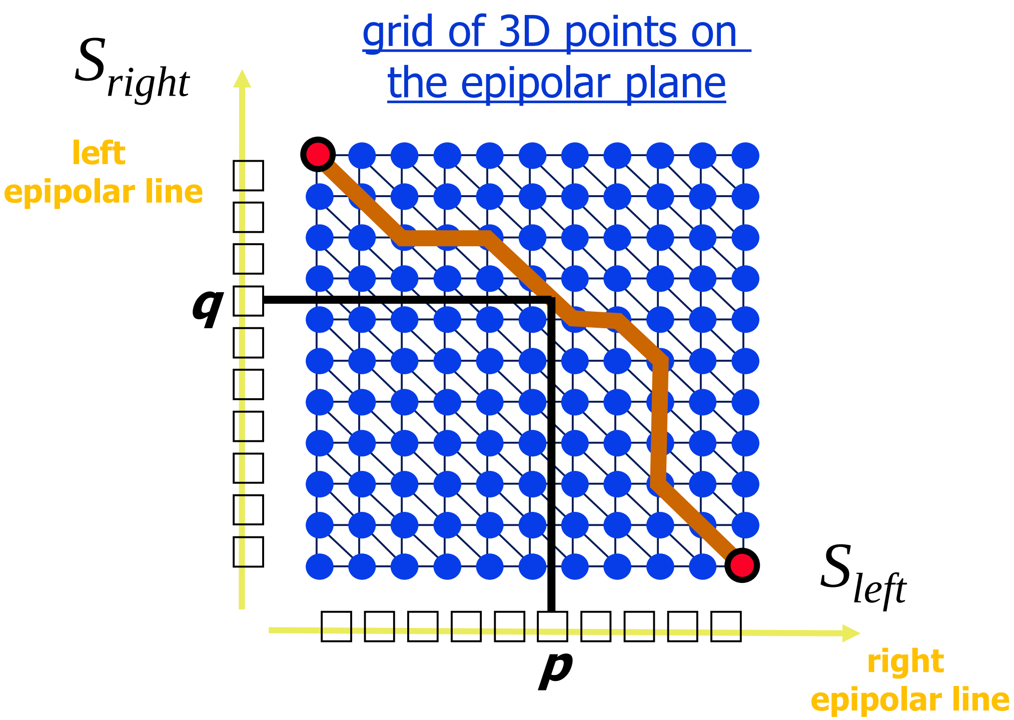




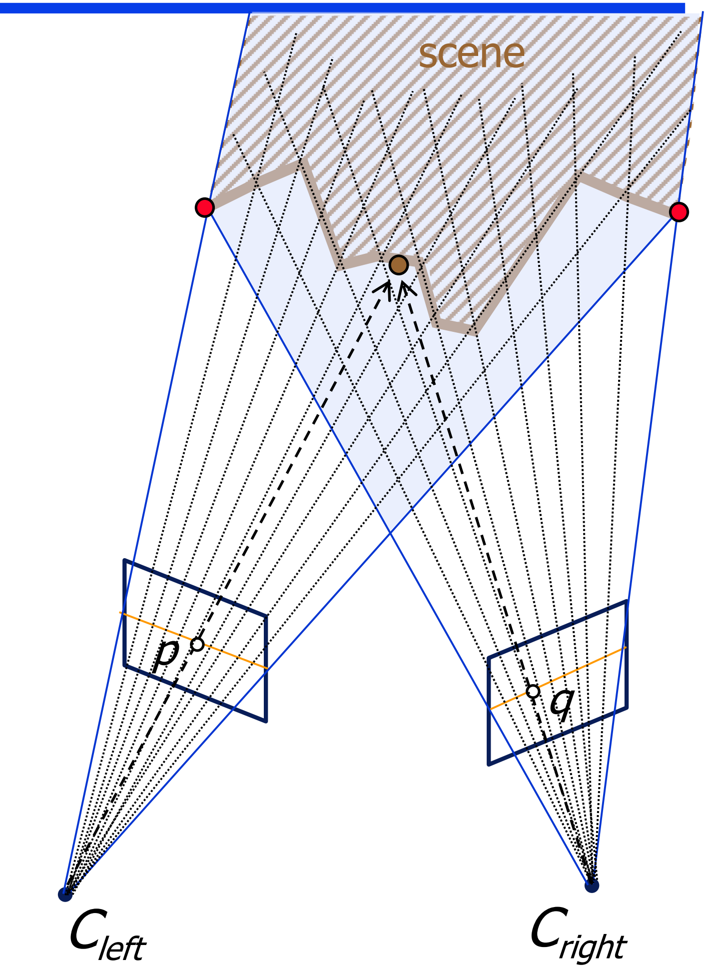
a **path** on this graph represents a matching function

“Shortest paths” for Scan-line stereo

3D interpretation:



a **path** on this graph represents a matching function

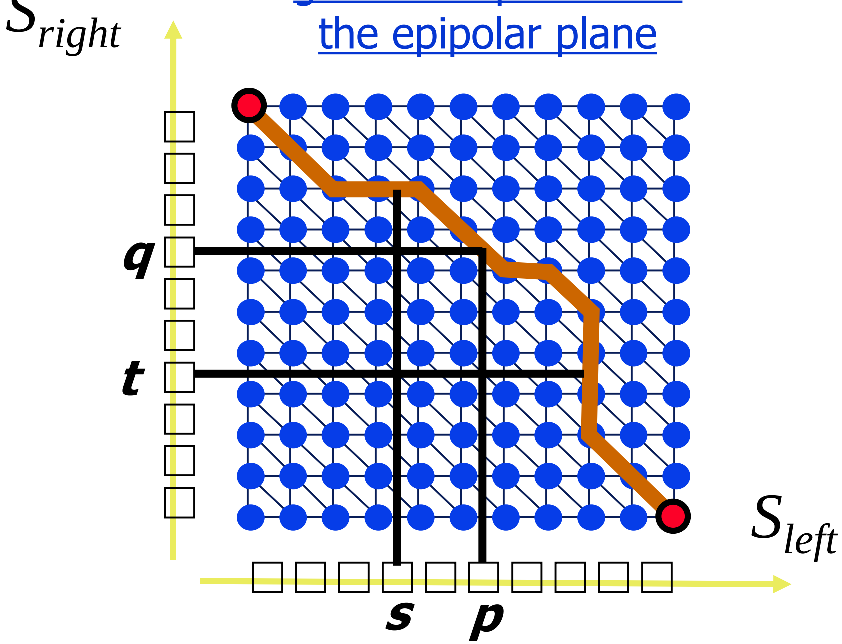


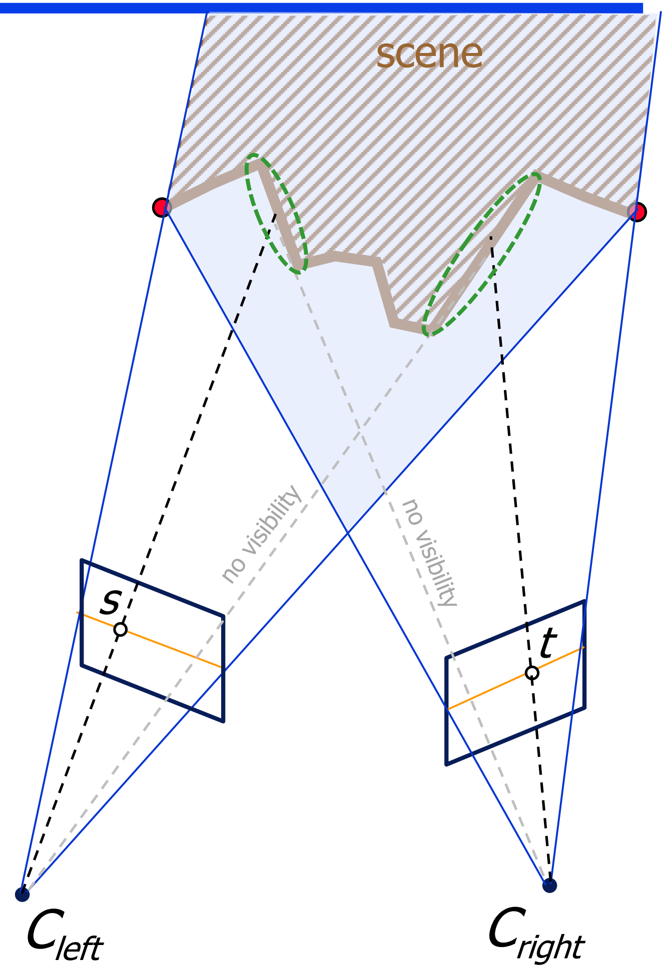
This path corresponds to an intersection of epipolar plane with 3D scene surface

"Shortest paths" for Scan-line stereo

3D interpretation:

[grid of 3D points on the epipolar plane]()

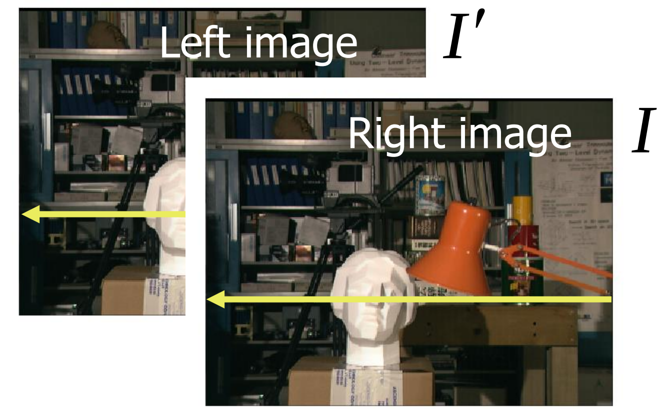


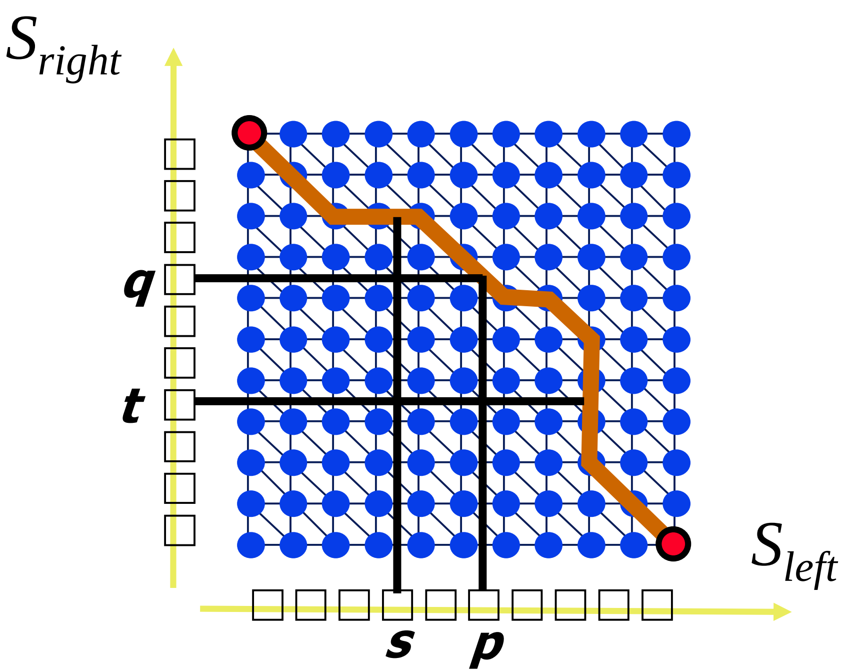


horizontal and vertical edges on the path imply "no correspondence" (*occlusion*)

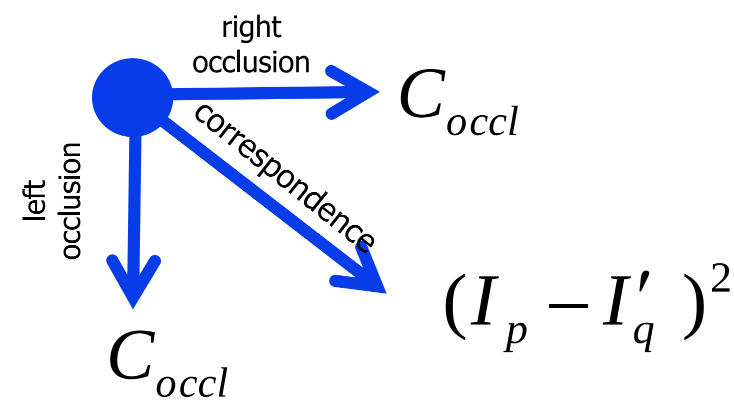
“Shortest paths” for Scan-line stereo

e.g. Ohta&Kanade’85, Cox at.al.’96



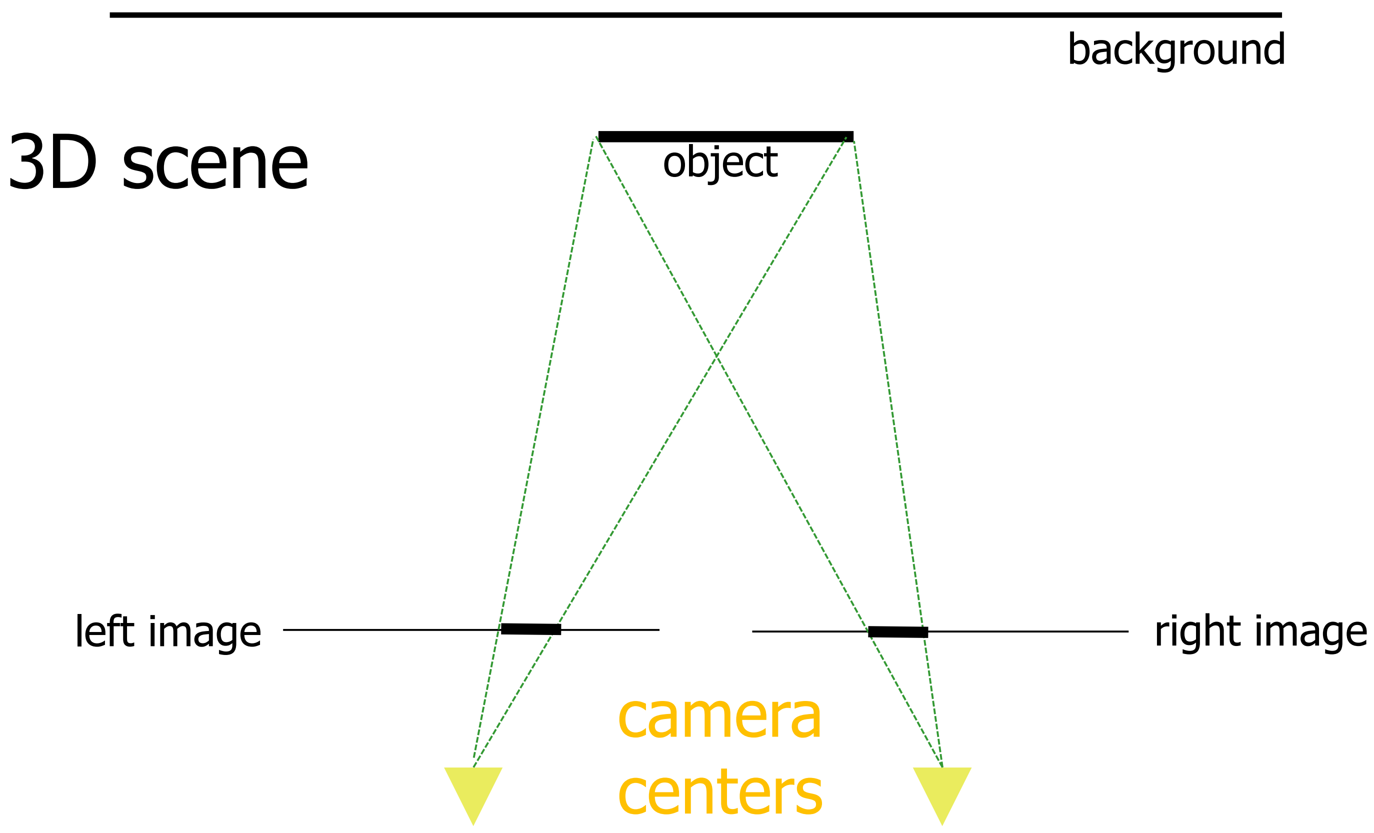


Edge weights:

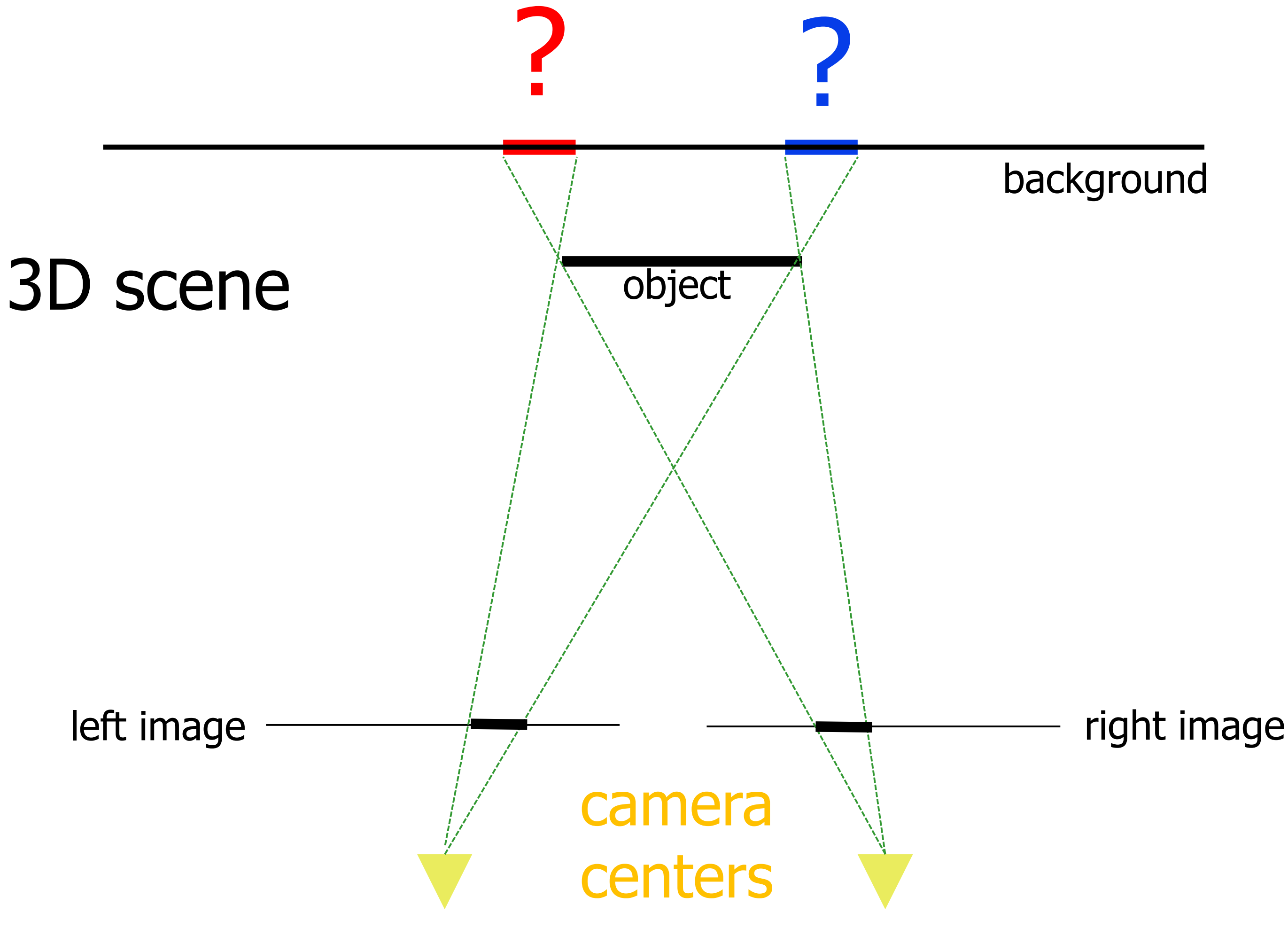


What is “occlusion” in general ?

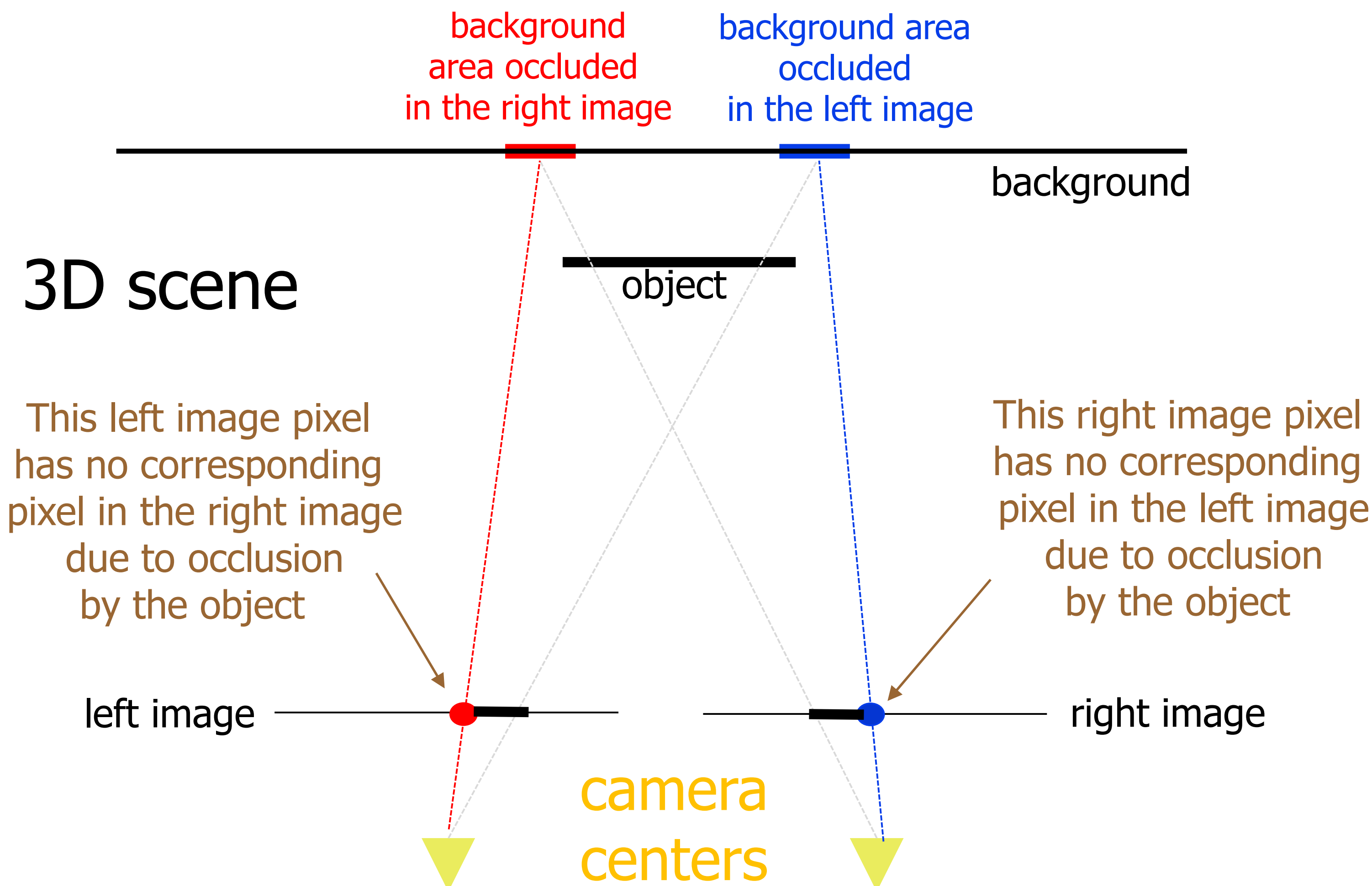
Occlusion in stereo



Occlusion in stereo

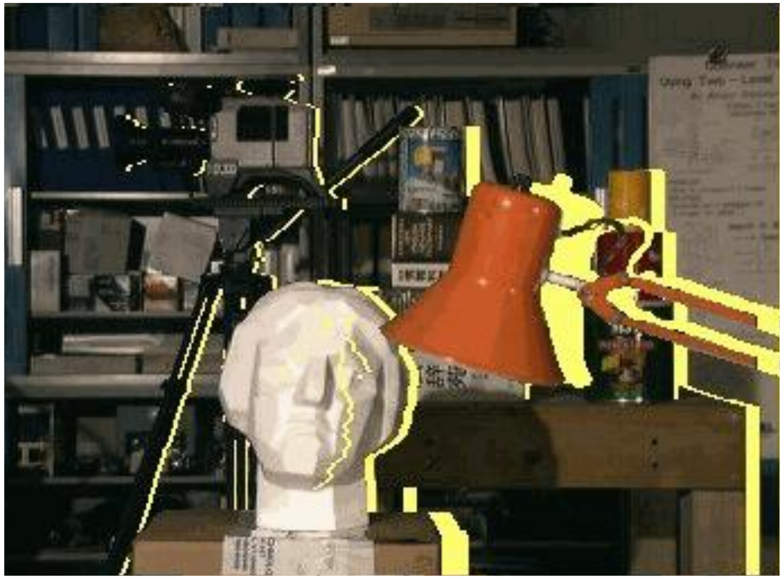


Occlusion in stereo



Note: **occlusions occur at depth discontinuities/jumps**

Stereo

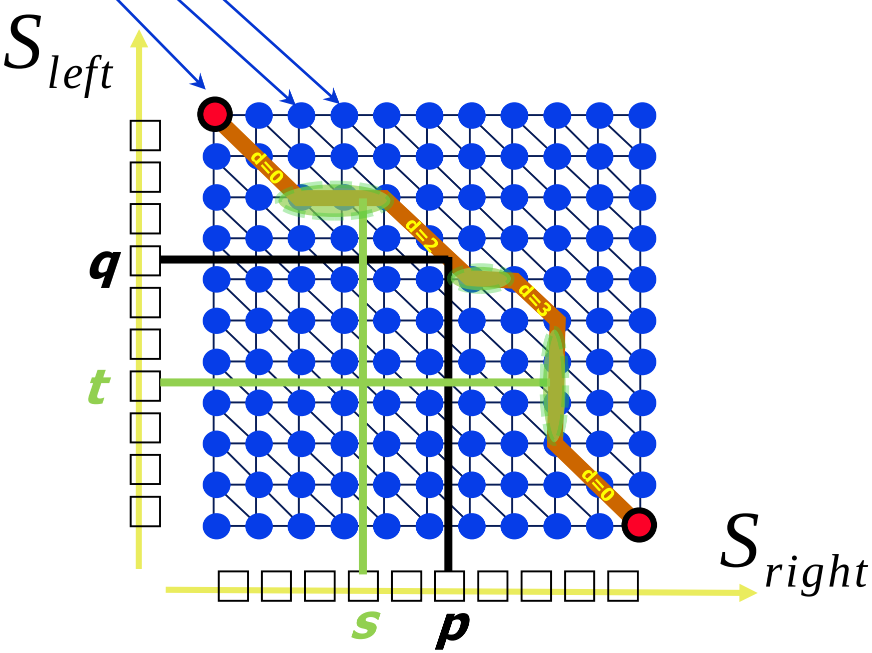


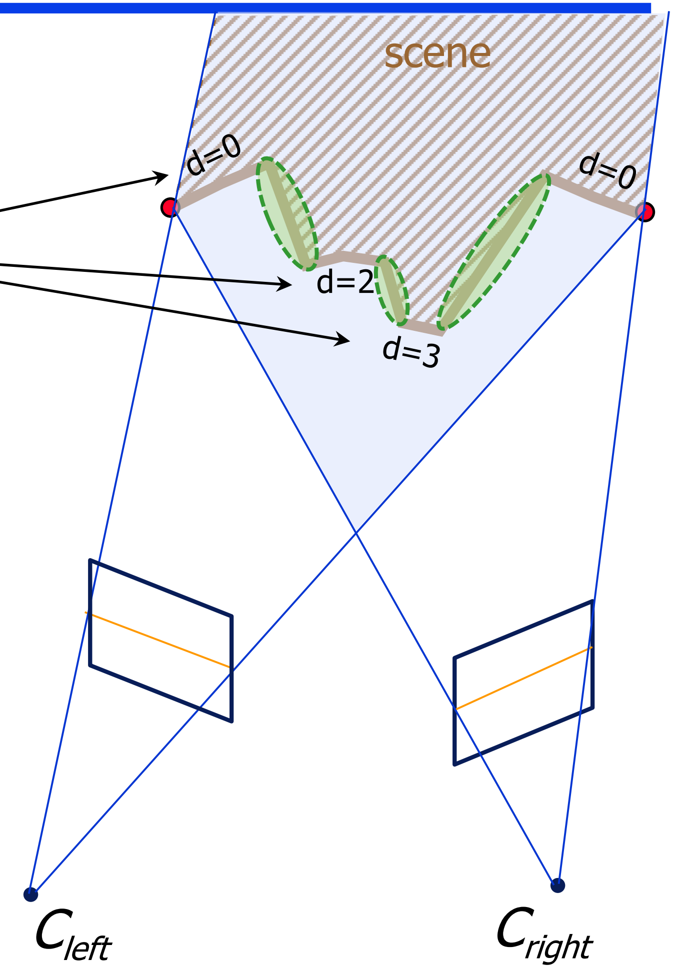
yellow marks occluded points in different viewpoints
(points not visible from the central/base viewpoint).

Note: **occlusions occur at depth discontinuities/jumps**

Occlusions vs disparity/depth jumps

NOTE: diagonal lines on this graph represent *disparity levels* (shifts between corresponding pixels) that can be seen as *depth layers*

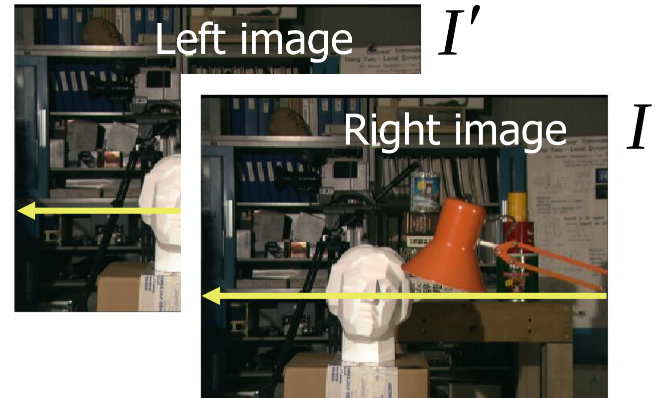


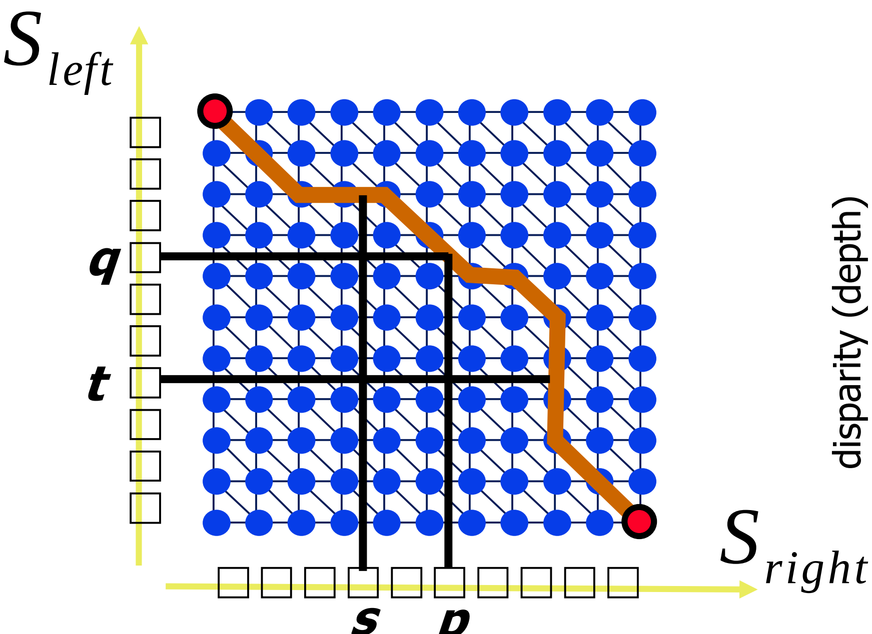


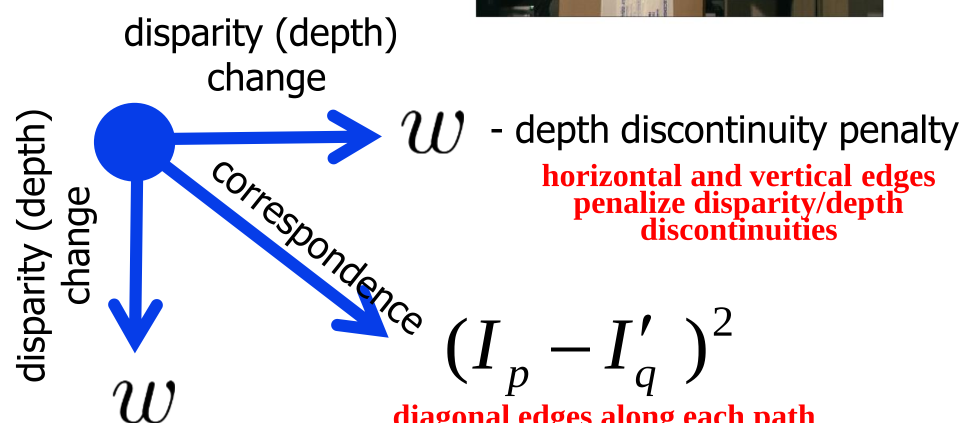
horizontal and vertical edges on this graph describe occlusions, as well as disparity jumps or depth discontinuities

Use Dijkstra to find *the shortest path* corresponding to certain edge costs

e.g. Ohta&Kanade'85, Cox at.al.'96







horizontal and vertical edges penalize disparity/depth discontinuities

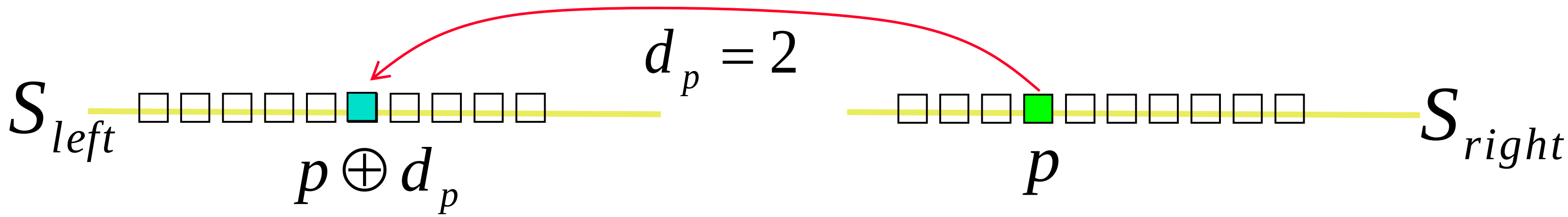
diagonal edges along each path integrate SSD between corresponding pixels

Each path implies certain depth/disparity configuration. Dijkstra can find the best one.

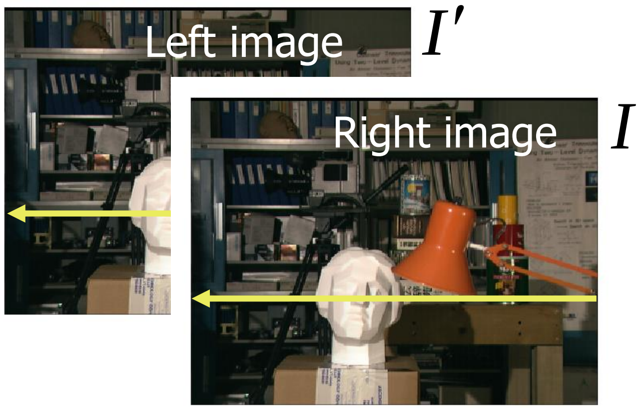
But, the actual implementation in OK'85 and C'96 uses *Viterbi* algorithm (DP) explicitly assigning "optimal" disparity labels d_p to all pixels p as follows…

UNIVERSITY OF WATERLOO

DP for scan-line stereo



Viterbi algorithm can be used to optimize the following energy of *disparities* $\mathbf{d} = \{d_p \mid p \in S\}$ of pixels p on a fixed scan-line S_{right}



$$E(\mathbf{d}) = \sum_{p \in S} \underbrace{D_p(d_p)}_{= \; |I_p - I'_{p \oplus d_p}|} + \sum_{p \in S} \underbrace{V(d_p, d_{p+1})}_{= \; w\,|d_p - d_{p+1}|}$$

photo consistency

spatial coherence

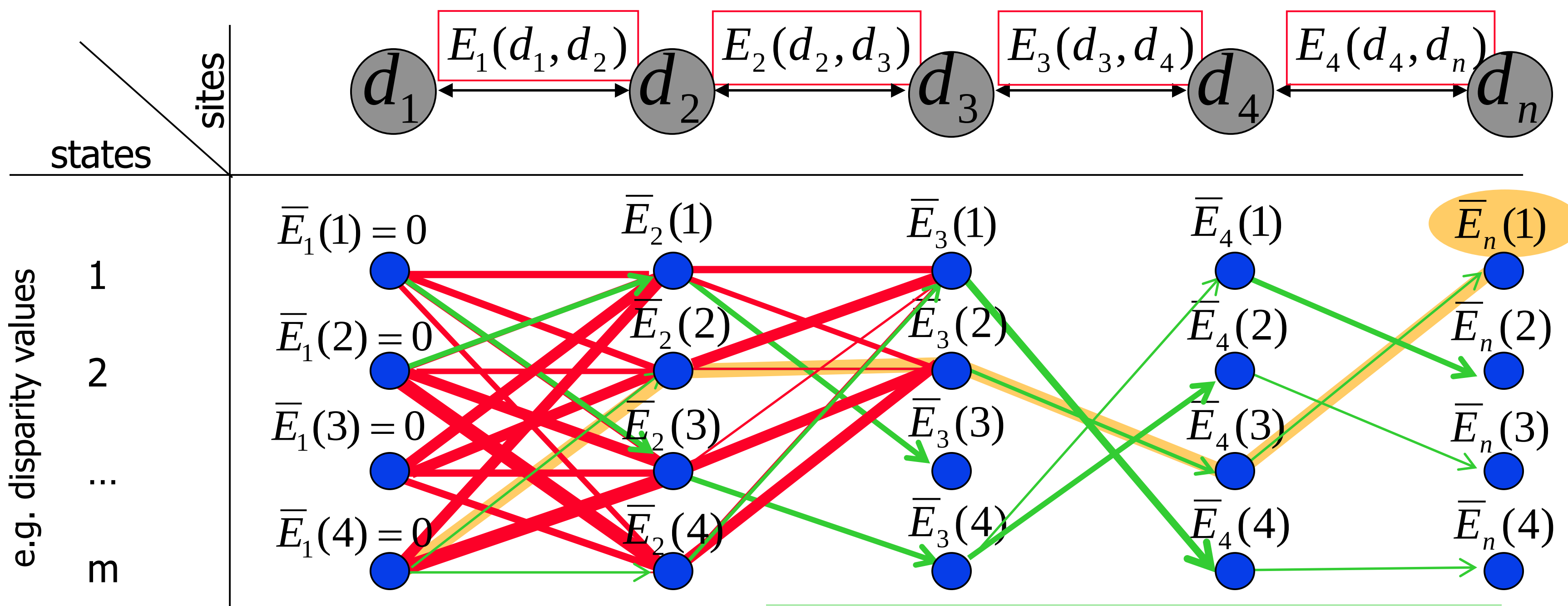
$$= \sum_{\{p,q\} \in N} E(d_p, d_q)$$

Viterbi can handle this on non-loopy graphs (e.g., scan-lines)

Dynamic Programming (DP)
Viterbi Algorithm

Consider **pair-wise interactions** between sites (pixels) on a **chain** (scan-line)

$$E_1(d_1,d_2)+E_2(d_2,d_3)+\ldots+E_{n-1}(d_{n-1},d_n)$$



$\bar{E}_p(k)$ - internal "energy counter" at "site" p and "state" k

$$\bar{E}_{p+1}(k)=\min_i\left(\bar{E}_p(i)+E_p(i,k)\right)$$

Complexity: $O(nm^2)$, worst case = best case

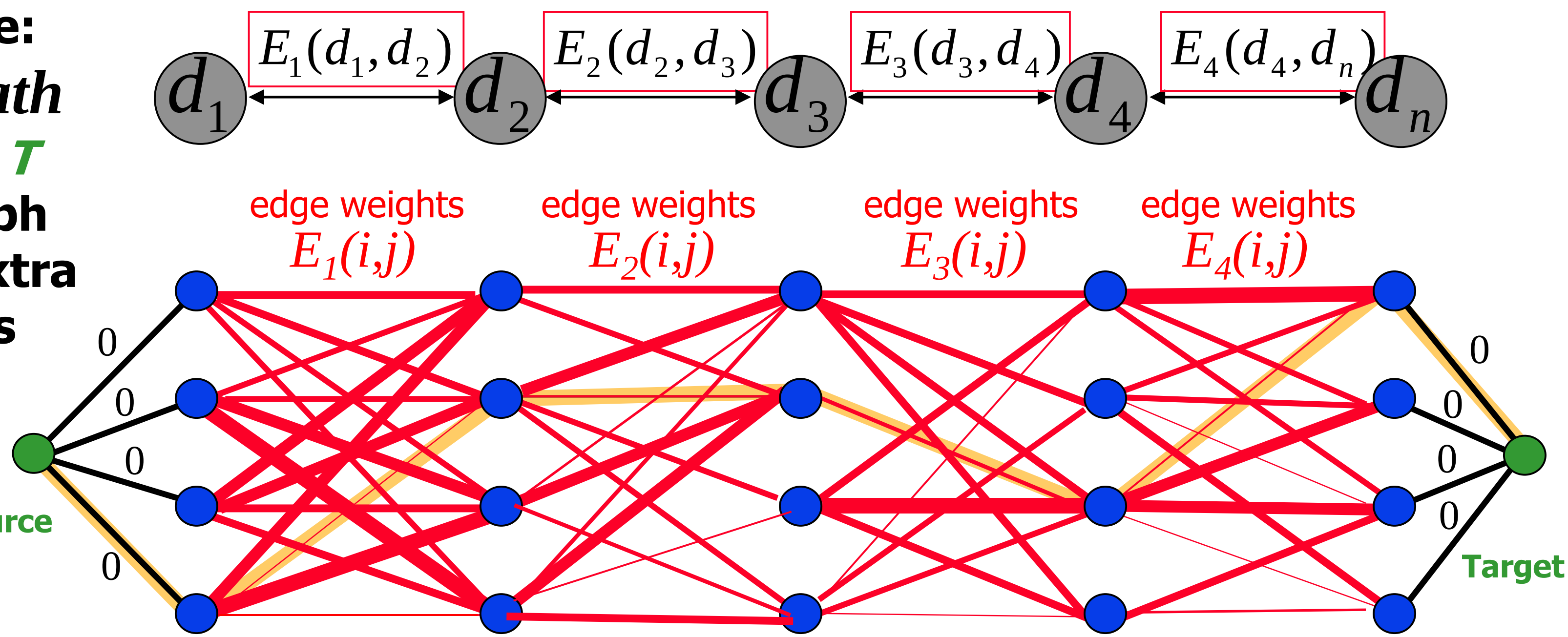
Q: how does this relate to the "shortest path" algorithm (Dijkstra)?

Dynamic Programming (DP)
Shortest paths Algorithm

Consider **pair-wise interactions** between sites (pixels) on a **chain** (scan-line)

$$E_1(d_1,d_2)+E_2(d_2,d_3)+\ldots+E_{n-1}(d_{n-1},d_n)$$

Alternative:
shortest path
from *S* to *T*
on the graph
with two extra
terminals



Complexity: $O(nm^2+nm\log(nm))$ - worst case

But, the best case could be better than Viterbi. Why?

Coherent disparity map on 2D grid?

- Scan-line stereo generates streaking artifacts
- Can't use Viterbi or Dijkstra to find globally optimal solutions on loopy graphs (e.g. grids) ☹

  (Note: there exist their extensions, e.g. *belief propagation*, *TRWS*, etc)

- Regularization problems in vision is an interesting domain for optimization algorithms

  (Note: it is known that *gradient descent* does not work well for such problems)

Example: **graph cut** algorithms can find globally optimal solutions for certain energies/losses on arbitrary (loopy) graphs

Estimating (optimizing) disparities: over **points** vs. **scan-lines** vs. **grid**

Consider energy (loss) function over disparities

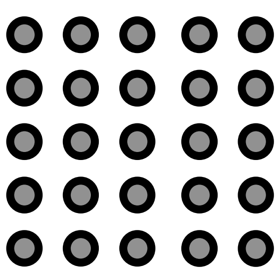
$\mathbf{d} = \{d_p \mid p \in G\}$ for pixels p on grid G

$$E(\mathbf{d}) = \sum_{p\in G} \underbrace{D_p(d_p)}_{=\,|I_p - I'_{p\oplus d_p}|} \quad + \sum_{\{p,q\}\in N} \underbrace{V(d_p,d_q)}_{=\,w\,|d_p - d_q|}$$

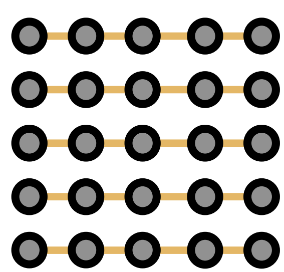
photo consistency

spatial coherence

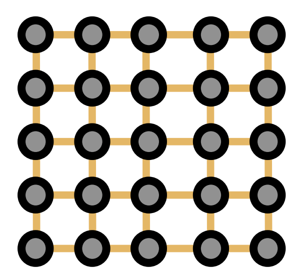
Consider three different neighborhood systems N:



$N = \varnothing$



$N = \{\{p,p\pm 1\} : p \in G\}$



$N = \{\{p,q\} \subset G : |pq| \leq 1\}$

Estimating (optimizing) disparities: over **points** vs. **scan-lines** vs. **grid**

Consider energy (loss) function over disparities

$\mathbf{d} = \{d_p \mid p \in G\}$ for pixels p on grid G

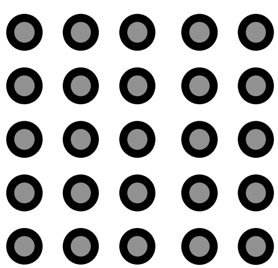
$$E(\mathbf{d}) = \sum_{p \in G} \underbrace{D_p(d_p)}_{=\, |I_p - I'_{p \oplus d_p}|} + \cancel{\sum_{\{p,q\} \in N} \underbrace{V(d_p, d_q)}_{=\, w\,|d_p - d_q|}}$$

photo consistency

spatial coherence

smoothness term disappears

CASE 1



$N = \varnothing$

Q: how to optimize $E(\mathbf{d})$ in this case?

$$\forall p \in G \quad \boxed{\hat{d}_p = \arg\min_d D_p(d)} \quad O(nm)$$

Q: How does this relate to window-based stereo?

Estimating (optimizing) disparities: over **points** vs. **scan-lines** vs. **grid**

Consider energy (loss) function over disparities
$\mathbf{d} = \{d_p \mid p \in G\}$ for pixels p on grid G

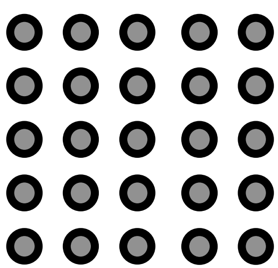
$$E(\mathbf{d}) = \sum_{p\in G} \underbrace{D_p(d_p)}_{=\,|I_p - I'_{p\oplus d_p}|} + \cancel{\sum_{\{p,q\}\in N} \underbrace{V(d_p,d_q)}_{=\,w|d_p - d_q|}}$$

photo consistency

~~spatial coherence~~

smoothness term disappears

CASE 1



$N = \varnothing$

Nodes/pixels do not interact (are independent).
Optimization of the sum of **unary terms**,
e.g. $\sum_{p\in G} D_p(d_p)$, is trivial: $O(nm)$

Estimating (optimizing) disparities: over **points** vs. **scan-lines** vs. **grid**

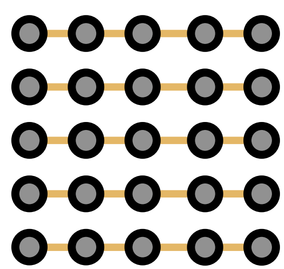
Consider energy (loss) function over disparities $\mathbf{d} = \{d_p \mid p \in G\}$ for pixels p on grid G

$$E(\mathbf{d}) = \sum_{p \in G} \underbrace{D_p(d_p)}_{= \, |I_p - I'_{p \oplus d_p}|} + \sum_{\{p,q\} \in N} \underbrace{V(d_p, d_q)}_{= \, w\,|d_p - d_q|}$$

photo consistency

spatial coherence

CASE 2



$N = \{\{p, p \pm 1\} : p \in G\}$

Pairwise coherence is enforced, but only between pixels on the same scan line.

Q: how do we optimize $E(\mathbf{d})$ now?

$O(nm^2)$

Estimating (optimizing) disparities: over **points** vs. **scan-lines** vs. **grid**

Consider energy (loss) function over disparities
$\mathbf{d} = \{d_p \mid p \in G\}$ for pixels p on grid G

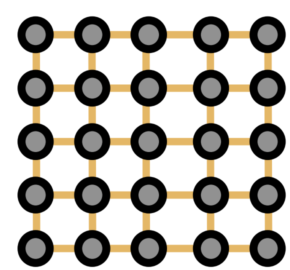
$$E(\mathbf{d}) = \sum_{p \in G} \underbrace{D_p(d_p)}_{= \; |I_p - I'_{p \oplus d_p}|} + \sum_{\{p,q\} \in N} \underbrace{V(d_p, d_q)}_{= \; w\,|d_p - d_q|}$$

photo consistency

spatial coherence

CASE 3

Pairwise smoothness of the disparity map is enforced both horizontally and vertically.



NOTE: *depth map* coherence should be isotropic as it describes 3D scene surface independent of scan-lines (epiplar lines) orientation.

$N = \{\{p,q\} \subset G : |pq| \le 1\}$

Estimating (optimizing) disparities: over **points** vs. **scan-lines** vs. **grid**

Consider energy (loss) function over disparities $\mathbf{d} = \{d_p \mid p \in G\}$ for pixels p on grid G

$$E(\mathbf{d}) = \sum_{p\in G} \underbrace{D_p(d_p)}_{= \, |I_p - I'_{p\oplus d_p}|} + \boxed{\sum_{\{p,q\}\in N} \underbrace{V(d_p, d_q)}_{= \, w\,|d_p - d_q|}}$$

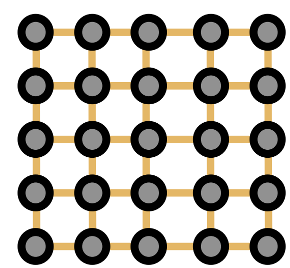
photo consistency — spatial coherence

How to optimize "pairwise" loss on loopy graphs?

NOTE 1: **Viterbi does not apply**, but its extensions (e.g. *message passing*) provide approximate solutions on loopy graphs.

NOTE 2: "*Gradient descent*" can find only local minima for a continuous relaxation of $E(\mathbf{d})$ combining <u>non-convex</u> photo-consistency (1st term) and convex *total variation* of $\mathbf{d}$ (2nd term).

CASE 3



$$N = \{\{p,q\} \subset G : |pq| \le 1\}$$

Graph cut
for spatially coherent stereo on 2D grids

One can **globally minimize** the following energy of disparities $\mathbf{d} = \{d_p \mid p \in G\}$ for pixels p on grid G

$$E(\mathbf{d}) = \sum_{p \in G} \underbrace{D_p(d_p)}_{=\ \boxed{|I_p - I'_{p \oplus d_p}|}} + \sum_{\{p,q\} \in N} \underbrace{V(d_p, d_q)}_{=\ \boxed{w_{pq} \cdot |d_p - d_q|}}$$

photo consistency

spatial coherence

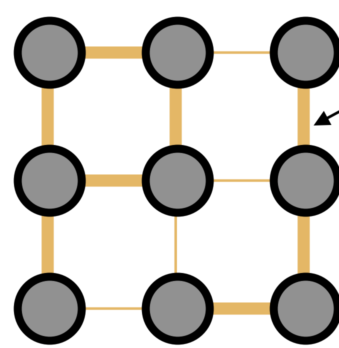
Unlike shortest paths or Viterbi, standard s/t **graph cut algorithms** can globally minimize certain types of pairwise energies **on loopy graphs**.

Graph cut
for spatially coherent stereo on 2D grids

One can **globally minimize** the following energy of disparities $\mathbf{d} = \{d_p \mid p \in G\}$ for pixels p on grid G

$$E(\mathbf{d}) = \sum_{p\in G} \underbrace{D_p(d_p)}_{=\;\boxed{|I_p - I'_{p\oplus d_p}|}} \quad + \sum_{\{p,q\}\in N} \underbrace{V(d_p, d_q)}_{=\;\boxed{w_{pq}\cdot|d_p - d_q|}}$$

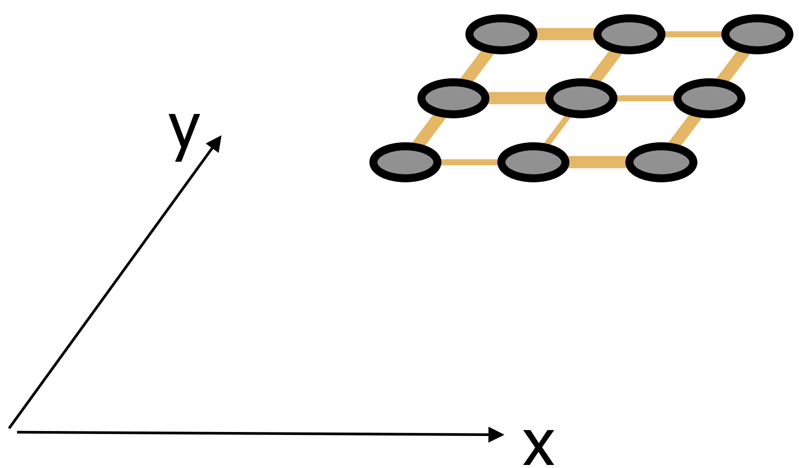
photo consistency — spatial coherence



w_{pq} - weights of neighborhood edges

(e.g. may be assigned according to local intensity contrast)

Multi-scan-line stereo with *s-t* graph cuts [Roy&Cox'98, Ishikawa 98]

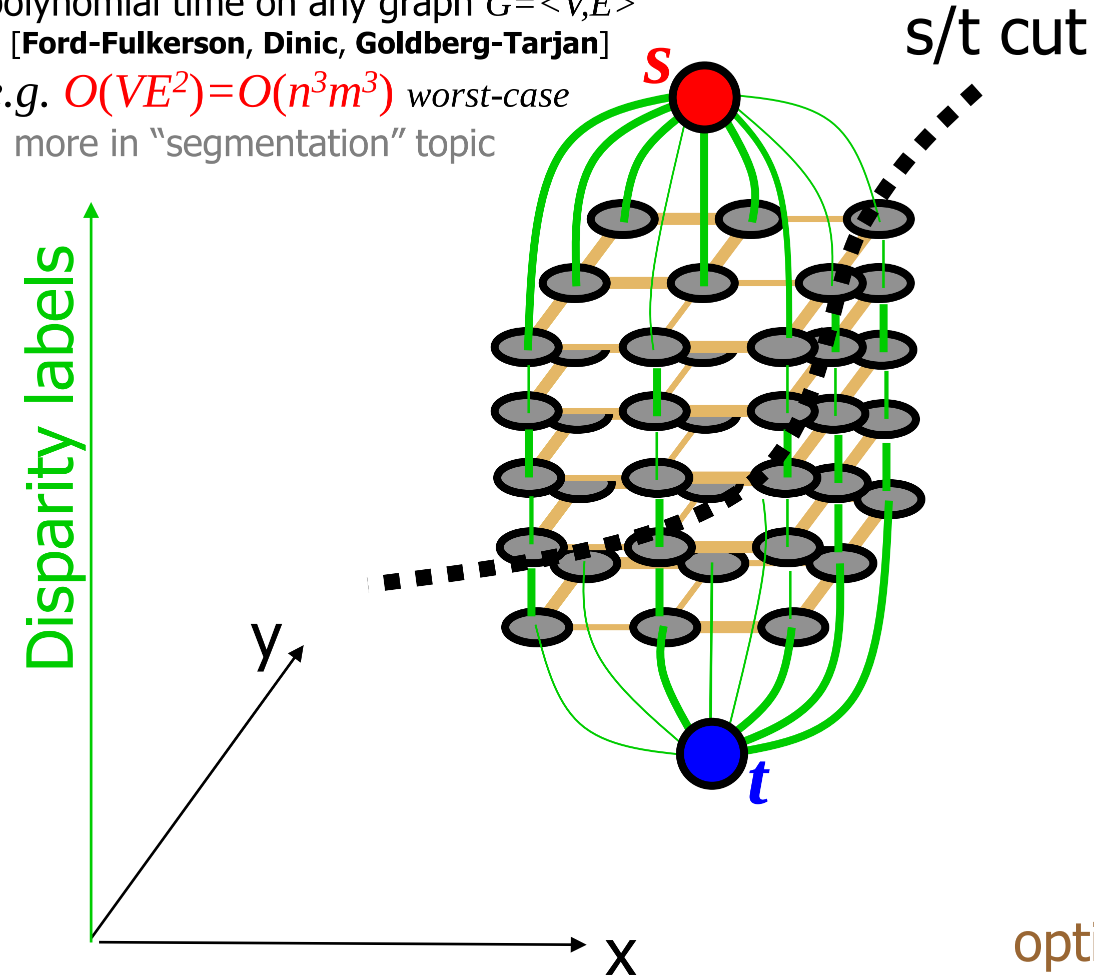


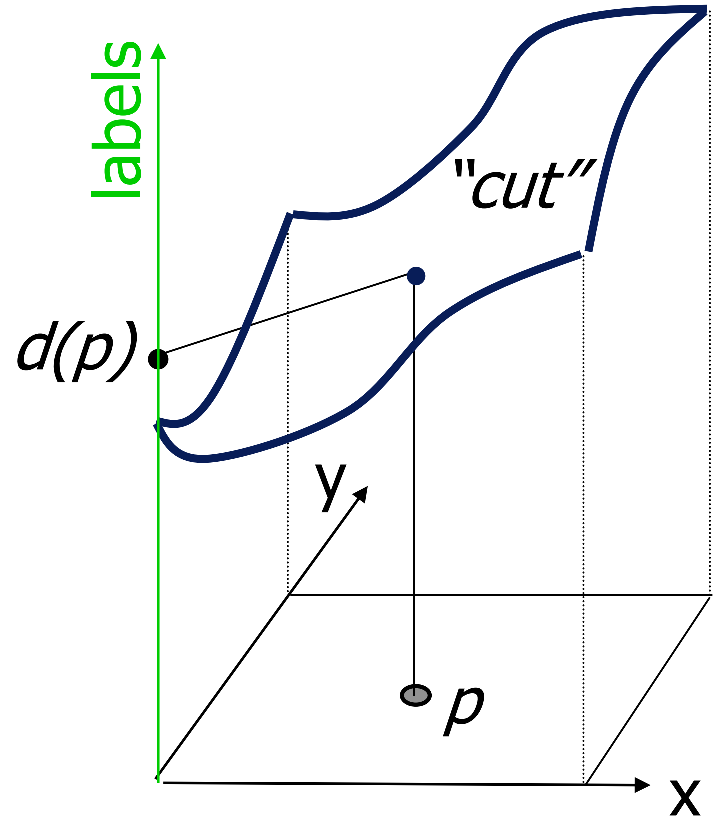
Multi-scan-line stereo with *s*-*t* graph cuts [Roy&Cox'98, Ishikawa 98]

Minimum s/t cuts can be found in low-order polynomial time on any graph $G=<V,E>$
[**Ford-Fulkerson**, **Dinic**, **Goldberg-Tarjan**]

e.g. $O(VE^2)=O(n^3m^3)$ *worst-case*

more in "segmentation" topic





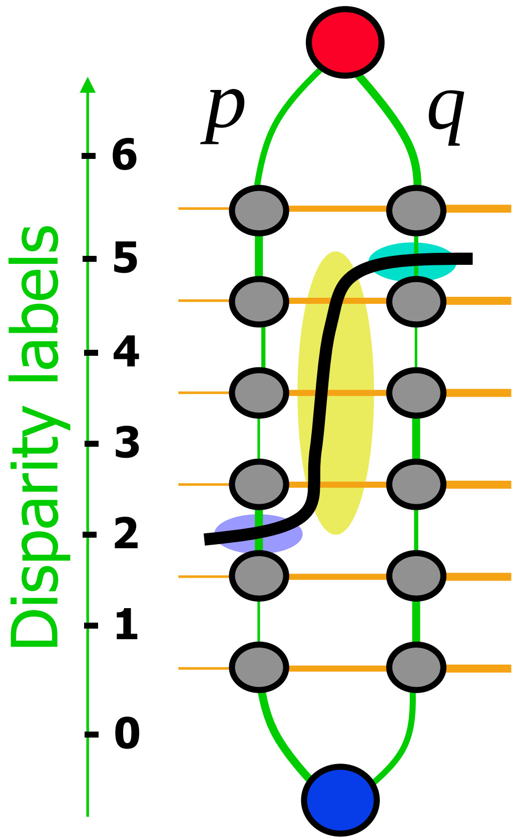
minimum cut will define optimal disparity map $\mathbf{d} = \{d_p\}$

assume that a cut has no folds (later slide will show how to make sure)

Ishikawa&Geiger 98

What energy do we minimize this way?

Concentrate on one pair of neighboring pixels $\{p,q\} \in N$



cost of vertical edges

$$E(d_p, d_q) = D_p(2) + D_q(5) + \ldots$$

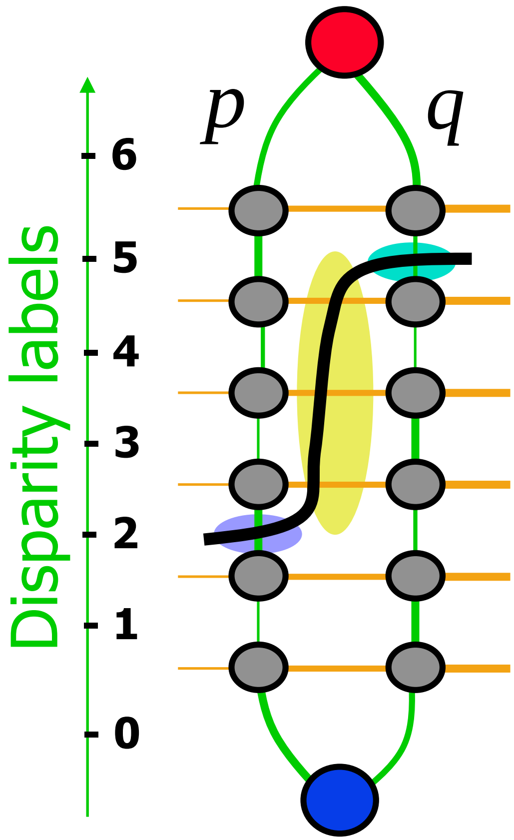
$$+ \quad w_{pq} \cdot |3| \quad + \ldots$$

cost of horizontal edges

Ishikawa&Geiger 98

What energy do we minimize this way?

Concentrate on one pair of neighboring pixels $\{p,q\} \in N$



cost of vertical edges

$$E(d_p, d_q) = D_p(d_p) + D_q(d_q) + \ldots$$

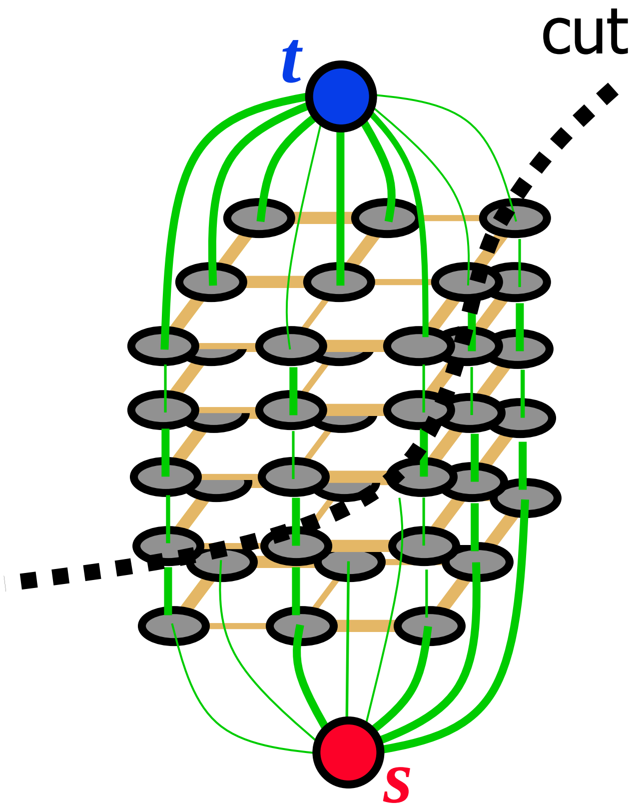
$$+ \quad w_{pq} \cdot |d_p - d_q| + \ldots$$

cost of horizontal edges

Ishikawa&Geiger 98

What energy do we minimize this way?

The combined energy over the entire grid G is



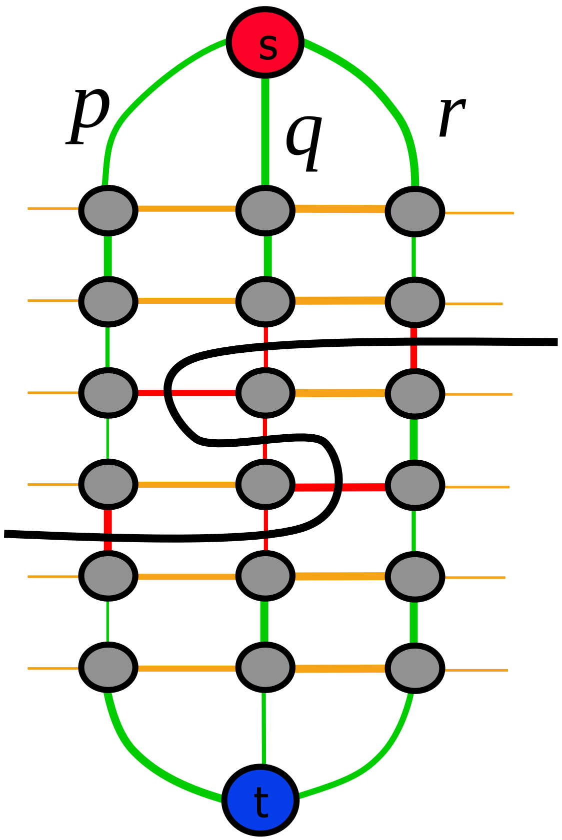
(**photo consistency**, e.g. SSD)
cost of vertical edges

$$E(\mathbf{d}) = \sum_{p \in G} D_p(d_p) + \sum_{\{p,q\} \in N} w_{pq} \cdot | d_p - d_q |$$

cost of horizontal edges
(**spatial consistency**)

How to avoid folding?

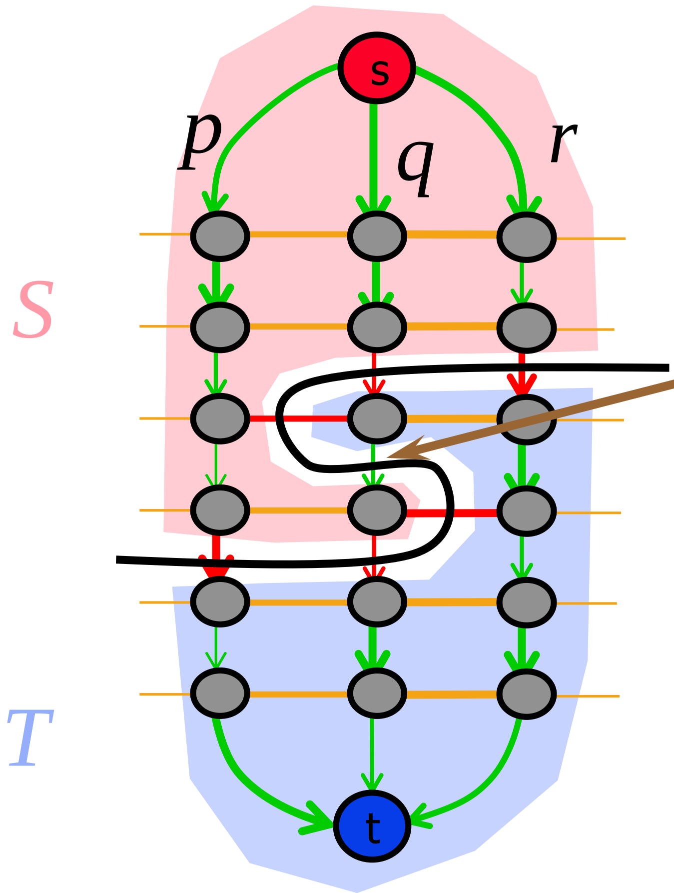
consider three pixels $\{p, q, r\}$



"severed" edges are shown in **red**

How to avoid folding?

consider three pixels $\{p,q,r\}$



introduce directed *t-links*

NOTE: this directed *t-link* is not "severed"

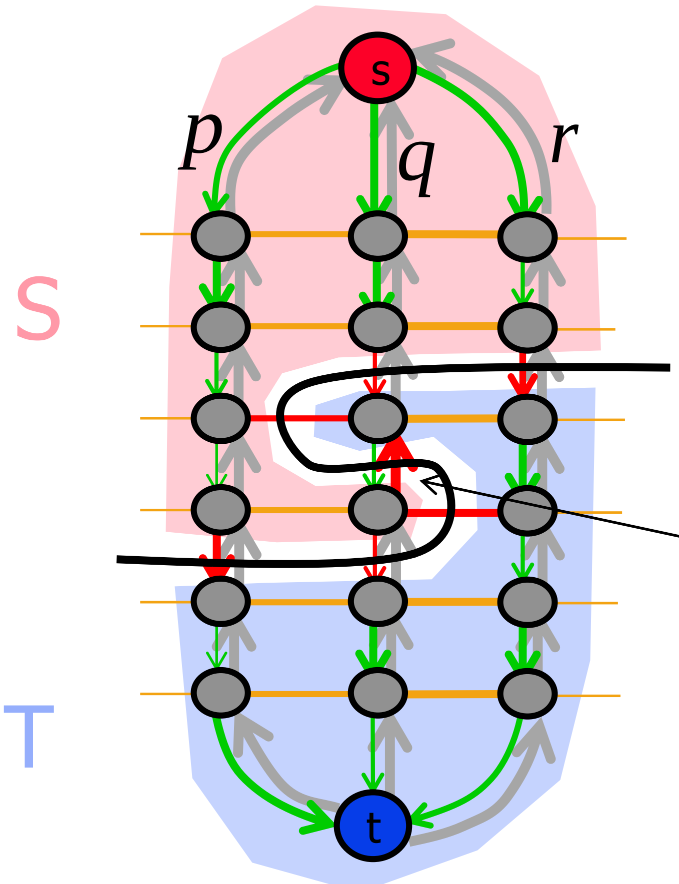
WHY?

Formally, s/t cut is a partitioning of graph nodes $C=\{S,T\}$ and its cost is $\|C\| = \sum_{\substack{(pq)\in N \\ p\in S \\ q\in T}} c_{pq}$

only edges from S to T matter

How to avoid folding?

consider three pixels $\{p, q, r\}$



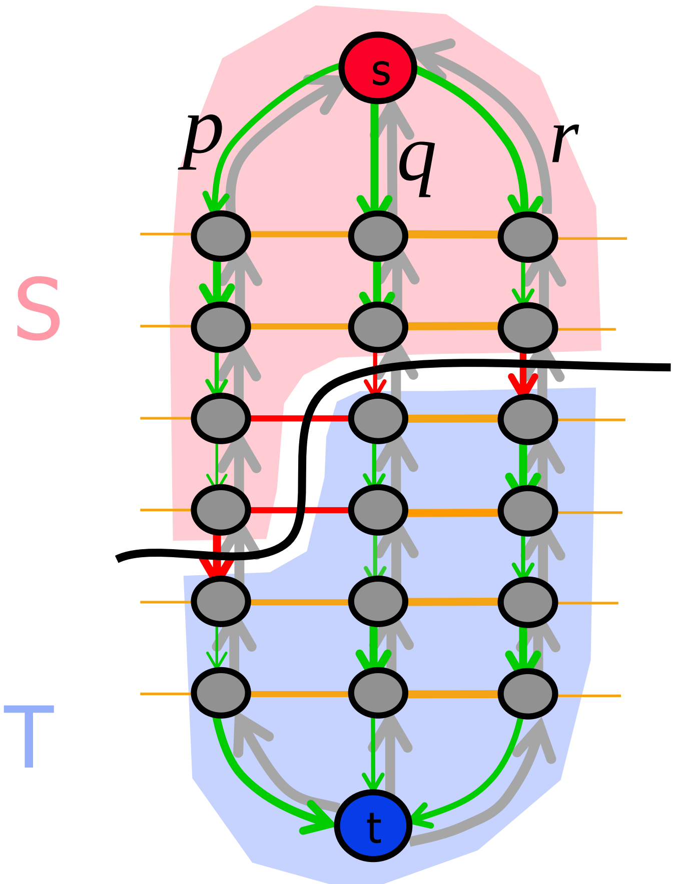
Solution prohibiting **folds**:

add <u>infinity</u> cost t-links in the "up" direction

NOTE: **folding cuts** $C = \{S, T\}$ sever at least one of such t-links making such cuts **infeasible**

How to avoid folding?

consider three pixels $\{p, q, r\}$

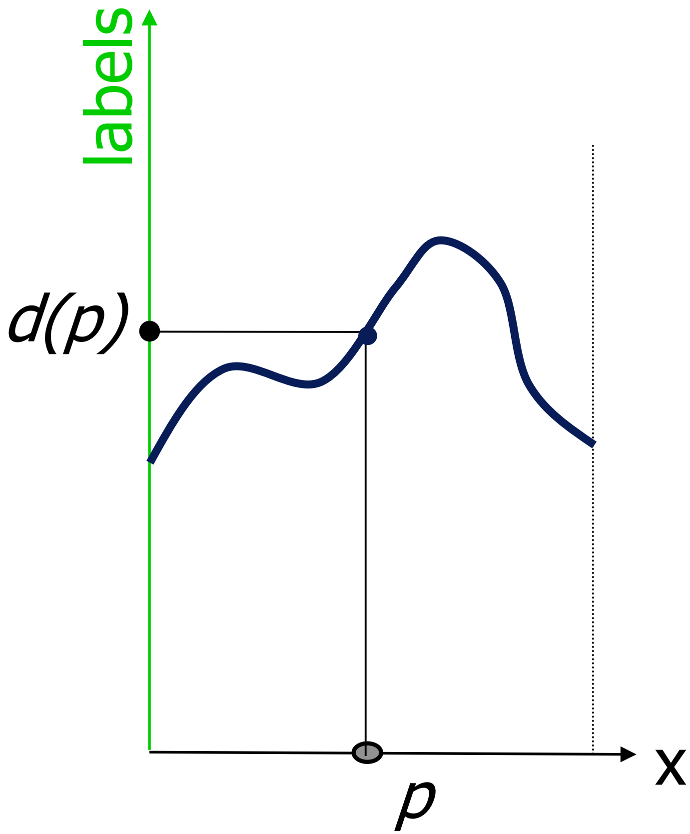


Solution prohibiting **folds**:

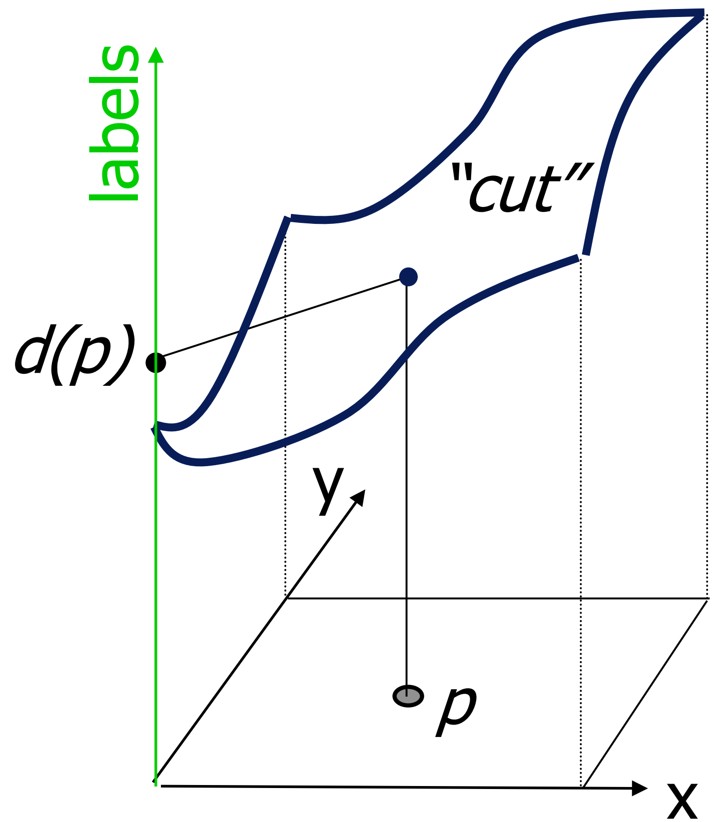
add <u>infinity</u> cost t-links in the "up" direction

NOTE: **non-folding cuts** $C = \{S, T\}$ do not sever such t-links

Scan-line stereo vs. Multi-scan-line stereo (on whole grid)

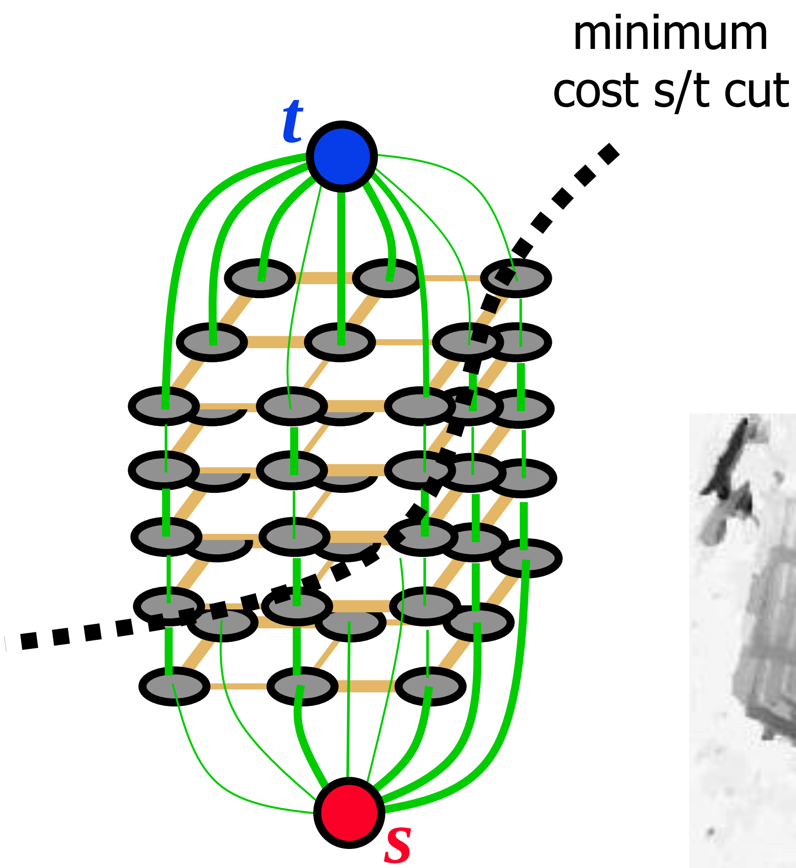


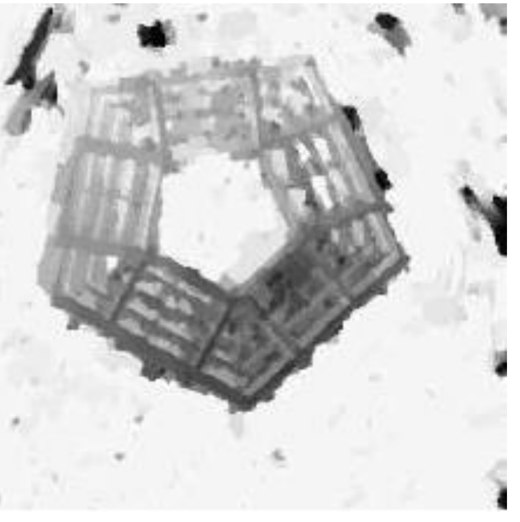
Dynamic Programming
(single scan line optimization)



s-t Graph Cuts
(grid optimization)

Some results from Roy&Cox

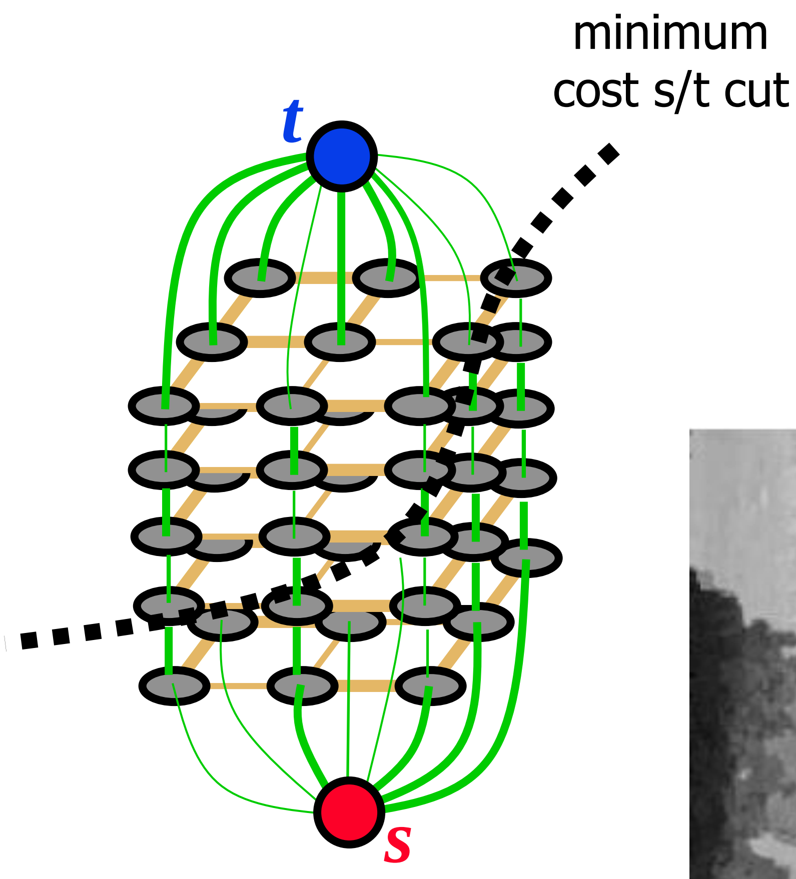




multi scan line stereo
(graph cuts)

single scan-line stereo
(DP)

Some results from Roy&Cox



multi scan line stereo
(graph cuts)

single scan-line stereo
(DP)

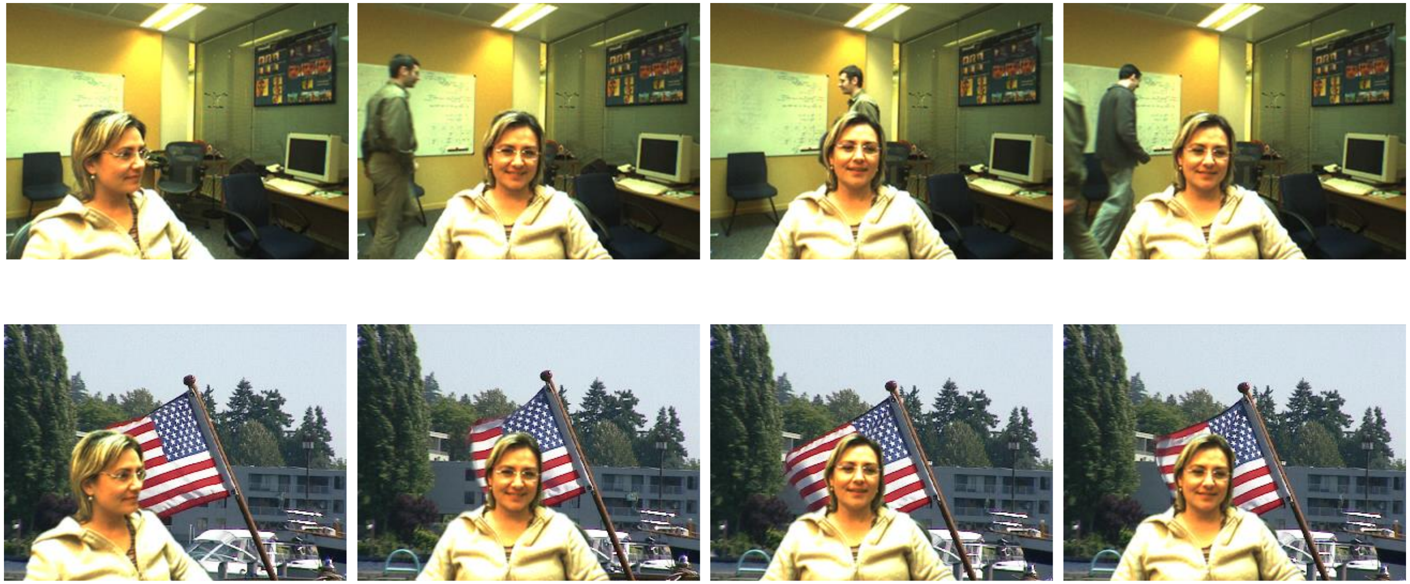
Simple Examples:
Stereo with only 2 depth layers

binary stereo

essentially,
depth-based
segmentation

[Kolmogorov et al. CVPR 2005, IJCV 2008]

Simple Examples:
Stereo with only 2 depth layers



background substitution

essentially,
depth-based
segmentation

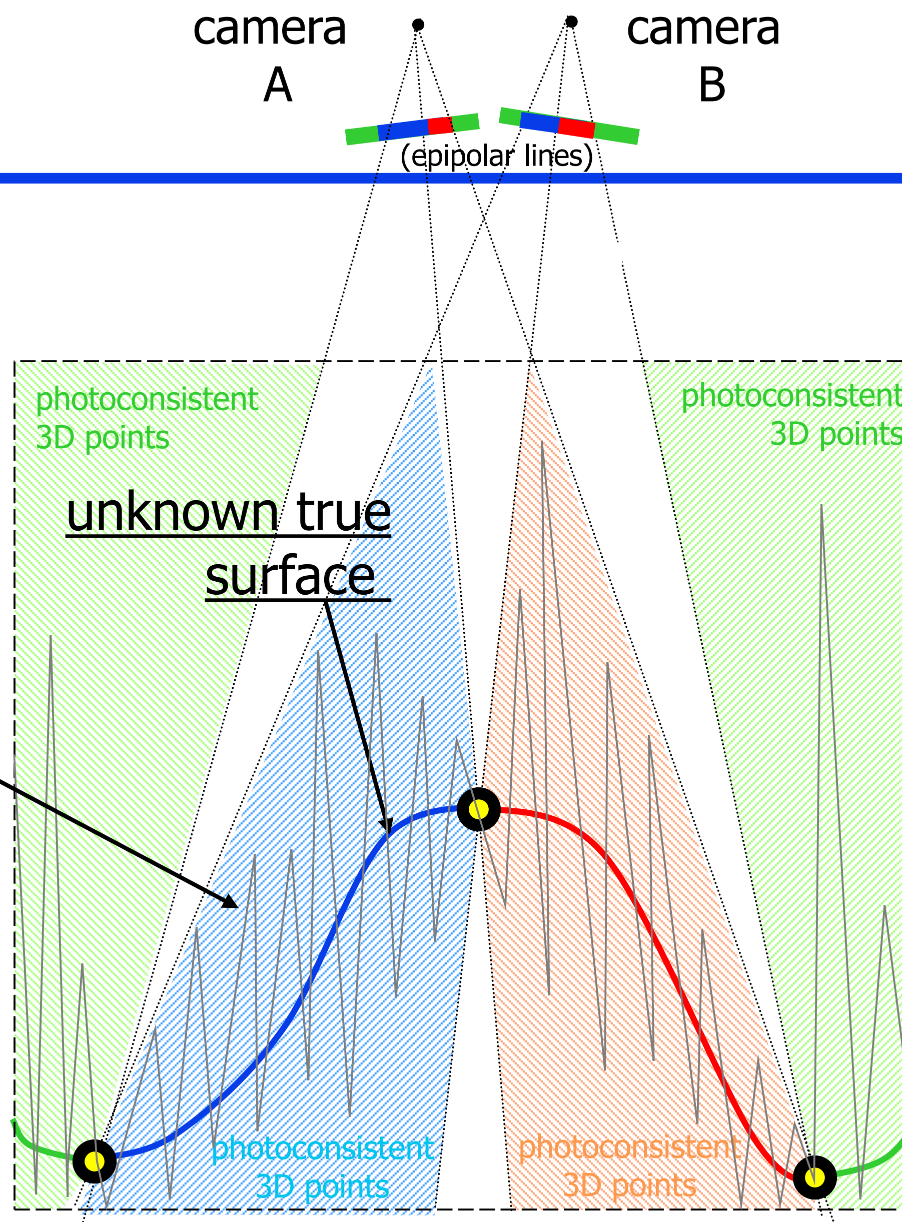
[Kolmogorov et al. CVPR 2005, IJCV 2008]

Features and Regularization

photo-consistency term

$$E(\mathbf{d}) \;=\; \sum_{p\in G} |I_p - I'_{p+d_p}|$$

photoconsistent depth map



3D volume where surface is being reconstructed (epipolar plane)

Features and Regularization

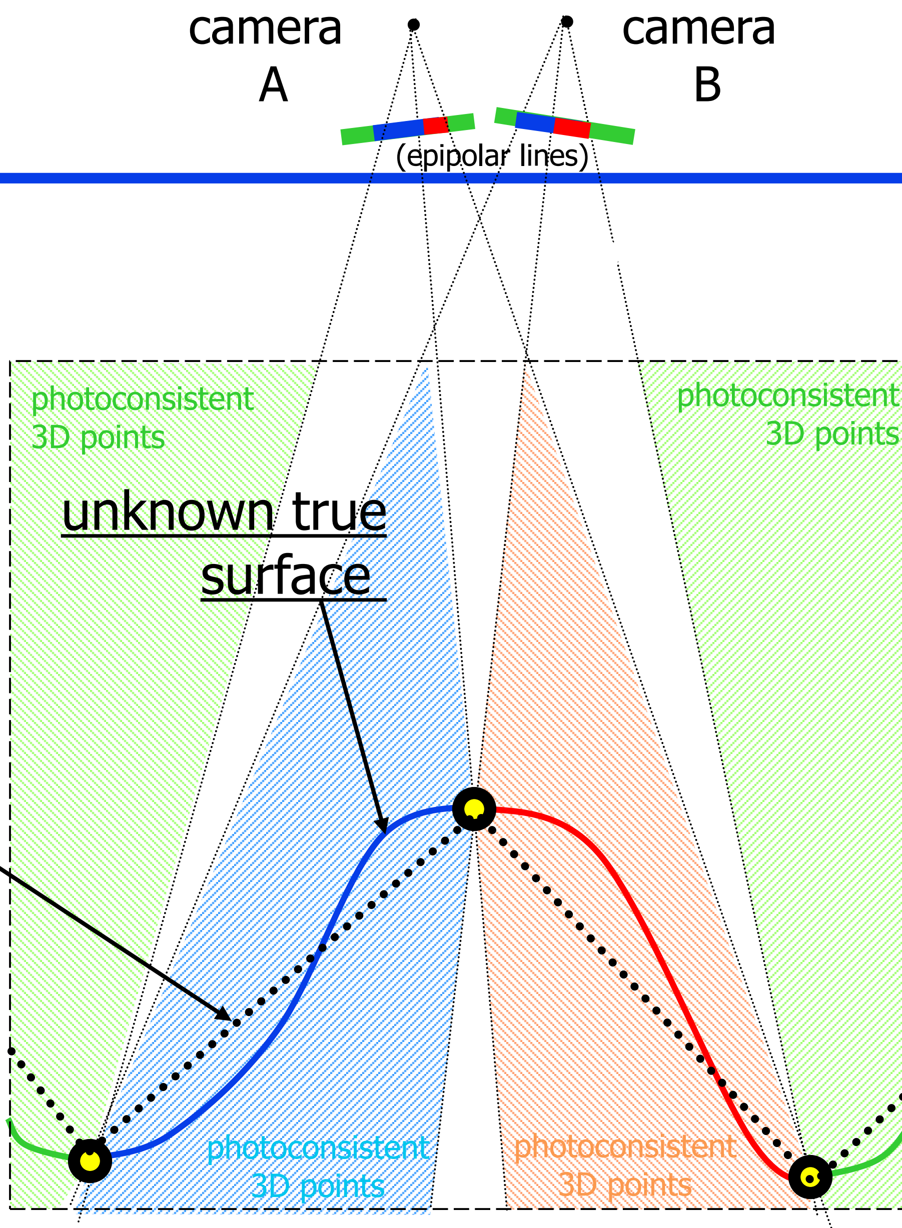
photo-consistency term

$$E(\mathbf{d}) = \sum_{p \in G} |I_p - I'_{p+d_p}| + \sum_{pq \in N} w\,|d_p - d_q|$$

regularization term

regularized depth map

- regularization helps to find **smooth** depth map consistent with points uniquely matched by photoconsistency
- regularization propagates information from textured regions (features) to ambiguous textureless regions

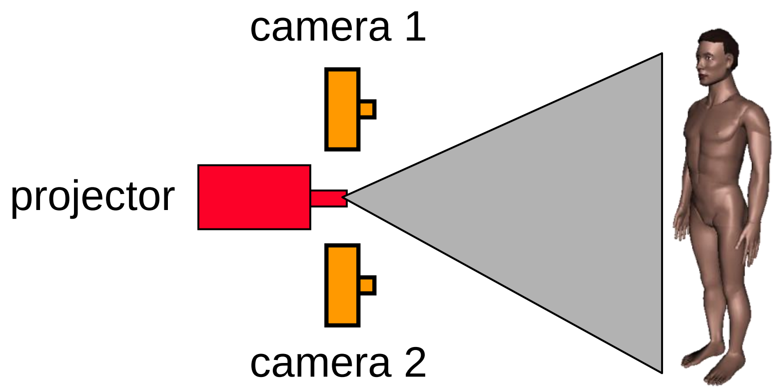


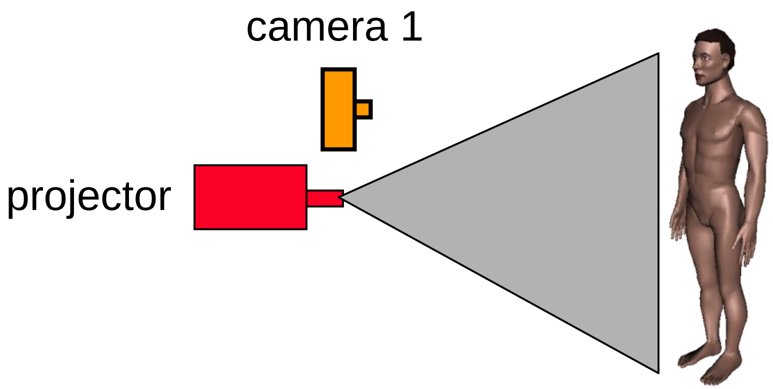
3D volume where surface is being reconstructed (epipolar plane)

More features/texture always helps!

Active Stereo (with structured light)

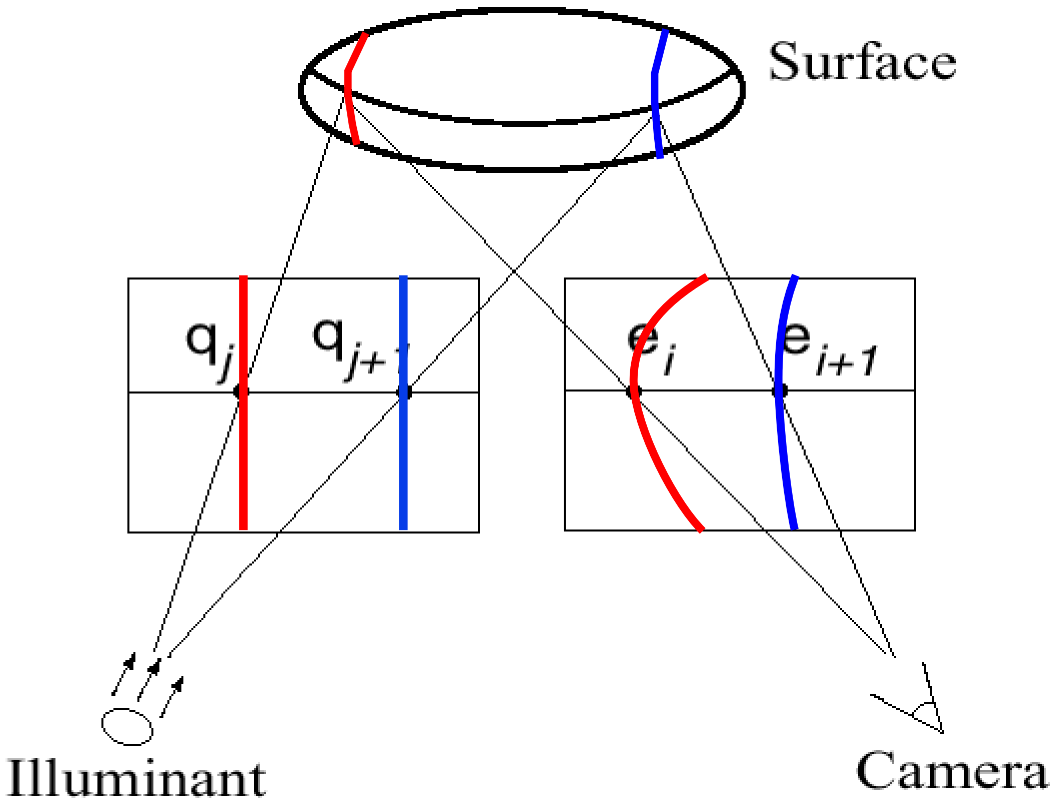
Li Zhang's one-shot stereo



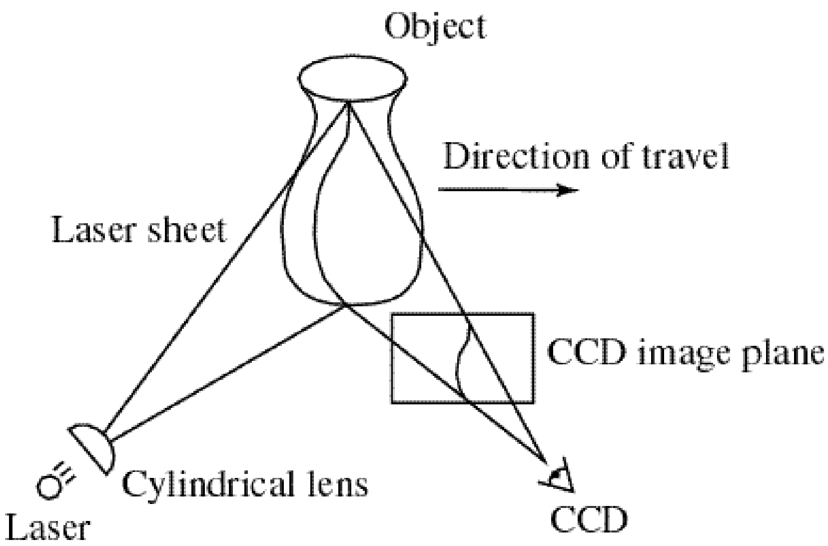


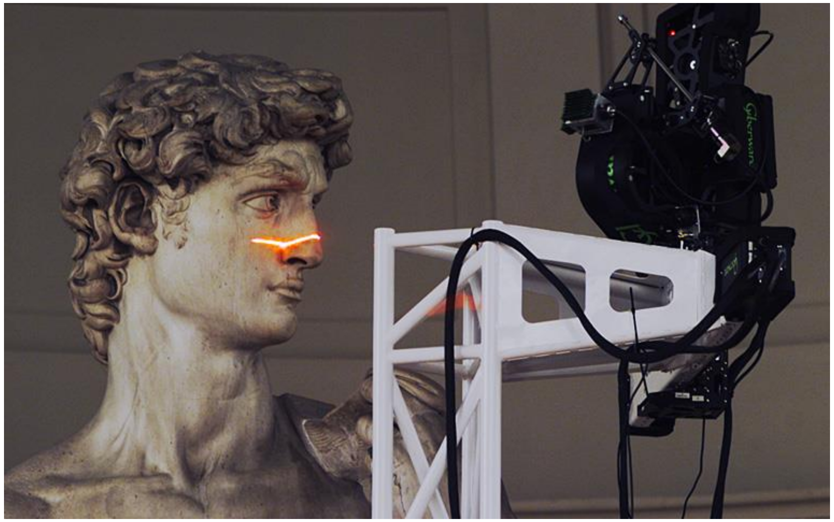
- Project "structured" light patterns onto the object
    - simplifies the correspondence problem

Active Stereo (with structured light)



Laser scanning





Digital Michelangelo Project [Levoy et al.]
http://graphics.stanford.edu/projects/mich/

- Optical triangulation
  - Project a single stripe of laser light
  - Scan it across the surface of the object
  - This is a very precise version of structured light scanning

Laser scanning

Digital Michelangelo Project [Levoy et al.]

http://graphics.stanford.edu/projects/mich/

Further considerations:

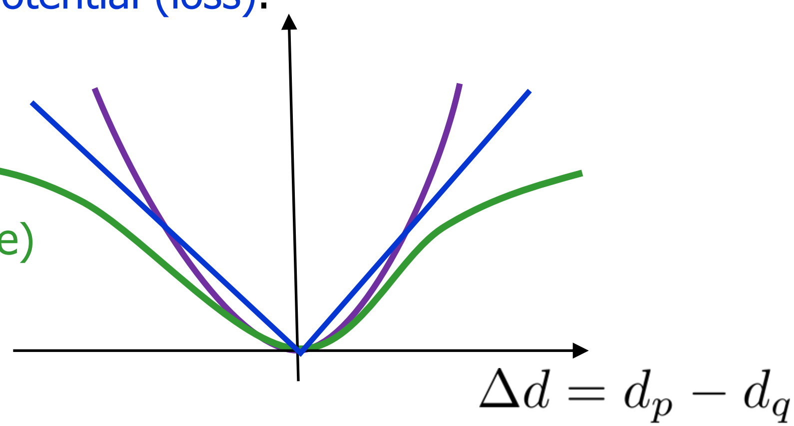
$$E(\mathbf{d}) = \sum_{p\in G} \underbrace{D_p(d_p)}_{\parallel} \; + \sum_{\{p,q\}\in N} \underbrace{V(d_p, d_q)}_{\parallel}$$

photo-consistency $\boxed{|I_p - I'_{p\oplus d_p}|}$ $\qquad$ $\boxed{w_{pq}\cdot|d_p - d_q|}$ spatial coherence

The last term is an example of **convex** regularization potential (loss).

- easier to optimize, but
- tend to over-smooth

practically preferred
robust regularization
(non convex – harder to optimize)



Note: once Δd is large enough,
there is no reason to keep increasing the penalty

Further considerations:

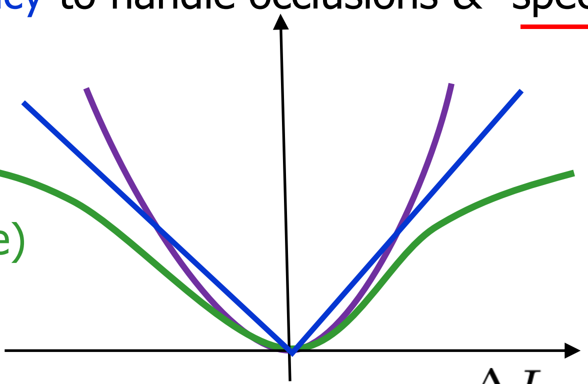
$$E(\mathbf{d}) = \sum_{p \in G} \underbrace{D_p(d_p)}_{\parallel} + \sum_{\{p,q\} \in N} \underbrace{V(d_p, d_q)}_{\parallel}$$

photo-consistency $\boxed{|\,I_p - I'_{p \oplus d_p}\,|}$ $\boxed{w_{pq} \cdot |\,d_p - d_q\,|}$ spatial coherence

?

Similarly, robust losses are needed for photo-consistency to handle occlusions & "specularities"

practically preferred
robust regularization
(non convex – harder to optimize)



Note: once ΔI is large enough,
there is no reason to keep increasing the penalty/loss

Further considerations:

$$E(\mathbf{d}) = \sum_{p \in G} D_p(d_p) + \sum_{\{p,q\} \in N} \cancel{V(d_p, d_q)}$$

photo-consistency $D_p(d_p) = \boxed{|I_p - I'_{p \oplus d_p}|}$

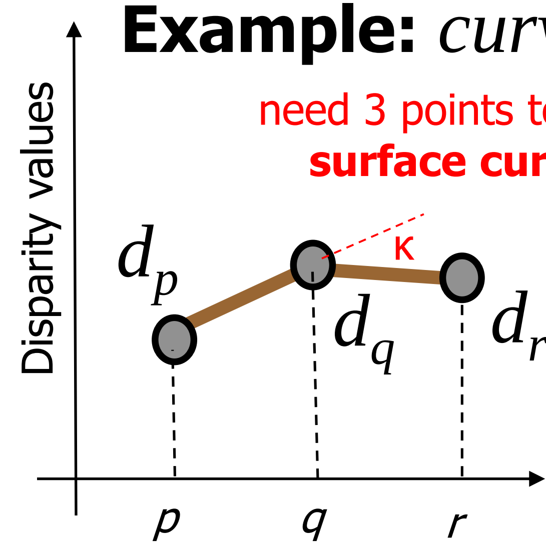
$$V(d_p, d_q, d_r) \quad p, q, r \in N$$

higher-order "coherence"

Many state-of-the-art methods use <u>higher-order regularizers</u>

Example: *curvature*

need 3 points to estimate **surface curvature**



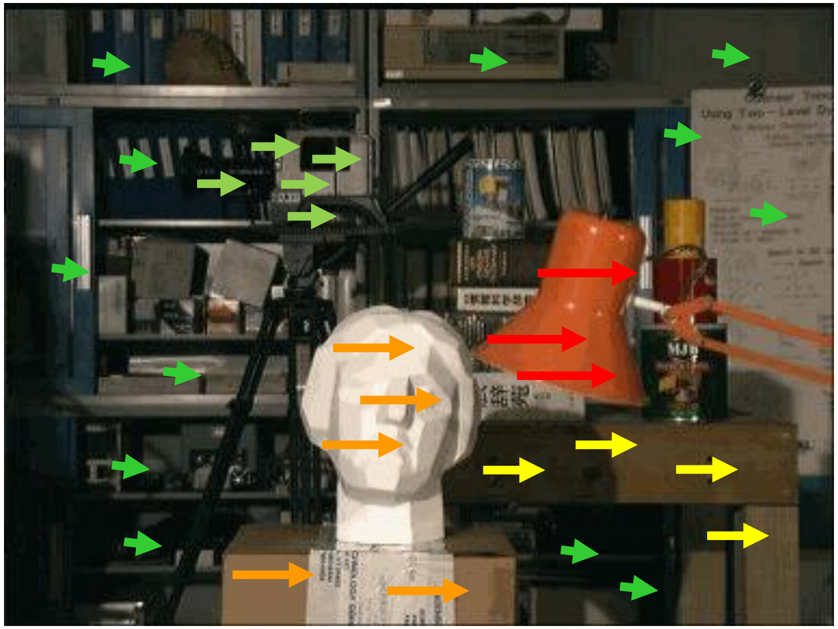
Q: why penalizing depth curvature instead of depth change?

From 1D correspondence (stereo) to 2D correspondence problems (motion)

1D shifts along **epipolar lines**.

Assumption for stereo:

only camera moves,
<u>3D scene is stationary</u>



vector field (motion) with a priori known direction

From 1D correspondence (stereo) to 2D correspondence problems (motion)

1D shifts along **epipolar lines**.

Assumption for stereo:

only camera moves,
<u>3D scene is stationary</u>

vector field (motion) with a priori *known direction*

⇨ We estimate only *magnitude* represented by a **scalar field** (disparity map)

From 1D correspondence (stereo) to 2D correspondence problems (motion)

In general, correspondences between two images may not be described by global models (like *homography*) or by shifts along known **epipolar lines**.

if 3D scene
is NOT stationary
motion is
vector field
with **arbitrary directions**
(no epipolar line constraints)

Society Of Robots

From 1D correspondence (stereo) to 2D correspondence problems (motion)

In general, correspondences between two images may not be described by global models (like *homography*) or by shifts along known **epipolar lines**.

For (non-rigid) motion the correspondences between two video frames are described by a general ***optical flow***

if 3D scene is NOT stationary motion is **vector field** with **arbitrary directions** (no epipolar line constraints)

SOCIETY OF ROBOTS

UNIVERSITY OF WATERLOO

From 1D correspondence (stereo) to 2D correspondence problems (motion)

$$E(\mathbf{v}) \;=\; \sum_{p\in G} \overset{\text{color-consistency}}{\underbrace{D_p(v_p)}_{\substack{\parallel \\ (I_p^t - I_{p+v_p}^{t+1})^2}}} \;+\; \sum_{\{p,q\}\in N} \overset{\text{regularity}}{\underbrace{V(v_p, v_q)}_{\substack{\parallel \\ w\cdot\|v_p - v_q\|^2}}}$$

Horn-Schunck 1981
optical flow regularization
- 2nd order optimization (pseudo Newton)
- Rox/Cox/Ishikawa's method only works for scalar-valued variables

optical flow
$\mathbf{v} = \{v_p\}$

if 3D scene is NOT stationary motion is **vector field** with **arbitrary directions**
(no epipolar line constraints)

Society Of Robots

more difficult problem

need 2D shift vectors v_p

(no epipolar line constraint)

From 1D correspondence (stereo) to 2D correspondence problems (motion)

State-of-the-art methods **segment** independently moving objects

We will discuss segmentation problem next

if 3D scene is <u>NOT stationary</u> motion is **vector field** with **arbitrary directions**
(no epipolar line constraints)

SOCIETY OF ROBOTS

optical flow

$$\mathbf{v} = \{v_p\}$$

more difficult problem

need 2D shift vectors v_p

(no epipolar line constraint)