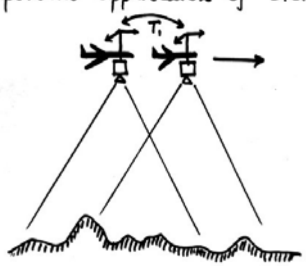
Binocular Stereo

- Take 2 images from different <u>known</u> viewpoints ⇒ <u>1st calibrate</u>
- Identify corresponding points between 2 images
- Derive the 2 lines on which world point lies
- Intersect 2 lines

Public Library, Stereoscopic Looking Room, Chicago, by Phillips, 1923

<u>Photogrammetry</u>

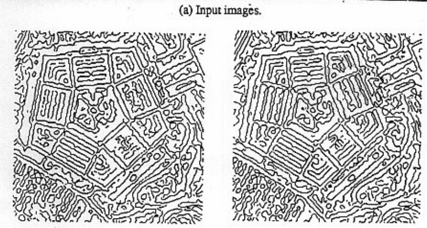
* Important application of Stereo.



* Recover 3-D terrain from sequence of overlapping images

* Relative positions of plane (T_1) must be known.

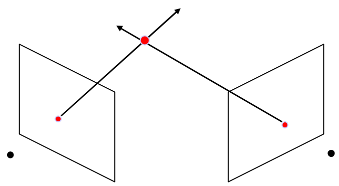
(a) Input images.



(b) Level 1 edge detection results.

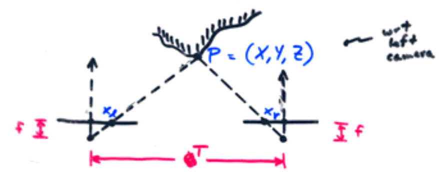
Figure 5.11. Stereo pair of the Pentagon.

Stereo



- Basic Principle: Triangulation
  - Gives reconstruction as intersection of two rays
  - Requires
    - calibration
    - ***point correspondence***

Stereo Geometry : Parallel Cameras



Known: f, focal length, of both cameras
T, baseline
x_l, x_r coords wrt cameras' principal points

By similar triangles

$$\frac{x_l}{f} = \frac{X}{Z} \qquad \frac{-x_r}{f} = \frac{(T-X)}{Z}$$

$$\Rightarrow x_l = \frac{fX}{Z}, \quad x_r = \frac{f(X-T)}{Z}$$

Substituting and simplifying we get

$$X = \frac{T x_l}{x_l - x_r} \qquad Y = \frac{T y_l}{x_l - x_r}$$

$$Z = \frac{Tf}{x_l - x_r}$$

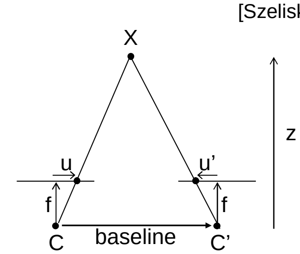
$$d \triangleq x_l - x_r \qquad \text{(horizontal) disparity}$$

$$\Rightarrow Z = f\frac{T}{d}$$

* $d = 0 \Rightarrow$ P at infinity
* Large $d \Rightarrow$ P close to cameras
* Z inversely proportional to d
* Z proportional to f and T
* Given fixed error in determining d, accuracy of Z increases with increasing baseline T, but then images are less similar

Depth from Disparity

input image (1 of 2) | depth map [Szeliski & Kang '95] | 3D rendering



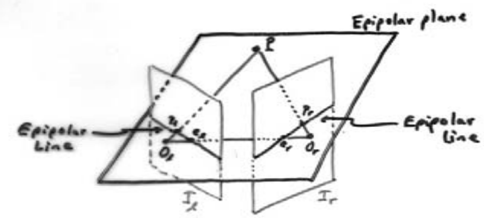
$$disparity = u - u' = \frac{baseline * f}{z}$$

Multi-View Geometry

- Different views of a scene are not unrelated
- Several relationships exist between two, three and more cameras

- *Question: Given an image point in one image, does this restrict the position of the corresponding image point in another image?*

EPIPOLAR GEOMETRY



- Left image I_l has optical center O_l and focal length f_l
- Right image I_r has optical center O_r and focal length f_r
- Scene pt $P = (X, Y, Z)$ and camera optical centers O_l and O_r define EPIPOLAR PLANE

Epipolar Geometry: Formalism

- Depth can be reconstructed based on corresponding points (disparity)
- Finding corresponding points is hard & computationally expensive
- Epipolar geometry helps to significantly reduce search from 2-D to 1-D line

Epipolar Geometry: Demo

Java Applet

http://www-sop.inria.fr/robotvis/personnel/sbougnou/Meta3DViewer/EpipolarGeo.html

Sylvain Bougnoux, INRIA Sophia Antipolis

- Scene point P projects to image point $p_l = (x_l, y_l, f_l)$ in left image and point $p_r = (x_r, y_r, f_r)$ in right image
- Epipolar plane contains P, O_l, O_r, p_l and p_r – called **co-planarity constraint**
- Given point p_l in left image, its corresponding point in right image is on line defined by intersection of epipolar plane defined by p_l, O_l, O_r and image I_r – called **epipolar line** of p_l
- In other words, p_l and O_l define a ray where P may lie; projection of this ray into I_r is the **epipolar line**

Marc Pollefeys, University of Leuven, Belgium, Siggraph 2001 Course

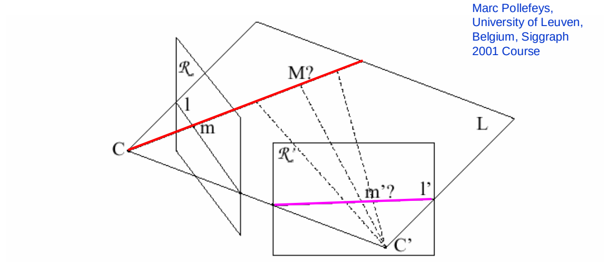


Figure 3.5: *Correspondence between two views. Even when the exact position of the 3D point* M *corresponding to the image point* m *is not known, it has to be on the line through* C *which intersects the image plane in* m. *Since this line projects to the line* l′ *in the other image, the corresponding point* m′ *should be located on this line. More generally, all the points located on the plane defined by* C, C′ *and* M *have their projection on* l *and* l′.

Epipolar Line Geometry

- **Epipolar Constraint**: The correct match for a point p_l is constrained to a 1D search along the epipolar line in I_r
- All epipolar planes defined by all points in I_l contain the line $O_l O_r$
  - $\Rightarrow$ All epipolar lines in I_r intersect at a point, e_r, called the **epipole**
- Left and right epipoles, e_l and e_r, defined by the intersection of line O_lO_r with the left and right images I_l and I_r, respectively

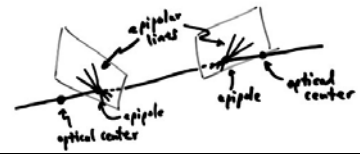
- If I_l and I_r parallel, the epipoles are at infinity, and the epipolar lines are parallel
- Given a pair of images, I_l and I_r, the warping transformation that projects I_l and I_r onto a plane parallel to O_lO_r is called RECTIFICATION. Usually done so that epipolar lines are parallel to new images' horizontal axes

⇒ Epipolar constraint = 1D search along image scanline

- EPIPOLAR CONSTRAINT: The correct match for a point p_l is constrained to a 1D search along the epipolar line in I_r
- All epipolar planes defined by all points in I_l contain the line O_lO_r.

⇒ All epipolar lines in I_r intersect at a point, e_r, called the epipole

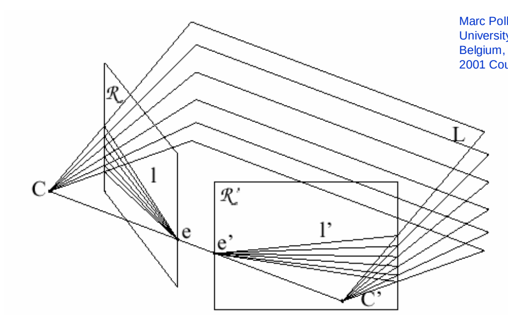
- Left and right epipoles defined by intersection of line O_lO_r with I_l and I_r, respectively



3

Epipolar Geometry

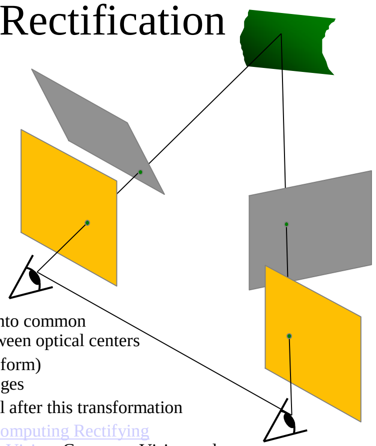
Marc Pollefeys, University of Leuven, Belgium, Siggraph 2001 Course



Epipolar Geometry: Rectification

- [Trucco 157-160]
- **Motivation**: Simplify search for corresponding points along scan lines (avoids interpolation and simplify sampling)
- **Technique**: Image planes parallel -> pairs of conjugate epipolar lines become collinear and parallel to image axis.

Stereo Image Rectification



- Image Reprojection
  - reproject image planes onto common plane parallel to line between optical centers
  - a homography (3x3 transform) applied to both input images
  - pixel motion is horizontal after this transformation
  - C. Loop and Z. Zhang, Computing Rectifying Homographies for Stereo Vision, Computer Vision and Pattern Recognition Conf., 1999

Rectification

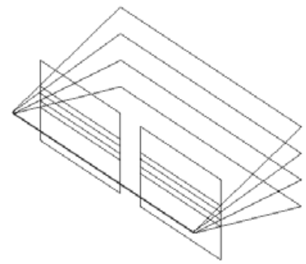
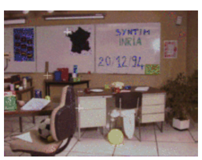


Figure 7.15: Standard stereo setup

Marc Pollefeys, University of Leuven, Belgium, Siggraph 2001 Course

Rectification Example

before




after


Rectification Procedure

Given: Intrinsic and extrinsic parameters for 2 cameras

1. Rotate left camera so that the epipole goes to infinity along the horizontal axis
   ⇒ left image parallel to baseline
2. Rotate right camera using same transformation
3. Rotate right camera by R, the transformation of the right camera frame with respect to the left camera
4. Adjust scale in both cameras

Implement as backward transformations, and resample using bilinear interpolation

Definitions

- **Conjugate Epipolar Line:** A pair of epipolar lines in I_l and I_r defined by P, O_l and O_r
- **Conjugate (i.e., corresponding) Pair:** A pair of matching image points from I_l and I_r that are projections of a single scene point

Can We Determine Epipolar Geometry?

Given scene point P, let

$$P_l = [X_l, Y_l, Z_l]^T$$
$$P_r = [X_r, Y_r, Z_r]^T$$

define P as a vector wrt left and right camera coordinate frames.

Let $p_l = [x_l, y_l, z_l]^T$

$p_r = [x_r, y_r, z_r]^T$

be projections of P onto left & right images

$\Rightarrow$ Relation b/w P_l and P_r is:

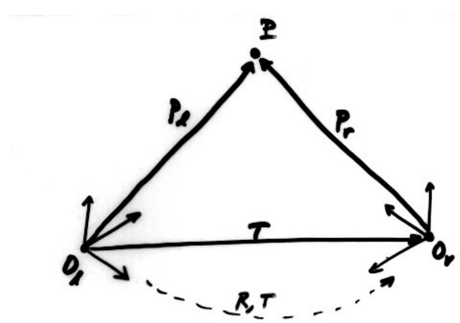
$$\begin{bmatrix} X_r \\ Y_r \\ Z_r \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_l \\ Y_l \\ Z_l \end{bmatrix} - \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

i.e.

$$P_r = R(P_l - T)$$

where $T = (O_r - O_l)$

$R^T R = R R^T = I$ (R orthogonal)



- Coplanarity constraint $\Rightarrow$

vectors P_l (defined by O_l, P)

T (defined by O_l O_r)

and $P_l - T$ (defined by O_r, P wrt O_l coord. frame)

are <u>coplanar</u>

$\Rightarrow$ mixed product $= 0$

$$(P_l - T)^T \cdot [T \times P_l] = 0$$

i.e., one vector lies in plane defined by the other 2 vectors

Since $P_r = R(P_l - T)$ and $R^T = R^{-1}$ $\Rightarrow$

$$(R^T P_r)^T T \times P_l = 0$$

Cross product of vectors can be rewritten as:

$$T \times P_l = S P_l = [T]_x P_l$$

where $S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}$ (rank = 2)

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

7

Notation and Definitions

- Given $a = (a_1\ a_2\ a_3)^T$

then
$$[a]_\times \triangleq \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

a 3×3, skew-symmetric matrix and singular matrix

- Given 2 3-vectors, a and b,

$$a \times b \triangleq a \wedge b \triangleq [a]_\times b \triangleq (a^T [b_\times])^T$$

Substituting we get:

$$\boxed{P_r^T E P_l = 0}$$ Epipolar Equation

where $E = RS$

E called Essential Matrix

Using perspective projection equation, can also show

$$p_r^T E p_l = 0$$

for corresponding image points p_l and p_r in I_l and I_r, respectively, though p_l and p_r are in camera coordinates.

But points known in image coordinates

⇒ need to know transformation from camera coords to image coords, i.e., intrinsic parameters of camera (focal length, coords of principal pt., effec. pixel size)

- Equivalently, epipolar equation can be written as:

$$\boxed{P_l^T E^T P_r = 0}$$

where $E^T = -[T_o]_\times R^T$

- Observation: Ep_l can be interpreted as the vector representing the epipolar line associated with p_l in the right image. So, the epipolar equation, $p_r^T E p_l = 0$ expresses the fact that p_r lies on the epipolar line associated with the vector Ep_l

- $E = [T]_\times R$ (is usual way of writing E) is 3×3 matrix, 5 DOFs (3 DoFs for T, 3 DoFs for R, but an overall scale ambiguity)

- Essential matrix encodes information about the extrinsic parameters (rotation and translation) only, ~~it~~ is defined in terms of camera coordinates

- If $\tilde{p}_l$ and $\tilde{p}_r$ are corresponding points in image (pixel) coordinates, then it can be shown:

$$\boxed{\tilde{p}_r^T F \tilde{p}_l = 0}$$

where $F = M_r^{-T} E M_l^{-1}$ ← FUNDAMENTAL MATRIX

$$M_l = \begin{pmatrix} -f_l/s_{x_l} & 0 & o_{x_l} \\ 0 & -f_l/s_{y_l} & o_{y_l} \\ 0 & 0 & 1 \end{pmatrix}$$

Intrinsic parameters of left camera:

f_l : focal length

(o_{x_l}, o_{y_l}) : coords of principal pt.

(s_{x_l}, s_{y_l}) : effective pixel size

• Fundamental Matrix Properties

* Encodes info. about **both** intrinsic and extrinsic params
  $\Rightarrow$ If you can estimate F, you can reconstruct the epipolar geometry with **no** info about cameras
* Has rank 2
* 3×3
* Has 7 DOFs
* Can be estimated from at least 8 correspondences
  "The 8-point algorithm"
  - Each point gives a homogeneous linear equation of form $\tilde{p}_r^T F \tilde{p}_\ell = 0$
  - Homogeneous system $\Rightarrow$ solution unique up to scale factor
  - Usually use > 8 points so system is overdetermined and solve using SVD

10

Fundamental Matrix Properties

- 3×3 matrix, rank 2, 7 DOFs (2 for e_ℓ, 2 for e_r, 3 for epipolar line homog.)
- If p_ℓ and p_r are corresponding points, then $p_r^T F p_\ell = 0$
- $\ell_r = F p_\ell$ is the epipolar line corresponding to p_ℓ
- $\ell_\ell = F^T p_r$ is the epipolar line corresponding to p_r
- $F e_\ell = 0$
- $F^T e_r = 0$

the epipoles

(i.e. e_r satisfies $e_r^T (F p_\ell) = (e_r^T F) p_\ell = 0$ for all p_ℓ)

- Given F, a corresponding pair of (canonical) camera matrices is
  $P_\ell = [I \mid 0]$, $P_r = [[e_r]_\times F \mid e_r]$
- Given 2 camera matrices, can also compute F

10.1

Computing the Fundamental Matrix

Given a correspondence $x \leftrightarrow x'$

we know $x'^T F x = 0$

or

$$x'x f_{11} + x'y f_{12} + x' f_{13} + \cdots + f_{33} = 0$$

or

$$(x'x, x'y, x', y'x, y'y, y', x, y, 1) \begin{pmatrix} f_{11} \\ f_{12} \\ f_{13} \\ \vdots \\ f_{33} \end{pmatrix} = 0$$

where $F = \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix}$

$$f = (f_{11}, f_{12}, \ldots, f_{33})^T$$

The Epipolar Constraint

$$\tilde{p}_r^T F \tilde{p}_\ell = 0$$

$$(u, v, 1) \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix} \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0$$

$$(uu', uv', u, vu', vv', v, u', v', 1) \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0$$

Homogeneous equation in coefficients of F

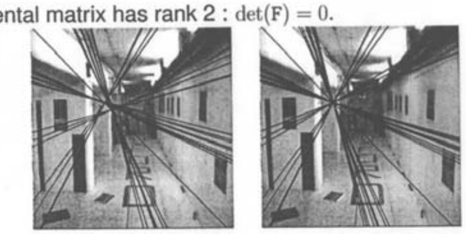
Given n conjugate pairs

$$Af = \begin{bmatrix} x_1' x_1 & x_1' y_1 & \cdots & x_1 & y_1 & 1 \\ \vdots & \vdots & & & \vdots & \\ x_n' x_n & x_n' y_n & \cdots & x_n & y_n & 1 \end{bmatrix} \begin{pmatrix} f_{11} \\ f_{12} \\ \vdots \\ f_{33} \end{pmatrix} = 0$$

- Solution determined up to scale only
  $\Rightarrow$ need at least 8 point correspondences
- 8 points $\Rightarrow$ unique solution
  >8 pts $\Rightarrow$ least-squares solution
  1. Form equations $Af = 0$
  2. Take SVD: $A = UDV^T$
  3. Solution is last column of V
     (eigenvector corresp to smallest eigenvalue)

The singularity constraint

Fundamental matrix has rank 2 : $\det(F) = 0$.



Left: Uncorrected F – epipolar lines are not coincident.

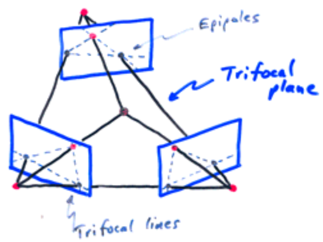
Right: Epipolar lines from corrected F.

- Least-squares solution does not enforce the singularity constraint — all epipolar lines intersect at epipole
  $\Rightarrow \det(F) = 0$
  (F has rank 2)
- 8-point Algorithm (Hartley)
  1. Compute linear solution:
     Solve $Af = 0$ to find F
  2. Constraint enforcement:
     Replace F by F', the "closest" singular matrix to F
- 8-point algorithm performs badly with noise:
  sensitive to origin position and scaling of data positions
  $\Rightarrow$ translate and scale points
  origin = centroid of pts
  scale so "average pt" is $(1,1,1)^T$

Normalized 8-point Algorithm

1. Normalization
   $\hat{x}_i = T x_i$
   $\hat{x}_i' = T' x_i'$
2. Linear Solution
   Compute $\hat{F}$ by solving $Af = 0$
3. Singularity Constraint
   Find closest singular $\hat{F}'$ to $\hat{F}$
4. Denormalization
   $F = T'^T \hat{F}' T$

Trinocular Epipolar Geometry

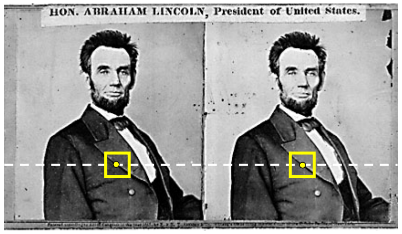


$$\begin{cases} p_1^T E_{12} p_2 = 0 \\ p_2^T E_{23} p_3 = 0 \\ p_3^T E_{31} p_1 = 0 \end{cases}$$

3 epipolar constraints (only 2 are independent)

Given E_{12}, E_{23}, E_{31} and corresponding points (p_2, p_3), can compute p_1
$\Rightarrow$ image "transfer"

Basic Stereo Algorithm

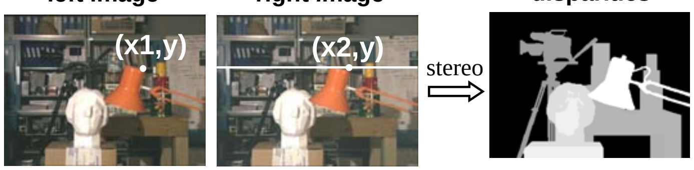


For each epipolar line
- For each pixel in the left image
  - compare with every pixel on same epipolar line in right image
  - pick pixel with minimum match cost

Improvement: match ***windows***

Stereo Correspondence

left image **right image** **disparities**



disparity = **x1-x2** is inversely proportional to depth

⇩

3D scene structure recovery

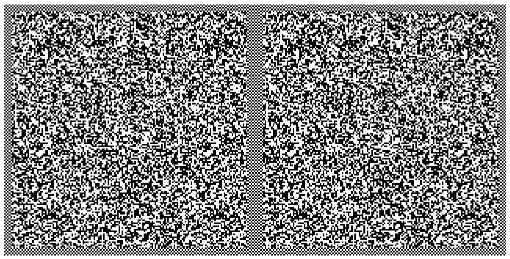
DECISIONS

- FEATURES FOR MATCHING
  - BRIGHTNESS VALUES
  - POINTS
  - EDGES
  - REGIONS
- MATCHING STRATEGY
  - BRUTE-FORCE
  - COARSE-TO-FINE (MULTI-RESOLUTION)
  - RELAXATION
  - DYNAMIC PROGRAMMING
- MATCHING CONSTRAINTS
  - EPIPOLAR LINES
  - UNIQUENESS
  - CONTINUITY

  ⋮

Stereo Matching

- Features vs. pixels?
  - Do we extract features prior to matching?



Julesz-style Random Dot Stereogram

Matching Methods

Feature-Based Matching

- Identify image coords where distinctive local features occur
- Examples:
  - Edge points
  - Corner points (e.g. Tomasi + Kanade)
  - Moravec's interest operator
- \+ Relatively robust and insensitive to Δ viewpoint, Δ illumination, Δ surface orientation
- − At depth discontinuities, edges do not remain fixed on surface
- − Produces only a sparse reconstruction

Intensity-Based Matching

* Assume physical point in scene projects to same brightness pattern in 2 views

* Use left image patch as a template and find corresponding right image patch

* Big window ⇒ more reliable statistically

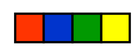
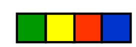
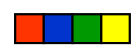
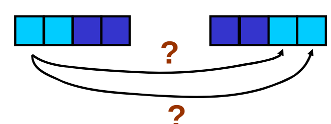
Small window ⇒ more precise; less likely to violate assumption

\+ Produces dense reconstruction

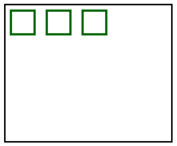
− Foreshortening, Δ illumination, etc. alter brightness pattern

* 2 main measures: SSD and CC

Difficulties in Stereo Correspondence

	left image	right image
Perfect case: **never happens!**		
1) Image noise:		
2) Low texture:		

Local Approach

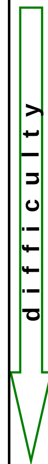


- Look at one image patch at at time
- Solve many small problems independently
- Faster, less accurate

Global Approach

- Look at the whole image
- Solve one large problem
- Slower, more accurate

How Difficult is Correspondence?



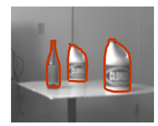
high texture

- **local** works for high texture
- enough texture in a patch to disambiguate

medium texture

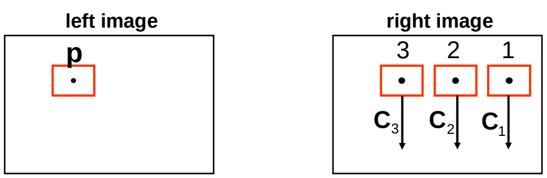
- **global** works up to medium texture
- propagates estimates from textured to untextured regions

low texture



- **salient regions** work up to low texture
- propagation fails; some regions are inherently ambiguous, match only unambiguous regions

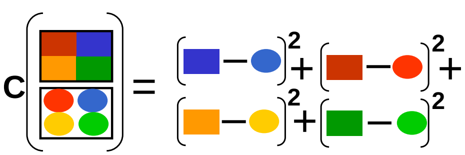
Local Approach [Levine'73]



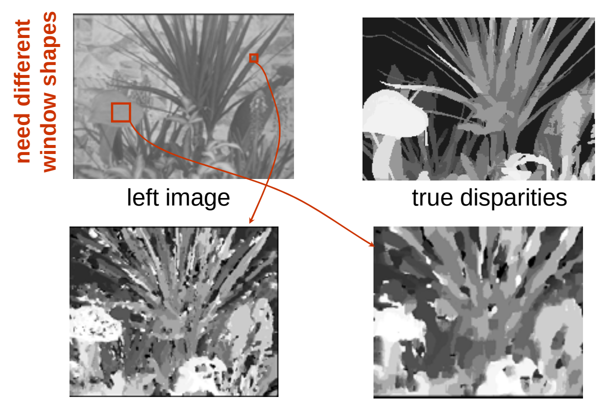
$\boxed{d_p = i \quad \text{which gives best } C_i}$

Common **C**

(SSD)



Fixed Window Size Problems



left image

true disparities

fixed small window

fixed large window

Window Size

W = 3

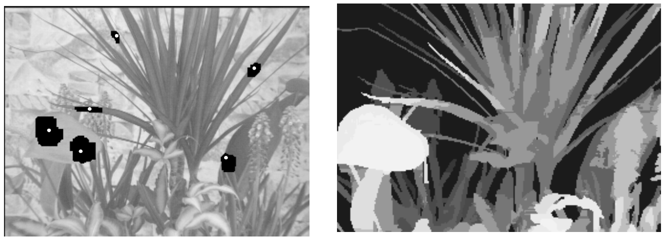
W = 20

- Effect of window size
  - Smaller window
    - +
    - –
  - Larger window
    - +
    - –

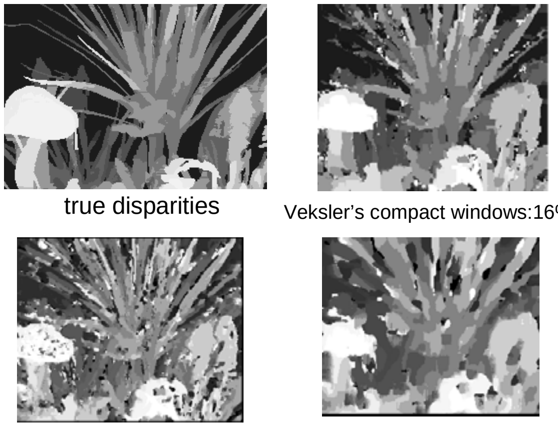
Better results with *adaptive window*

- T. Kanade and M. Okutomi, A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment, Proc. Int. Conf. Robotics and Automation, 1991
- D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion, Int. J. Computer Vision, **28**(2):155-174, 1998

Sample Compact Windows [Veksler 2001]



Comparison to Fixed Window



true disparities

Veksler's compact windows:16% errors

fixed small window: 33% errors

fixed large window: 30% errors

Results (% Errors)

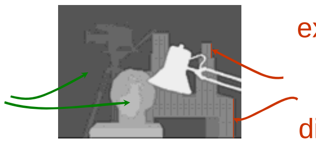
	Algorithm	Tsukuba	Venus	Sawtooth	Map
all global	Layered	1.58	1.52	0.34	0.37
	Graph cuts	1.94	1.79	1.30	0.31
	Belief prop	1.15	1.00	0.98	0.84
	GC+occl.	1.27	2.79	0.36	1.79
	Graph cuts	1.86	1.69	0.42	2.39
	Multiw. Cut	8.08	0.53	0.61	0.26
	Veksler's var. windows	3.36	1.67	1.61	0.33

Constraints

1) corresponding pixels should be close in color

2) most nearby pixels should have similar disparity

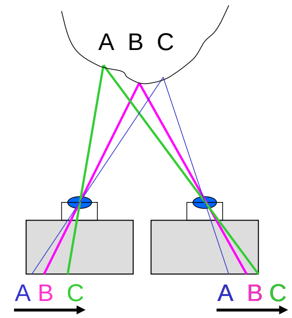
disparity continuous in most places



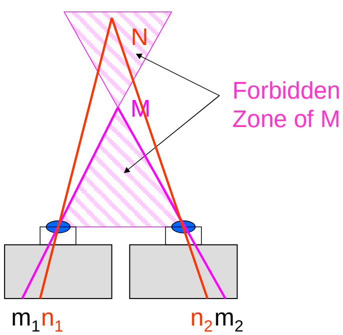
except a few places: disparity discontinuity

Additional geometric constraints for correspondence

- **Ordering of points**: Continuous surface: same order in both images.
- Is that always true?



Forbidden Zone



Practical applications:

- Object bulges out: ok
- In general: ordering across whole image is not reliable feature
- Use ordering constraints for neighbors of M within small neighborhood only

Constraints:

- **Uniqueness**
  Each Edge Point can Match at most 1 Point in other Image
  (Each Point Corresponds to a Single Point in World)
- **Figural Continuity**
  Edge Points along a Contour should Match Edges along a Similar Contour in other Image
  (Contours in Scene appear Similarly in 2 Images)
- **Disparity Gradient**
  Nearby Edge Points in Image should have Similar Disparities
  (Points usually associated with same Surface)
- **Multi-Resolution**
  Edge Points which occur at Multiple Resolutions are more likely to be Physically Significant
- **Detailed Match**
  Matching Edge Points should have Similar Properties (e.g. Orientation and Contrast)
- Monotonicity
  Order of matching points preserved in L and R

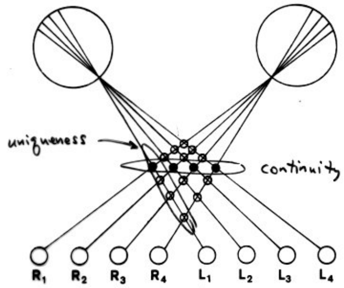
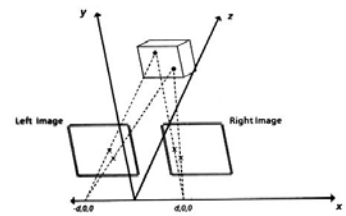


Figure 2.2
The False Targets Problem. Each of the four points in one view could match any of the four projections in the other view. Of the 16 possible matches indicated by the circles, only those indicated by the filled circles are actually perceived. (Redrawn from Julesz [1971, fig. 4.5-1], and Marr and Poggio [1979, p.302]).

Disparity Gradient (Pollard, Mayhew and Frisby 1985) (Prazdny 1985)

- pair of edges from same surface in scene appear with similar disparities
- allowed disparity difference increases with separation between matches
- sometimes discriminates only weakly



CONTINUITY CONSTRAINT

(MARR AND POGGIO ; GRIMSON)

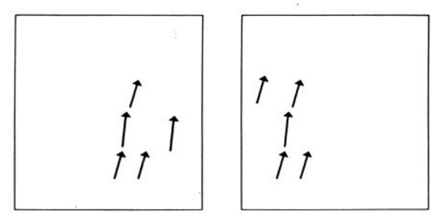
- NEARBY IMAGE POINTS ARE PROJECTIONS OF NEARBY 3D POINTS

⇒ SMOOTHNESS IN DISPARITY MAP

- DOESN'T APPLY AT REGION BOUNDARIES AND NON-OPAQUE OBJECTS

Figural Continuity: (Mayhew and Frisby 1981)

- edges on a contour in one image match edges along a similar contour in the other image
- non-contour edges do not meet requirements



Types of Stereo Algorithms

1. Local Methods based on Correlation
   * Normalized cross-correlation or SSD match using m x m window centered on each point
   * Computes dense depth map
2. Global Optimization
   * Define an energy function

$$E(f) = E_{smooth}(f) + E_{data}(f)$$

where f is the disparity value at a given pixel, p.

Example:

$$E_{data} = \sum_{p} [I(p) - I'(p + disparity(p))]^2$$

$$E_{smooth} = \sum_{p} (\text{\# adjacent pixels w/ different disparity than } p\text{'s})$$

* E_{smooth} should be piecewise-smooth, not smooth everywhere, to allow for depth discontinuities
* Minimize energy function E using optimization methods e.g. dynamic programming, simulated annealing
* May find local minimum
* Computes dense depth map

Marr-Poggio Stereo Algorithm

1. Convolve 2 rectified images with $\nabla^2 G_\sigma$ filters of size $\sigma_1 < \sigma_2 < \sigma_3 < \sigma_4$
2. Detect zero-crossings in all images
3. At coarsest scale, σ_4, match zero-crossings with same parity and roughly same orientation in a $[-w_\sigma, +w_\sigma]$ disparity range with $w_\sigma = 2\sqrt{2}\,\sigma$
4. Use disparities found at coarser scales to cause unmatched regions at finer scales to come into correspondence

⇒ Result is a sparse depth map

3 Constraints in Marr-Poggio

1. UNIQUENESS

   EACH POINT IN LEFT IMAGE CAN MATCH ONLY 1 POINT IN RIGHT IMAGE, CORRESPONDING TO FACT THAT A SINGLE DISPARITY VALUE CAN BE ASSIGNED

2. CONTINUITY

   SURFACE SMOOTHNESS ⇒ DISPARITY SMOOTHNESS ALMOST EVERYWHERE (EXCEPT AT DEPTH DISCONTINUITIES — OCCLUDING CONTOURS)

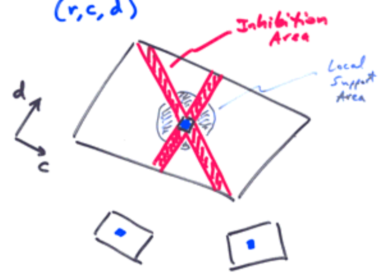
3. MULTI-RESOLUTION COARSE-TO-FINE TRACKING

Zitnick & Kanade's Algorithm

www-2.cs.cmu.edu/~clz/stereo.html

IEEE Trans. PAMI 22(7), 2000

3D Disparity Space Representation
(r,c,d)



1. Construct 3D array (r,c,d) for each pixel in reference image and disparity range

2. Compute initial match values
$$L_0(r,c,d) = \text{NCC}(I_L, I_R, r, c, d)$$
→ computes match between $I_L(r,c)$ and $I_R(r, c+d)$

3. Iteratively update match values until match values converge
$$L_{n+1}(r,c,d) = L_0(r,c,d) * R_n(r,c,d)$$
where
$$R_n(r,c,d) = \left(\frac{S_n(r,c,d)}{\sum_{\Psi(r,c,d)} S_n(r'',c'',d'')} \right)^{\alpha}$$
← inhibition area

and where
$$S_n(r,c,d) = \sum_{\Phi} L_n(r+r', c+c', d+d')$$
← local support area

$\alpha > 1$

Φ corresponds to smoothness assumption

Ψ corresponds to uniqueness assumption

4. For each pixel (r,c), find (r,c,d) with max match value

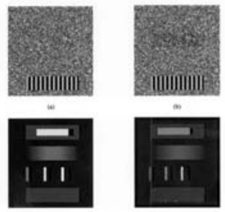
5. If max match value $> t$, then output disparity d; otherwise, classify as "occluded"

* Converges to 1 at correct matches

* To prevent over-smoothing & loss of detail

  $L_0 * R_n$ means only pairs with similar initial intensities will contribute to match value computation

(Figure 6(e)) are smooth while recovering several details at the same time. The slanted roof of the lower building and the water tower on the rooftop are clearly visible. Depth discontinuities around the small building attached to the tower are preserved. 15 iterations were used and the inhibition constant was set to 2.



(a) (b) (c) (d)

Figure 3: Synthetic Scene, 50% density: (a) Reference (left) image; (b) Right image; (c) True Disparity map, black areas are occluded; (d) Disparity map found using 3x3x3 local support area, black areas are detected occlusions.

Random Dot Stereogram

Local Support Area RxCxD	% Disparity Correct	% Occlusion Correct	% Occlusion Found
3x3x3	99.44	97.11	79.61
5x5x3	99.29	95.41	71.05
7x7x3	98.73	81.10	58.42

Table 1: The percentage of disparities found correctly, the percentage of the detected occlusions that are correct and the percentage of the true occlusions found for three different local support area sizes using the random dot stereo pair.

9

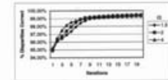
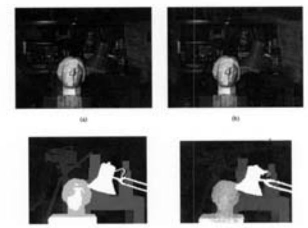


Figure 4: Convergence rate for inhibition constant α of 1.5, 2 and 4 over 20 iterations using the random dot stereogram.



(a) (b) (c) (d)

Figure 5: Head scene provided by University of Tsukuba: (a) Reference (left) image; (b) Right image; (c) Ground truth disparity map with black areas occluded, provided courtesy of U. of Tsukuba; (d) Disparity map found using our algorithm with a 5x5x3 local support area, black areas are detected occlusions. The match values were allowed to completely converge. Disparity values for narrow objects such as the lamp stem are found correctly.

10

U. of Tsukuba Stereo Image Pair

Local Support Area RxCxD	% Disparity Correct	% Occlusion Correct	% Occlusion Found
3x3x3	97.12	46.30	60.15
5x5x3	98.02	66.58	51.84
7x7x3	97.73	63.23	44.85

Table 2: The percentage of disparities found correctly, the percentage of the detected occlusions that are correct and the percentage of the true occlusions found for three different local support area sizes using the U. of Tsukuba stereo pair.

Confusion matrix for the disparity map obtained from U. of Tsukuba data.

	Ground Truth **Occluded**	Ground Truth **Non-occluded**	Total
Occluded	860	285	1,145
Non-Occluded	1,042	Correct 82,597 Incorrect 1,121	84,760
Total	1,902	84,003	85,905

Table 3: The number of occluded and non-occluded pixels found using our algorithm compared to the ground truth data provided by University of Tsukuba. A 5x5x3 area was used for the local support and the disparity values were allowed to completely converge.

Method	Errors > ±1
Zitnick and Kanade	1.4
GPM-MRF [4]	2.8
LOG-filtered L_1 [4]	9.0
Normalized correlation [4]	10.0
Nakamura *et al.* [19] (25 images)	0.3
Nakamura *et al.* [19] (9 images)	0.9

Table 4: Comparison of various algorithms using the ground truth data supplied by University of Tsukuba. Error rates of greater than one pixel in disparity are for pixels labeled non-occluded in the ground truth data. GPM-MRF [4] has approximately twice the error rate of our algorithm. LOG-filtered L_1 and Normalized correlation are supplied for comparison to more conventional algorithms. The University of Tsukuba group provides their results using a 3x3 and 5x5 camera array. The error results for their method use fewer pixels since the chance of a pixel being occluded increases with the number of camera angles used.

(a) (b) (c)

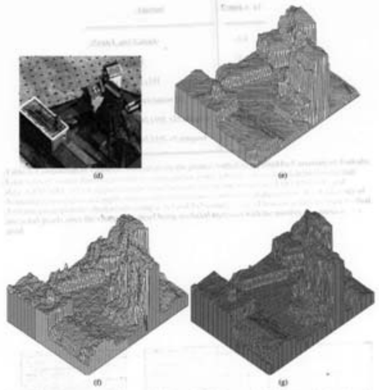


Figure 6: Coal mine scene; (a) Reference (left) image; (b) Right image; (c) Disparity map obtained by using proposed method with a 3x3x3 local support area, black areas are detected occlusions; (d) Real oblique view of the coal mine model; (e) Isometric plot of the disparity map of Figure 6(c); (f) Isometric plot of the disparity map using multi-baseline stereo with three images as presented in [21]; (g) Isometric plot of the disparity map using multi-baseline stereo with adaptive window with three images as presented in [12].

Global Approach [Horn'81, Poggio'84, …]

encode desirable properties of **d** in **E**(**d**):

$$\mathbf{E}(\mathbf{d}) = \mathbf{E}\left(\begin{array}{|c|c|c|} \hline d_p & d_q & d_r \\ \hline & & \\ \hline & & \\ \hline \end{array}\right)$$

$$\underbrace{\arg\min_{\mathbf{d}} \mathbf{E}(\mathbf{d})}^{\text{MAP-MRF}} = \underbrace{\sum_{p \in P} \mathbf{M}(d_p)}_{\text{match pixels of similar color}} + \underbrace{\sum_{\{p,q\} \in \text{Neighbors}} \mathbf{P}(d_p, d_q)}_{\text{most nearby pixels have similar disparity}}$$

NP-hard problem ⇒ need approximations

Stereo as Energy Minimization

- Matching cost formulated as energy
  - "data" term penalizing bad matches

$$D(x,y,d) = |\mathbf{I}(x,y) - \mathbf{J}(x+d,y)|$$

  - "neighborhood term" encouraging **spatial smoothness (continuity; disparity gradient)**

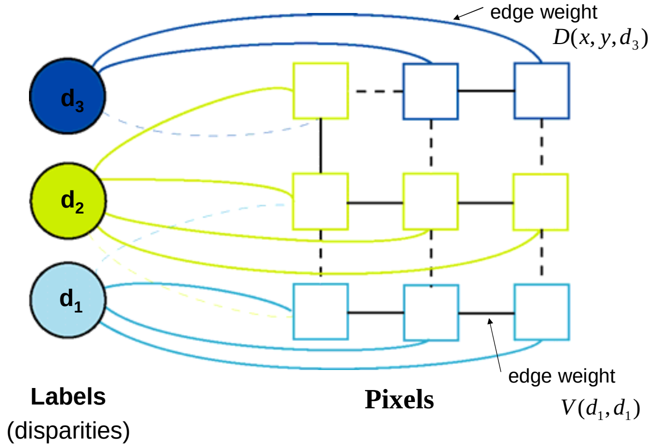
$$V(d_1,d_2) = \text{cost of adjacent pixels with labels d1 and d2}$$
$$= |d_1 - d_2| \quad \text{(or something similar)}$$

$$E = \sum_{(x,y)} D(x,y,d_{x,y}) + \sum_{neighbors\,(x1,y1),(x2,y2)} V(d_{x1,y1}, d_{x2,y2})$$

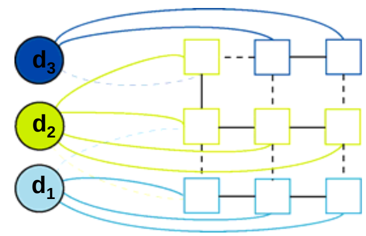
Minimization Methods

1. Continuous **d**: Gradient Descent
   - Gets stuck in local minimum
2. Discrete **d**: Simulated Annealing [Geman and Geman, PAMI 1984]
   - Takes forever or gets stuck in local minimum

Stereo as a Graph Problem [Boykov, 1999]

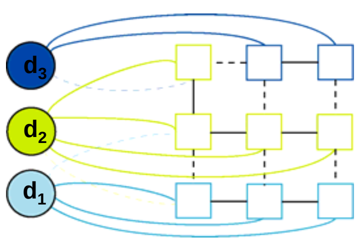


Graph Definition



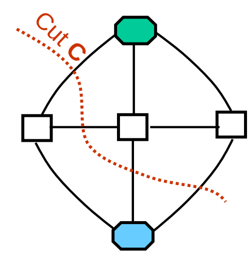
- Initial state
  - Each pixel connected to it's immediate neighbors
  - Each disparity label connected to all of the pixels

Stereo Matching by Graph Cuts



- Graph Cut
  - Delete enough edges so that
    - each pixel is (transitively) connected to exactly one label node
  - Cost of a cut: sum of deleted edge weights
  - Finding min cost cut equivalent to finding global minimum of the energy function

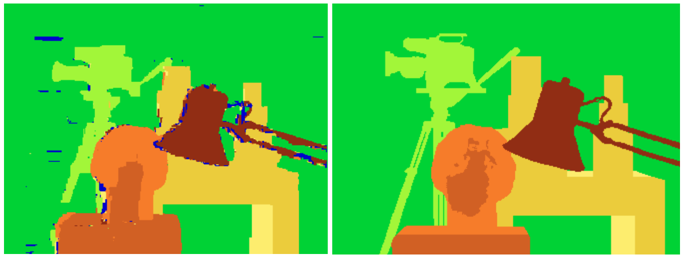
Graph Cuts



- Graph **G**=(**V**,**E**)
- Edge weight **w**: **E** → **R**$^+$
- Cost(**C**) = $\sum_{\text{edges in C}}$ **w**(edge)
- Problem: find min Cost cut

- Solved in polynomial time w/ min-cut/max-flow
- Boykov and Kolmogorov algorithm
  - runs in seconds

Results of Boykov's Graph Cut Algorithm



Results	Ground truth

Boykov et al., Fast Approximate Energy Minimization via Graph Cuts, *Proc. Int. Conf. Computer Vision*, 1999

Local: Compact Window | **Global: Expansion**

high texture

18 sec
16% error

10 sec
0.33% error

75 sec, $\lambda = 5$
16% error

33 sec, $\lambda = 100$
0.35% error

medium texture

12 sec, 3.36% error

32 sec, 1.86% error, $\lambda = 20$

Difficulties

- Parameter selection

smaller λ allows more discontinuities

$$E(d) = \sum_{p \in P} M(d_p) + \lambda \sum_{\{p,q\} \in N} \delta(d_p \neq d_q)$$

optimal $\lambda = 5$

optimal $\lambda = 20$

- Running time: from 34 to 86 seconds

Computing a Multi-way Cut

- With two labels: classical min-cut problem
  - Solvable by standard network flow algorithms
    - polynomial time in theory, nearly linear in practice
- More than 2 labels: NP-hard [Dahlhaus *et al.*, STOC '92]
  - But efficient approximation algorithms exist
    - Within a factor of 2 of optimal
    - Computes local minimum in a strong sense
      - even very large moves will not improve the energy
    - Y. Boykov, O. Veksler and R. Zabih, Fast Approximate Energy Minimization via Graph Cuts, *Proc. Int. Conf. Computer Vision*, 1999
  - Basic idea
    - reduce to a series of 2-way-cut sub-problems, using one of:
      - swap move: pixels with label L1 can change to L2, and vice-versa
      - expansion move: any pixel can change it's label to L1

State of the Art

left image

true disparities

Late 90's state of the art

5.23% errors

Recent state of the art

1.86% errors

Evaluation of Stereo Algorithms

http://bj.middlebury.edu/~schar/stereo/web/results.php

"A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Computer Vision*, 2002

Database by D. Scharstein and R. Szeliski

% errors

Algorithm	Tsukuba	Sawtooth	Venus	Map
Layered	1.58	0.34	1.52	0.37
Graph cuts	1.94	1.30	1.79	0.31
Belief prop.	1.15	0.98	1.00	0.84
GC+occl.	1.27	0.36	2.79	1.79
Graph cuts	1.86	0.42	1.69	2.39
Multiw. cut	8.08	0.61	0.53	0.26
Comp. win.	3.36	1.61	1.67	0.33
Realtime	4.25	1.32	1.53	0.81
Bay. diff.	6.49	1.45	4.00	0.20
Cooperative	3.49	2.03	2.57	0.22
SSD+MF	5.23	2.21	3.74	0.66
Stoch. diff.	3.95	2.45	2.45	1.31
Genetic	2.96	2.21	2.49	1.04
Pix-to-pix	5.12	2.31	6.30	0.50
Max flow	2.98	3.47	2.16	3.13
Scanl. opt.	5.08	4.06	9.44	1.84
Dyn. prog.	4.12	4.84	10.1	3.33
Shao	9.67	4.25	6.01	2.36
MMHM	9.76	4.76	6.48	8.42
Max. surf.	11.10	5.51	4.36	4.17

Multi-Baseline Stereo

- Okutomi and Kanade, 1991
- When images rectified,

  disparity $d = x_l - x_r = \frac{fb}{z}$

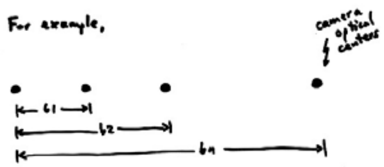
  where f= focal length, b= baseline length

  $\Rightarrow \boxed{\frac{d}{b} = \frac{f}{z}}$

- For fixed scene pt, $\frac{f}{z}$ = constant

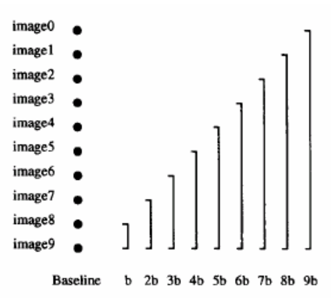
  $\Rightarrow$ take multiple images from cameras w/ varying b and combine them

For example,



The Effect of Baseline on Depth Estimation

Figure 2: An example scene. The grid pattern in the background has ambiguity of matching.



ALGORITHM

1. Edge Enhancement & Noise Suppression
   $\nabla^2 G$
   Implemented in hardware as 3 7×7 cascaded Gaussians and 1 7×7 Laplacian.
   $\Rightarrow$ approximates 25×25 $\nabla^2 G$ filter

2. Match and Combine
   Given: $n+1$ cameras, where one is called Base and other called Inspection, I_i
   Use n stereo pairs: (Base, I_i)

   2.1 Rectify
   Rectify each inspection image wrt Base by warping and resampling

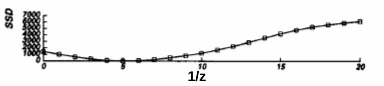
2.2 Match
For each pixel (i,j) in Base
  For each pixel (k,l) in epipolar line of (i,j) in I
    Compute SSD for $W \times W$ block of pixels centered on (i,j) in Base and (k,l) in I. I.e.,

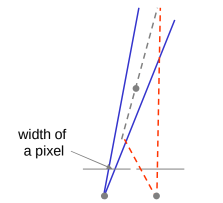
$$SSD_I(i,j,\mathcal{Z}) = \sum_{(s,t) \in W(i,j)} \Big(I\big(s + c_1 (\underbrace{b_I \mathcal{Z}}_{= d_I}),\ t + c_2 (b_I \mathcal{Z})\big) - \text{Base}(s,t) \Big)^2$$

where $z = (c_1, c_2)$ = unit vector in epipolar line direction in I

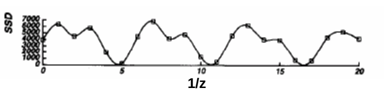
$$\mathcal{Z} = \frac{f_I}{z} = \frac{b_I}{d_I}$$

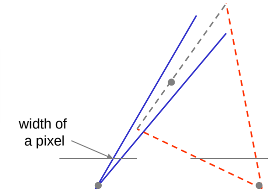
$$(\Rightarrow b_I \mathcal{Z} = d_I)$$





pixel matching score





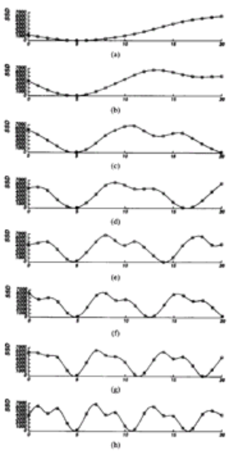


Fig. 5. SSD values versus inverse distance: (a) $B = b$; (b) $B = 2b$; (c) $B = 3b$; (d) $B = 4b$; (e) $B = 5b$; (f) $B = 6b$; (g) $B = 7b$; (h) $B = 8b$. The horizontal axis is normalized such that $8bF = 1$.

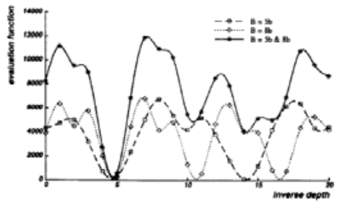


Fig. 6. Combining two stereo pairs with different baselines.

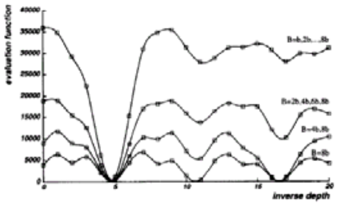
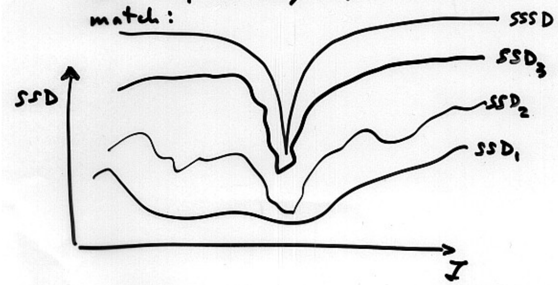


Fig. 7. Combining multiple baseline stereo pairs.

Result is for each inspection image and displacement, $\mathcal{I}$, a measure of match:



2.3 Combine Evidence

Sum SSD values:

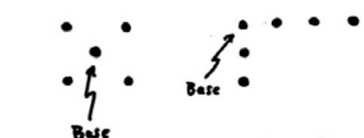
$$SSSD(i,j,\mathcal{I}) = \sum_{I} SSD_I(i,j,\mathcal{I})$$

3. Estimate Depth Map

Find value of $\mathcal{I}$ that minimizes SSSD: Fit quadratic function to data points and interpolate to estimate min $\mathcal{I}$.

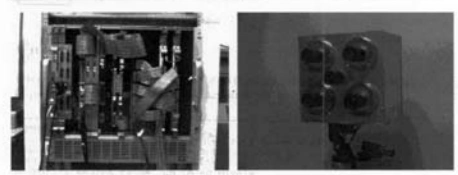
Depth $z = \mathcal{I}/f$ at each pixel.

- Camera Configurations Used



256 x 240 images
30 frames per second
disparity range 60 pixels

Video-Rate Stereo Machine



Stereo vision and multi-baseline method

Stereo ranging, which uses correspondence between sets of two or more images for depth measurement, has many advantages. It is passive and it does not emit any radio or light energy. With appropriate imaging geometry, optics, and high-resolution cameras, stereo can produce a dense, precise range image of even distant scenes. Our video-rate stereo machine is based on a new stereo technique which has been developed and tested at CMU over years. It uses multiple images obtained by multiple cameras to produce different baselines in lengths and in directions. The multi-baseline stereo method takes advantage of the redundancy contained in multi-stereo pairs, resulting in a straightforward algorithm which is appropriate for hardware implementation.

Real-Time Stereo

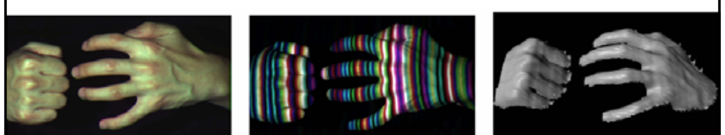
Nomad robot searches for meteorites in Antartica
http://www.frc.ri.cmu.edu/projects/meteorobot/index.html

- Used for robot navigation (and other tasks)
  - Several software-based real-time stereo techniques have been developed (most based on simple discrete search)

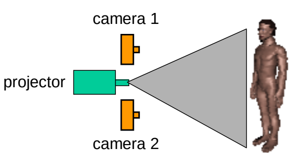
Stereo Reconstruction Pipeline

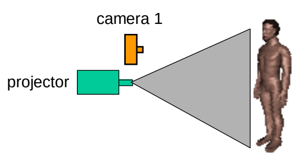
- Steps
  - Calibrate cameras
  - Rectify images
  - Compute disparity
  - Estimate depth
- What will cause errors?
  - Camera calibration errors
  - Poor image resolution
  - Occlusions
  - Violations of brightness constancy (specular reflections)
  - Large motions
  - Low-contrast image regions

Active Stereo with Structured Light



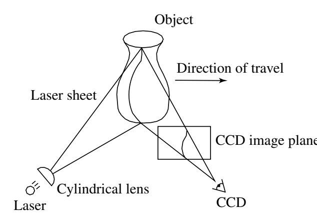
Li Zhang's one-shot stereo

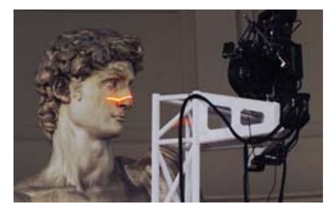




- Project "structured" light patterns onto the object
  - simplifies the correspondence problem

Laser Scanning





Digital Michelangelo Project
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

Portable 3D Laser Scanners

Minolta Vivid 910 can scan
300,000 points in 2.5 sec