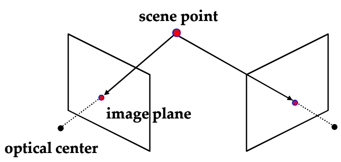
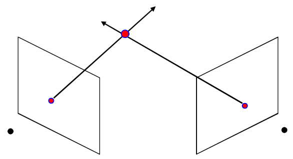
Stereo: 3D from Two Views



Stereo

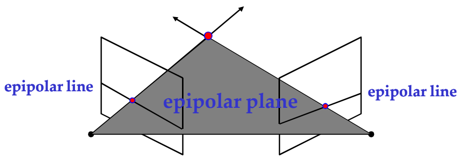


- Basic Principle: Triangulation
  - Gives reconstruction as intersection of two rays
  - Requires
    - calibration
    - ***point correspondence***

Stereo Correspondence

Determine Pixel Correspondence

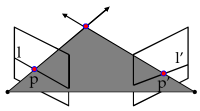
- Pairs of points that correspond to same scene point



- Epipolar Constraint
  - Reduces correspondence problem to 1D search along *conjugate epipolar lines*

Fundamental Matrix

Let *p* be a point in left image, *p'* in right image



Epipolar relation

- *p* maps to epipolar line *l'*
- *p'* maps to epipolar line *l*

Epipolar mapping described by a 3x3 matrix *F*

$$l' = Fp$$
$$l^T = p'^T F$$

It follows that

$$p'^T F p = 0$$

Fundamental Matrix

This matrix F is called

- the "Essential Matrix"
  - when image intrinsic parameters are known
- the "Fundamental Matrix"
  - more generally (uncalibrated case)

Can solve for F from point correspondences

- Each (p, p') pair gives one linear equation in entries of F

$$p'^T F p = 0$$

8 points give enough to solve for F (8-point algorithm)

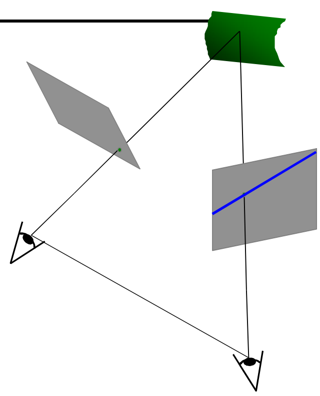
So far…

Compute F

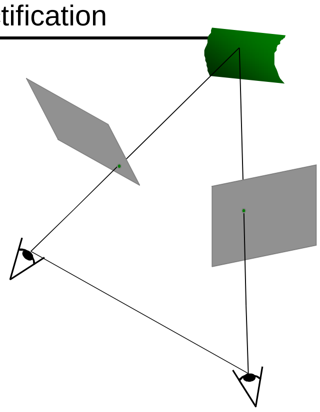
For each point

- Compute epipolar line using F
- Search along the epipolar line

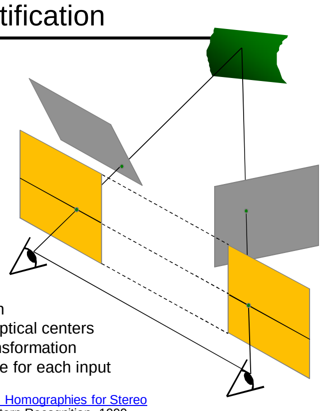
- But slanted epipolar lines are hard to search along!



Stereo Image Rectification

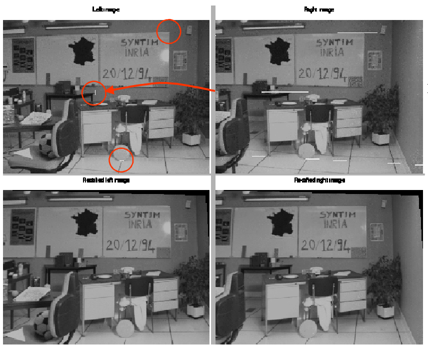


Stereo Image Rectification

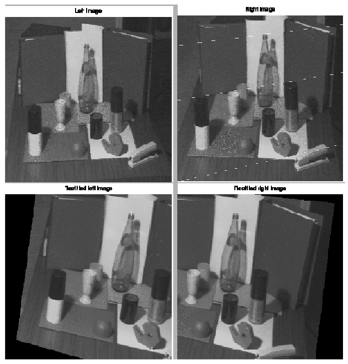


- reproject image planes onto a common plane parallel to the line between optical centers
- pixel motion is horizontal after this transformation
- two homographies (3x3 transform), one for each input image reprojection
  - C. Loop and Z. Zhang. Computing Rectifying Homographies for Stereo Vision. IEEE Conf. Computer Vision and Pattern Recognition, 1999.

Epipolar Rectified Images

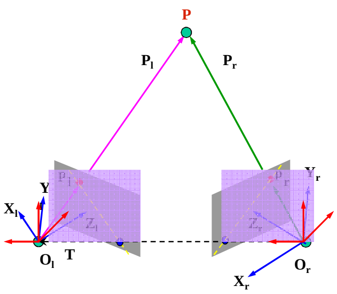


Epipolar Rectified Images



Source: A. Fusiello, Verona, 2000]

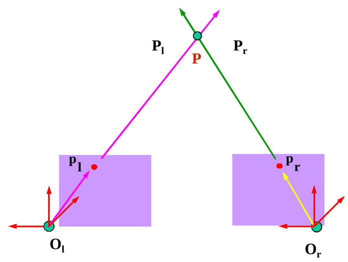
Stereo Rectification



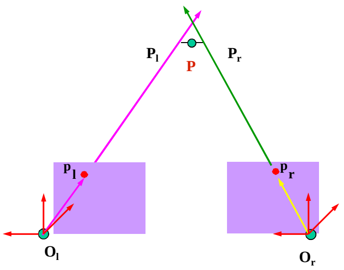
Stereo System with Parallel Optical Axes

- Epipoles are at infinity
- Horizontal epipolar lines

Reconstruction (3-D): Idealized



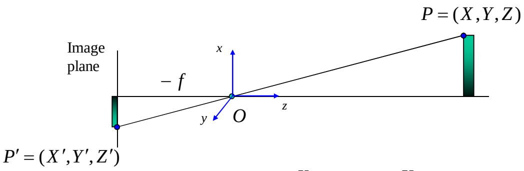
Reconstruction (3-D): Real



Summary Stereo Vision

- Epipolar Geometry: Corresponding points lie on epipolar line
- Essential/Fundamental matrix: Defines this line
- Eight-Point Algorithm: Recovers Fundamental Matrix
- Rectification by Homography: Epipolar lines parallel to scan lines
- Reconstruction: Find point correspondences

Pinhole Camera Model

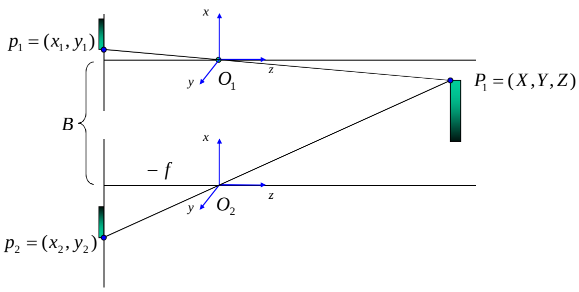


$$Z' = -f \quad X' = -f\frac{X}{Z} \quad Y' = -f\frac{Y}{Z}$$

$$x = -X' \quad y = -Y'$$

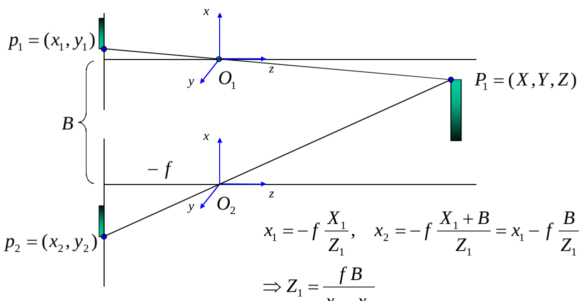
$$(X, Y, Z) \rightarrow (x, y) = (f\frac{X}{Z}, f\frac{Y}{Z})$$

Basic Stereo Derivations



Derive expression for Z as a function of x_1, x_2, f, B

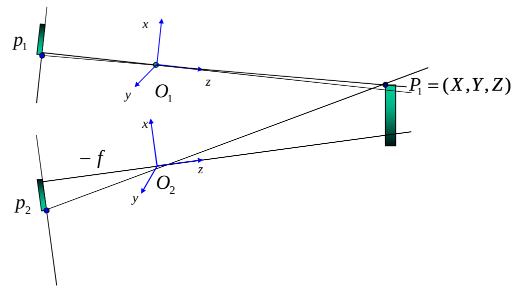
Basic Stereo Derivations



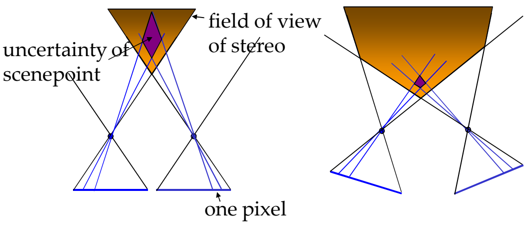
$$x_1 = -f\frac{X_1}{Z_1}, \quad x_2 = -f\frac{X_1 + B}{Z_1} = x_1 - f\frac{B}{Z_1}$$

$$\Rightarrow Z_1 = \frac{fB}{x_1 - x_2}$$

What If…?



Vergence



Optical axes of the two cameras need not be parallel

Field of view decreases with increase in baseline and vergence (the right image a bit deceptive)

Accuracy increases with baseline and vergence

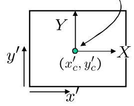
Stereo Matching Algorithms

Match Pixels on Epipolar Lines

- Assume brightness constancy
- Numerous approaches
  - dynamic programming [Baker 81,Ohta 85]
  - smoothness functionals
  - more images (trinocular, N-ocular) [Okutomi 93]
  - graph cuts [Boykov 00]
- A good survey and evaluation: http://www.middlebury.edu/stereo/

Camera Parameters

- A camera is described by several parameters
  - Translation T of the optical center from the origin of world coords
  - Rotation R of the image plane
  - focal length f, principle point (x'_c, y'_c), pixel size (s_x, s_y)
  - blue parameters are called "extrinsics," red are "intrinsics"
- Projection equation



$$\mathbf{x} = \begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{\Pi X}$$

  - The projection matrix models the cumulative effect of all parameters
  - Useful to decompose into a series of operations

$$\mathbf{\Pi} = \underbrace{\begin{bmatrix} -fs_x & 0 & x'_c \\ 0 & -fs_y & y'_c \\ 0 & 0 & 1 \end{bmatrix}}_{\text{intrinsics}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{projection}} \underbrace{\begin{bmatrix} \mathbf{R}_{3x3} & \mathbf{0}_{3x1} \\ \mathbf{0}_{1x3} & 1 \end{bmatrix}}_{\text{rotation}} \underbrace{\begin{bmatrix} \mathbf{I}_{3x3} & \mathbf{T}_{3x1} \\ \mathbf{0}_{1x3} & 1 \end{bmatrix}}_{\text{translation}}$$

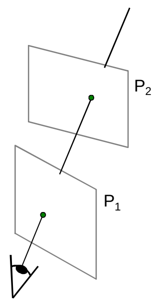
identity matrix

Homography

Projective mapping between any two planes with the same center of projection

- Rectangles are mapped to arbitrary quadrilateral
- Parallel lines: not preserved
- Straight lines: Preserved
- Same as: project, rotate, reproject

$$\underbrace{\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix}}_{\mathbf{p'}} = \underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\mathbf{p}}$$



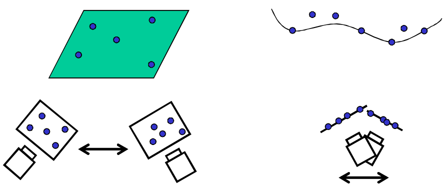
To apply a homography **H**

- Compute **p' = Hp** (regular matrix multiply)
- Convert **p'** from homogeneous to image coordinates

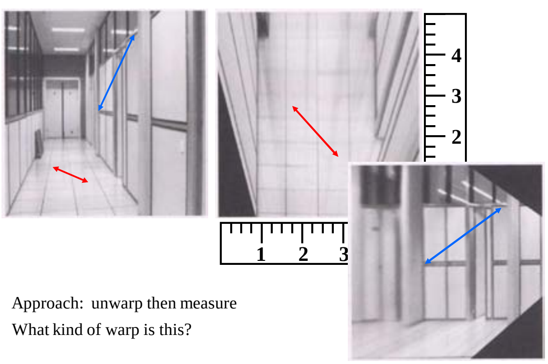
Homography

Homography is a singular case of the Fundamental Matrix

- Two views of **coplanar points**
- Two views that **share the same center of projection**



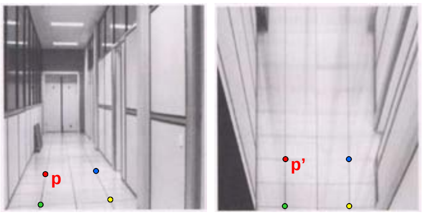
Measurements on Planes



Approach: unwarp then measure

What kind of warp is this?

Image Rectification



To unwarp (rectify) an image

- solve for homography **H** given **p** and **p'**
- solve equations of the form: w**p'** = **Hp**
  - linear in unknowns: w and coefficients of **H**
  - H is defined up to an arbitrary scale factor
  - how many points are necessary to solve for **H**?

Solving for Homographies

$$\begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} \cong \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

$$x'_i = \frac{h_{00}x_i + h_{01}y_i + h_{02}}{h_{20}x_i + h_{21}y_i + h_{22}}$$

$$y'_i = \frac{h_{10}x_i + h_{11}y_i + h_{12}}{h_{20}x_i + h_{21}y_i + h_{22}}$$

$$\begin{aligned} x'_i(h_{20}x_i + h_{21}y_i + h_{22}) &= h_{00}x_i + h_{01}y_i + h_{02} \\ y'_i(h_{20}x_i + h_{21}y_i + h_{22}) &= h_{10}x_i + h_{11}y_i + h_{12} \end{aligned}$$

$$\begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x'_i x_i & -x'_i y_i & -x'_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -y'_i x_i & -y'_i y_i & -y'_i \end{bmatrix} \begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

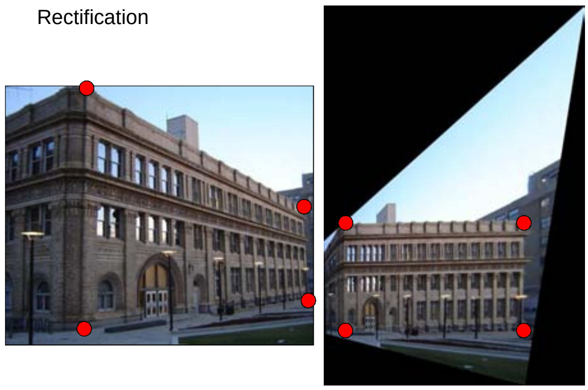
Solving for Homographies

$$\underbrace{\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1 x_1 & -x'_1 y_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1 x_1 & -y'_1 y_1 & -y'_1 \\ & & & & \vdots & & & & \\ x_n & y_n & 1 & 0 & 0 & 0 & -x'_n x_n & -x'_n y_n & -x'_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -y'_n x_n & -y'_n y_n & -y'_n \end{bmatrix}}_{\mathbf{A}\ (2n \times 9)} \underbrace{\begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix}}_{\mathbf{h}\ (9)} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\mathbf{0}\ (2n)}$$

- Total least squares
  - Since **h** is only defined up to scale, solve for unit vector $\hat{\mathbf{h}}$
  - Minimize $\|\mathbf{A}\hat{\mathbf{h}}\|^2$
    $$\|\mathbf{A}\hat{\mathbf{h}}\|^2 = (\mathbf{A}\hat{\mathbf{h}})^T\mathbf{A}\hat{\mathbf{h}} = \hat{\mathbf{h}}^T\mathbf{A}^T\mathbf{A}\hat{\mathbf{h}}$$
  - Solution: $\hat{\mathbf{h}}$ = eigenvector of $\mathbf{A}^T\mathbf{A}$ with smallest eigenvalue
  - Works with 4 or more points (more points more accurate)

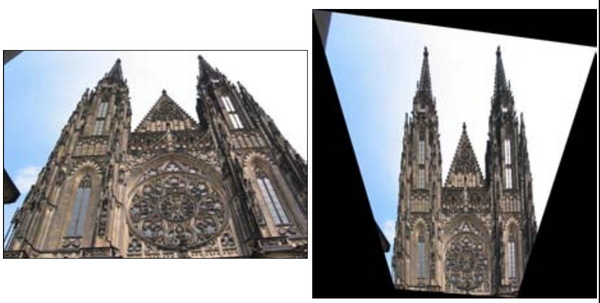
Example

Rectification

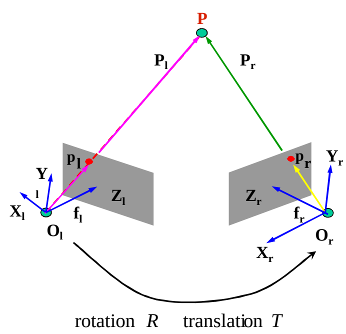


Example

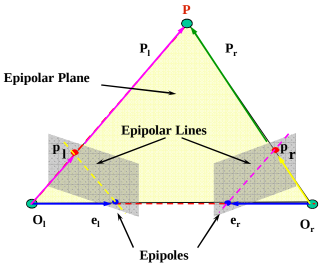
Rectification



Epipolar Geometry



Epipolar Geometry



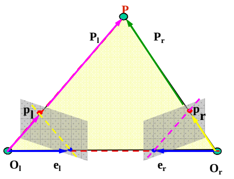
Epipolar Geometry

Epipolar plane: plane going through point P and the centers of projection (COPs) of the two cameras

Epipoles: The image in one camera of the COP of the other

Epipolar Constraint: Corresponding points must lie on epipolar lines

Essential Matrix



$$T \times P_l = SP_l$$

$$S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}$$

Coordinate Transformation: $P_r = R(P_l - T)$

Coplanarity T, P_l, $P_l - T$: $(P_l - T)^T (T \times P_l) = 0$

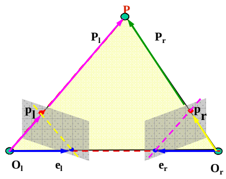
Resolves to $(R^T P_r)^T (T \times P_l) = 0$

$$(R^T P_r)^T SP_l = 0$$

$$P_r^T RSP_l = 0$$

Essential Matrix $E = RS$ $\quad P_r^T EP_l = 0$

Essential Matrix



$$P_r^T EP_l = 0 \quad \Rightarrow p_r^T Ep_l = 0$$

Projective Line: $u_r = Ep_l$

Essential Matrix $E = RS$ $\quad S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}$

Fundamental Matrix

Same as Essential Matrix in Camera Pixel Coordinates

$$p_r^T Ep_l = 0$$

$$\Downarrow$$

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

Pixel coordinates $\quad F = M_r^{-1T} EM_l^{-1}$

Intrinsic parameters

Intrinsic Parameters

$$M = \begin{pmatrix} -f/s_x & 0 & o_x \\ 0 & -f/s_y & o_y \\ 0 & 0 & 1 \end{pmatrix}$$

Computing F: The Eight-Point Algorithm

Problem: Recover F (3-3 matrix of rank 2)

Idea: Get 8 points:

$$\bar{\mathbf{p}}_\mathbf{r}(1)^\mathrm{T}\mathbf{F}\bar{\mathbf{p}}_\mathbf{l}(1) = 0$$
$$\vdots$$
$$\bar{\mathbf{p}}_\mathbf{r}(8)^\mathrm{T}\mathbf{F}\bar{\mathbf{p}}_\mathbf{l}(8) = 0$$

Minimize:

$$\operatorname*{argmin}_F \sum_{i=1}^{8} \left(\bar{\mathbf{p}}_\mathbf{r}(i)^\mathrm{T}\mathbf{F}\bar{\mathbf{p}}_\mathbf{l}(i)\right)^2$$

Notice: Argument *linear* in coefficients of F

Computing F: The Eight-Point Algorithm

Idea: Compile points into matrix A

$$\begin{pmatrix} & & \\ & A & \\ & & \end{pmatrix} \begin{pmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\bar{\mathbf{p}}_\mathbf{r}(i)^\mathrm{T}\mathbf{F}\bar{\mathbf{p}}_\mathbf{l}(i) = 0$$

Computing F: The Eight-Point Algorithm

Run Singular Value Decomposition of A

- Appendix A.6, page 322-325
- See also G. Strang: Linear algebra and its applications

$$Ax = 0$$

$$A = UDV^T \qquad \text{via SVD}$$

Least squares solution: column of V corresponding to the smallest eigenvalue of A

Computing F: The Eight-Point Algorithm

Decompose A via SVD: $A = UDV^T$

Solution: F is column of V corresponding to the smallest eigenvector of A

In practice: F will be of rank 3, not 2. Correct by

- SVD decomposition of F
- Set smallest eigenvalue to 0
- Reconstruct F'

$$F = U'D'V'^T$$
$$D'' = \begin{pmatrix} D'(1) & & D'(2) & & 0 \end{pmatrix}$$
$$F = U'D''V'^T$$

Recitification

Idea: Align Epipolar Lines with Scan Lines.

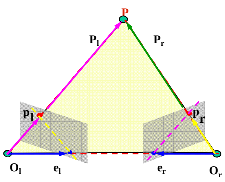
Question: What type transformation?

Locating the Epipoles

$$\bar{\mathbf{p}}_\mathbf{r}{}^\mathrm{T}\mathbf{F}\bar{\mathbf{p}}_\mathbf{l} = 0$$
$$\Downarrow$$
$$\bar{\mathbf{p}}_\mathbf{r}{}^\mathrm{T}\mathbf{F}\bar{\mathbf{e}}_\mathbf{l} = 0$$

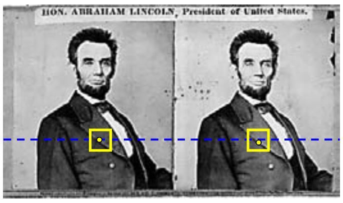
e_l lies on all the epipolar lines of the left image

$$\Downarrow$$
$$\mathbf{F}\bar{\mathbf{e}}_\mathbf{l} = 0$$



- Input: Fundamental Matrix F
  - Find the SVD of F $\quad \mathbf{F} = \mathbf{UDV}^T$
  - The epipole e_l is the column of V corresponding to the null singular value (as shown above)
  - The epipole e_r is the column of U corresponding to the null singular value (similar treatment as for e_l)
- Output: Epipole e_l and e_r

Basic Stereo Algorithm



For each epipolar line

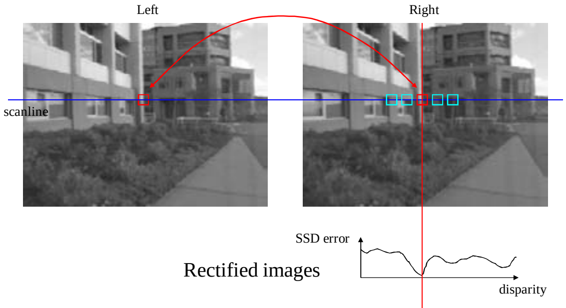
For each pixel in the left image

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

Improvement: match ***windows***

- This should look familar...
- Correlation, Sum of Squared Difference (SSD), etc.

Correspondence via Correlation



(Same as max-correlation / max-cosine for normalized image patch)

Image Metrics

(Normalized) Sum of Squared Differences

$$C_{\text{SSD}}(d) = \sum_{(u,v)\in W_m(x,y)} [\hat{I}_L(u,v) - \hat{I}_R(u-d,v)]^2$$

$$= \|w_L - w_R(d)\|^2$$

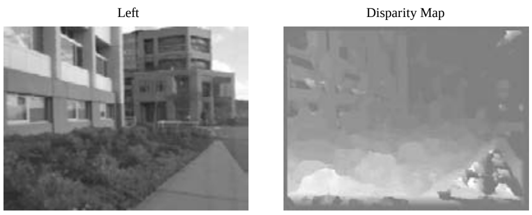
Normalized Correlation

$$C_{\text{NC}}(d) = \sum_{(u,v)\in W_m(x,y)} \hat{I}_L(u,v)\hat{I}_R(u-d,v)$$

$$= w_L \cdot w_R(d)$$

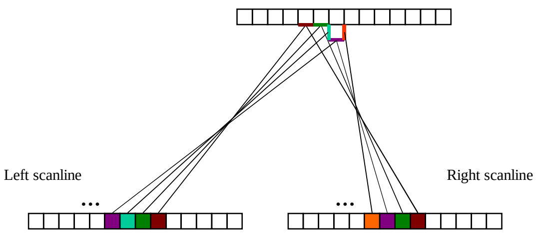
$$d^* = \arg\min_d \|w_L - w_R(d)\|^2 = \arg\max_d w_L \cdot w_R(d)$$

Correspondence Using Correlation

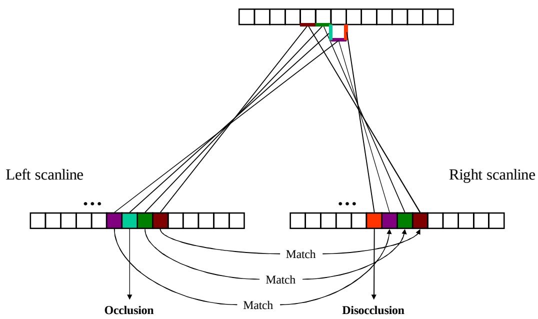


Images courtesy of Point Grey Research

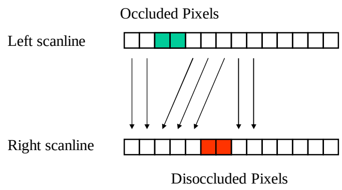
Stereo Correspondences



Stereo Correspondences



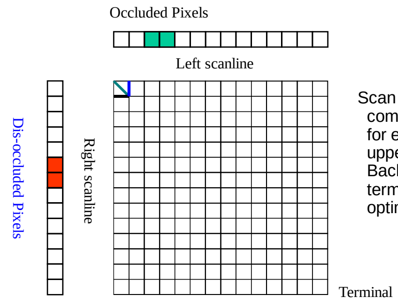
Search Over Correspondences



Three cases:

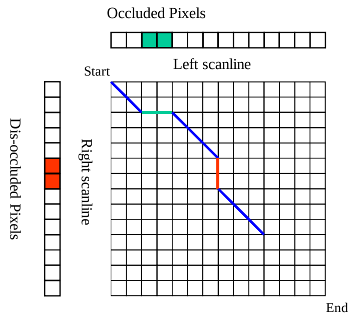
- Sequential – cost of match
- Occluded – cost of no match
- Disoccluded – cost of no match

Stereo Matching with Dynamic Programming



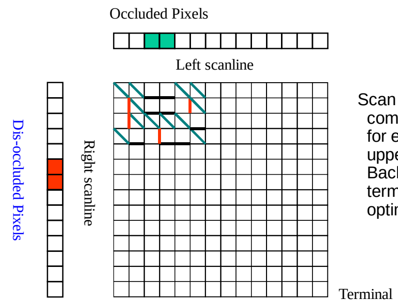
Scan across grid computing optimal cost for each node given its upper-left neighbors. Backtrack from the terminal to get the optimal path.

Stereo Matching with Dynamic Programming



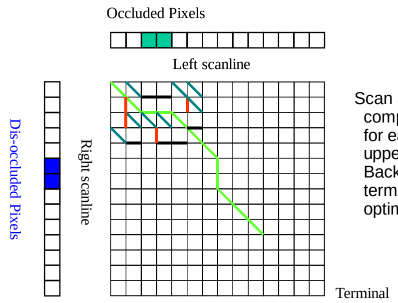
Dynamic programming yields the optimal path through grid. This is the best set of matches that satisfy the ordering constraint

Stereo Matching with Dynamic Programming



Scan across grid computing optimal cost for each node given its upper-left neighbors. Backtrack from the terminal to get the optimal path.

Stereo Matching with Dynamic Programming

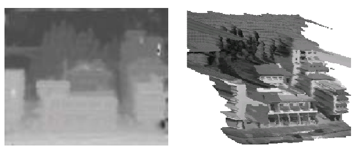


Scan across grid computing optimal cost for each node given its upper-left neighbors. Backtrack from the terminal to get the optimal path.

Dense Stereo Matching: Examples

View extrapolation results

input

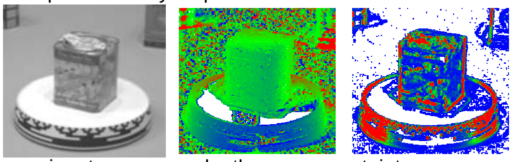


depth image   novel view

[Matthies,Szeliski,Kanade'88]

Dense Stereo Matching

Compute certainty map from correlations



input depth map certainty map

DP for Correspondence

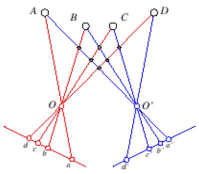
Does this always work?

When would it fail?

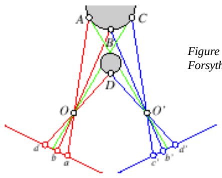
- Failure Example 1
- Failure Example 2
- Failure Example 3

Correspondence Problem 1

It is fundamentally ambiguous, even with stereo constraints



Ordering constraint…

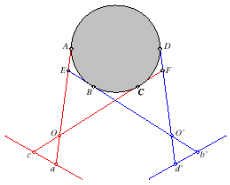


…and its failure

Figure from Forsyth & Ponce

Correspondence Problem 2

Correspondence fail for smooth surfaces



There is currently no good solution to the correspondence problem

Correspondence Problem 3

Regions without texture

Highly Specular surfaces

Translucent objects

Window size

W = 3 W = 20

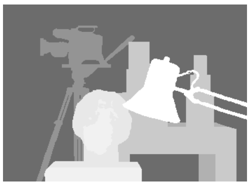
Effect of window size

- Smaller window
  - +
  - –
- Larger window
  - +
  - –

Stereo results

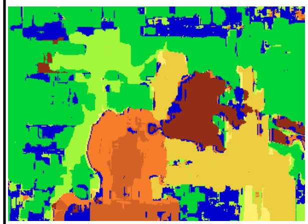
- Data from University of Tsukuba
- Similar results on other images without ground truth

Scene



Ground truth

Results with window search



Window-based matching
(best window size)

Ground truth

Better methods exist...

State of the art method

Boykov et al., Fast Approximate Energy Minimization via Graph Cuts, International Conference on Computer Vision, September 1999.

Ground truth

Stereo Example

left image

right image

depth map

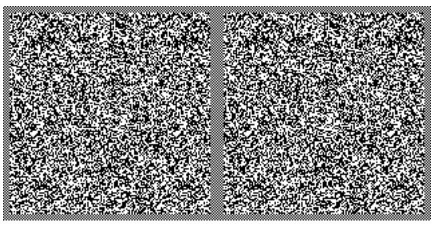
Stereo reconstruction pipeline

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

Stereo matching

Need texture for matching



Julesz-style Random Dot Stereogram