Binocular Stereo

- Take 2 images from different known viewpoints \( \Rightarrow 1^{\text{st}} \) calibrate
- 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
Photogrammetry

- Important application of Stereo.

- Recover 3-D terrain from sequence of overlapping images.

- Relative positions of plane ($T_i$) must be known.

Figure 3.11. Stereo pair of the Pentagon.
Stereo

- Basic Principle: Triangulation
  - Gives reconstruction as intersection of two rays
  - Requires
    - calibration
    - point correspondence
Substituting and simplifying we set:

\[ X = \frac{Tx}{x'_2 - x_2'} \quad Y = \frac{Ty}{x'_2 - x_2'} \]

\[ Z = \frac{Tz}{x'_2 - x_2'} \]

\[ d = x'_2 - x_2 \quad \text{(horizontal disparity)} \]

\[ \Rightarrow Z = \frac{f}{d} \]

- \( d = 0 \Rightarrow P \text{ at infinity} \)
- Large \( d \Rightarrow P \text{ close to cameras} \)
- \( Z \) 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 when images are less similar

**Depth from Disparity**

input image (1 of 2)  depth map  3D rendering

[Szeliski & Kang '95]

\[
\text{disparity} = u - u' = \frac{\text{baseline} \cdot f}{z}
\]
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: 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**

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 $O$ 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 $O$, $O'$ 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 $O_l O_r$. 
  $⇒$ 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 $O_l O_r$ with the left and right images $I_l$ and $I_r$, respectively.
- If \( I_\ell \) and \( I_r \) are parallel, the epipoles are at infinity, and the epipolar lines are parallel.

- Given a pair of images, \( I_\ell \) and \( I_r \), the warping transformation that projects \( I_\ell \) and \( I_r \) onto a plane parallel to \( O_\ell O_r \) is called **rectification**. Usually done so that epipolar lines are parallel to new images' horizontal axes.
  \[ \Rightarrow \text{Epipolar constraint} = \text{1D search along image scanline} \]

- **Epipolar Constraint**: The correct match for a point \( p_\ell \) is constrained to a 1D search along the epipolar line in \( I_r \).

- All epipolar planes defined by all points in \( I_\ell \) contain the line \( O_\ell O_r \).
  \[ \Rightarrow \text{All epipolar lines in } I_r \text{ intersect at a point, } c_r, \text{ called the epipole} \]

- Left and right epipoles defined by intersection of line \( O_\ell O_r \) with \( I_\ell \) and \( I_r \), respectively.
Epipolar Geometry

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

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

\[
P = [X, Y, Z]^T
\]

\[
P_l = [X_l, Y_l, Z_l]^T
\]

Define \( P \) as a vector with left and right camera coordinate frames.

Let

\[
l = [z, x, y]^T
\]

\[
l_r = [x_r, y_r, z_r]^T
\]

be projections of \( P \) into left and right images.

⇒ Relation \( l = l_r \) is:

\[
\begin{bmatrix}
X_l \\
Y_l
\end{bmatrix}
= \begin{bmatrix}
R_{11} & R_{12} & R_{13} \\
R_{21} & R_{22} & R_{23}
\end{bmatrix}
\begin{bmatrix}
X
\\
Y
\end{bmatrix}
- \begin{bmatrix}
l_z \\
l_x
\end{bmatrix}
\]

i.e.,

\[
P = R(p - T)
\]

where \( T = (O_r - O_l) \)

\[
R^T R = R^T R = I \quad \text{(Rinv)}
\]
**Coplanarity constraint**

Vectors \( p_x \) (defined by \( O_x, P \))

\[
T \quad \text{(defined by } O_x, O_r) \]

and \( p_x - T \) (defined by \( O_r, P \))

are coplanar.

\[
\Rightarrow \text{mixed product} = 0 \]

\[
(p_x - T)^T (p_x - T) = 0
\]

Since \( p_x = R(p_x - T) \quad \Rightarrow R = R^T \)

\[
(R^T p_x)^T T = p_x = 0
\]

Cross product of vectors can be rewritten as:

\[
T \times p_x = \mathbf{S} p_x = [T]_x p_x
\]

where \( \mathbf{S} = \begin{bmatrix} 0 & -T_y & T_x \\ T_y & 0 & -T_x \\ -T_x & T_y & 0 \end{bmatrix} \quad \text{(rank 2)} \)

\[
T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}
\]
Notation and Definitions

- Given \( a = (a_1, a_2, a_3)^T \)
  then \( [a]_x = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \)
  a 3x3 skew-symmetric matrix and singular matrix.
- Given 2 3-vectors, \( a \) and \( b \),
  \( a \times b \equiv \mathbf{a} \wedge \mathbf{b} \equiv [a]_x b \equiv (a^T [b])^T \)

Substituting we get:

\[ P_t E P_x = 0 \]

Epipolar Equation

where \( E = RS \)

\( E \) called Essential Matrix

Using perspective projection equation, can also show

\[ P_t E P_x = 0 \]

Corresponding for image points \( p_x \) and \( p_y \) in

\( I_x \) and \( I_y \), respectively, though

\( p_x \) and \( p_y \) are in camera coordinates.

But points known in image coordinates

\( \Rightarrow \) need to know transformation

from camera coords to image coords,

i.e., intrinsic parameters of camera

camera (focal length, camera shift, etc.)
- Equivalently, epipolar equation can be written as:

\[
P_r^T E_p r = 0
\]

where \( E = [T_{12}] R^T \)

- Observation: \( E_p \) can be interpreted as the vector representing the epipolar line associated with \( P_2 \) in the right image. So, the epipolar equation, \( P_r^T E_p r = 0 \) expresses the fact that \( P_2 \) lies on the epipolar line associated with the vector \( E_p \).

\( E = [T_{12}] R \) (is usual way of writing \( E \))

is a 3x3 matrix, 5 DoFs

is 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, defined in terms of camera coordinates.

- If \( \overline{P}_2 \) and \( \overline{P}_r \) are corresponding points in image (pixel) coordinates, then it can be shown:

\[
\overline{P}_r^T F \overline{P}_2 = 0
\]

where

\[
F = R_r^T E M_{12}^{-1} \quad \text{FUNDAMENTAL MATRIX}
\]

\[
M_{12} = \begin{pmatrix}
-\frac{1}{u_2} & 0 & 0 \\
0 & -\frac{1}{v_2} & 0 \\
0 & 0 & 1
\end{pmatrix}
\]

Intrinsic parameters of left camera:

\( f_x \) : focal length

\( (u_x, v_x) \) : coords of principal pt.

\( (e_x, f_x) \) : effective pixel size
Fundamental Matrix Properties

- It encodes info about both intrinsic and extrinsic params
  - If you can estimate F, you can reconstruct the epipolar geometry with no info about cameras
  - Has rank 2
  - 3x3
  - Has 7 dofs
  - Can be estimated from at least 8 correspondences
    - "The 8-point algorithm"
      - Each point gives a homogeneous linear equation of form \( p_i^T F p_j = 0 \)
      - Homogeneous system has solution unique up to scale factor
    - Usually use > 8 points so system is overdetermined and solve using SVD
Computing the Fundamental Matrix

Given a correspondence \( x \leftrightarrow x' \)
we know \( x' F x = 0 \)
or
\( x' f_{11} + x' y f_{12} + x' y' f_{13} + \cdots + f_{33} = 0 \)
or
\[
\begin{pmatrix}
x \\
y \\
y' \\
x' \\
y' \\
1
\end{pmatrix}
\begin{pmatrix}
f_{11} \\
f_{12} \\
f_{13} \\
f_{21} \\
f_{22} \\
f_{23} \\
f_{31} \\
f_{32} \\
f_{33}
\end{pmatrix} = 0
\]

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

\( p_r^T F p_2 = 0 \)

\[
\begin{pmatrix}
u' \\
v
\end{pmatrix}
\begin{pmatrix}
f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \\ 1
\end{pmatrix}
\begin{pmatrix}
u' \\
v
\end{pmatrix} = 0
\]

\[
\begin{pmatrix}
u u' \\
u v' \\
u u \\
v u \\
u v \\
u v' \\
u v \\
u v' \\
u v \\
u v'
\end{pmatrix}
\begin{pmatrix}
f_{11} \\
f_{12} \\
f_{13} \\
f_{21} \\
f_{22} \\
f_{23} \\
f_{31} \\
f_{32} \\
f_{33}
\end{pmatrix} = 0
\]

Homogeneous equation in coefficients of \( F \)
Given \( n \) conjugate pairs

\[
Af = \begin{bmatrix}
 x_1 x_2 & x_1 y_2 & \cdots & x_1 y_n & 1 \\
 x_2 x_3 & x_2 y_3 & \cdots & x_2 y_n & 1 \\
 \vdots & \vdots & \ddots & \vdots & \vdots \\
 x_n x_{n+1} & x_n y_{n+1} & \cdots & x_n y_n & 1 \\
\end{bmatrix}
\begin{bmatrix}
 f_{11} \\
f_{12} \\
f_{22} \\
\end{bmatrix} = 0
\]

- Solution determined up to scale only
  \( \Rightarrow \) need at least \( 2 \) point correspondences

- 8 points \( \Rightarrow \) unique solution
  \( \geq 9 \) pts \( \Rightarrow \) least-squares solution

1. Form equations \( Af = 0 \)
2. Take SVD: \( A = UDV^T \)
3. Solution is last column of \( V \)
   (eigenvector corresponds 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
  \[ \Rightarrow \text{translate and scale points origin to centroid of pts, scale so \"average pt\" is } (1,1,1) \]

**Normalized 8-point Algorithm**

1. **Normalization**
   \[ \hat{x}_i = T x_i \]
   \[ \hat{x}'_i = T' x'_i \]
2. **Linear Solution**
   Compute \( F \) by solving \( AF = 0 \)
3. **Singularity Constraint**
   Find closest singular \( F' \) to \( F \)
4. **Denormalization**
   \[ F = T' F' T \]
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

Stereo Correspondence

\[ \text{disparity} = x_1 - x_2 \] 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
  - EPISPOLAR 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 $\Delta$ viewpoint, $\Delta$ illumination, $\Delta$ surface orientation

- At depth discontinuities, edges do not remain fixed on surface
- Produces only a sparse reconstruction
Difficulties in Stereo Correspondence

Perfect case: never happens!

1) Image noise:

2) Low texture:
Local Approach

- Look at one image patch at a 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?

- **local** works for high texture
  - enough texture in a patch to disambiguate
- **global** works up to medium texture
  - propagates estimates from textured to untextured regions
- **salient regions** work up to low texture
  - propagation fails; some regions are inherently ambiguous, match only unambiguous regions
Local Approach [Levine’73]

\[ d_p = i \text{ which gives best } C_i \]

Common \( C \) (SSD)

\[ C = \sqrt{\left( \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2 \right)} \]

Fixed Window Size Problems

need different window shapes

left image

true disparities

fixed small window

fixed large window
Window Size

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

Better results with *adaptive window*


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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Tsukuba</th>
<th>Venus</th>
<th>Sawtooth</th>
<th>Map</th>
</tr>
</thead>
<tbody>
<tr>
<td>All global</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Layered</td>
<td>1.58</td>
<td>1.52</td>
<td>0.34</td>
<td>0.37</td>
</tr>
<tr>
<td>Graph cuts</td>
<td>1.94</td>
<td>1.79</td>
<td>1.30</td>
<td>0.31</td>
</tr>
<tr>
<td>Belief prop</td>
<td>1.15</td>
<td>1.00</td>
<td>0.98</td>
<td>0.84</td>
</tr>
<tr>
<td>GC+occl.</td>
<td>1.27</td>
<td>2.79</td>
<td>0.36</td>
<td>1.79</td>
</tr>
<tr>
<td>Graph cuts</td>
<td>1.86</td>
<td>1.69</td>
<td>0.42</td>
<td>2.39</td>
</tr>
<tr>
<td>Multiw. Cut</td>
<td>8.08</td>
<td>0.53</td>
<td>0.61</td>
<td>0.26</td>
</tr>
<tr>
<td>Veksler's var.</td>
<td>3.36</td>
<td>1.67</td>
<td>1.61</td>
<td>0.33</td>
</tr>
</tbody>
</table>
Constraints

1) corresponding pixels should be close in color

![Similar color pixels](image)

2) most nearby pixels should have similar disparity

![Disparity continuity](image)

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
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

$\Rightarrow$ 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

---

Left Image  
Right Image
Types of Stereo Algorithms

1. Local Methods based on Correlation
   - Normalized cross-correlation or SSD match using
   - mean window centered on each point
   - Compute dense depth map

2. Global Optimization
   - Define an energy function
     \[ E(f) = E_{\text{smooth}}(f) + E_{\text{data}}(f) \]
     where \( f \) is the disparity value at a given pixel, \( p \).

   - Example:
     \[ E_{\text{data}} = \sum_\mathbf{p} [I(p) - I(p + \text{disparity}(p))]^2 \]
     \[ E_{\text{smooth}} = \sum_\mathbf{p} \text{smoothness (of adjacent pixels of different disparity from } p) \]

   - \( E_{\text{smooth}} \) 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
   - Computevdense depth map
Marr-Poggio Stereo Algorithm

1. Convolve 2 rectified images with $\nabla^2_G^t$ filters of size $s_1 < s_2 < s_3 < s_4$

2. Detect zero-crossings in all images

3. At coarsest scale, $s_4$, match zero-crossings with same parity and roughly same orientation in a $[-W_r, +W_r]$ disparity range with $W_r = 2 \sigma_4$

4. Use disparities found at coarser scales to cause unmatched regions at finer scales to come into correspondence.

⇒ Result is a sparse depth map

---

2 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)

Inhibition Area

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

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

3. Iteratively update match values
   until match values converge.
   \(L_{n+1}(r, c, d) = L_n(r, c, d) + R_n(r, c, d)\)
   where
   \(R_n(r, c, d) = \frac{1}{\sum_{i=1}^{N} S_n(r', c', d')} \sum_{i=1}^{N} S_n(r', c', d') \cdot \omega(r', c', d')\)
and where

\[ S_n(r,s,d) = \sum_{\phi \in R} \log(r+r', c+c', d+d') \]

\( \alpha > 1 \)

\( \phi \) corresponds to smoothness assumption

\( \psi \) corresponds to uniqueness assumption

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

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

\( \times \) converges to 1 at correct matches

\( \times \) to prevent over-smoothing & loss of detail

\( L_0 \times R_n \) means only pairs with similar initial intensities will contribute to match value computation
Figure 6(c) are reused while recovering several details at the same time. The shaded roof of the lower building and the windows near the edge are clearly visible. Depth information is preserved. 25 iterations were used and the inhibition constant was set to 2.

Figure 3: Synthetic scene, NMI (a) Reference (b) Left image (c) Right image (d) The Disparity map. Black areas are occluded. (e) Disparity map kernel using size 3x3 (f) Final support area, black areas are detected occlusions.

Random Occlusion Results

<table>
<thead>
<tr>
<th>Local Support Size</th>
<th>% Support</th>
<th>% Disparity</th>
<th>% Occlusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>2x2x0</td>
<td>99.88</td>
<td>91.51</td>
<td>19.49</td>
</tr>
<tr>
<td>5x5x0</td>
<td>99.29</td>
<td>85.41</td>
<td>14.59</td>
</tr>
<tr>
<td>7x7x0</td>
<td>98.72</td>
<td>91.10</td>
<td>8.90</td>
</tr>
</tbody>
</table>

Table 1: The percentage of disparity kernel correctly, the percentage of the detected occlusion that are correct and the percentage of the false occlusion found for three different local support area sizes using the maximum-scale match.

Figure 7: Convergence rate for inhibition constant of 3, 5, and 8 over 50 iterations using the maximum post-processing.

Figure 5: One image provided for Columbia (a) Reference (b) Left image (c) Right image (d) Observed self-disparity map with black areas occluded. Provided contours of the object (e) Disparity map kernel using size 3x3 (f) Final support area, black areas are detected occlusions. The match values were distorted in completely occlusion. Disparity values for those object such as the lamp areas are included correctly.
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 Tokushima stereo pair.

<table>
<thead>
<tr>
<th>Local Support Area</th>
<th>% Disparity Correct</th>
<th>% Occlusion Correct</th>
<th>% Occlusion Found</th>
</tr>
</thead>
<tbody>
<tr>
<td>3x3</td>
<td>97.12</td>
<td>48.50</td>
<td>80.16</td>
</tr>
<tr>
<td>5x5</td>
<td>98.02</td>
<td>99.58</td>
<td>51.84</td>
</tr>
<tr>
<td>7x7</td>
<td>97.73</td>
<td>63.23</td>
<td>44.85</td>
</tr>
</tbody>
</table>

Table 3: The number of occluded and non-occluded pixels found using our algorithm compared to the ground truth data provided by University of Tokushima. A 3x3 area was used for the local support and the disparity values were allowed to completely converge.

<table>
<thead>
<tr>
<th></th>
<th>Ground Truth Occluded</th>
<th>Ground Truth Non-Occluded</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Occluded</td>
<td>960</td>
<td>285</td>
<td>1245</td>
</tr>
<tr>
<td>Non-Occluded</td>
<td>1,002</td>
<td></td>
<td>8,769</td>
</tr>
</tbody>
</table>

Table 4: Comparison of various algorithms using the ground truth data supplied by University of Tokushima. Error rates of greater than one pixel in disparity per pixel labeled non-occluded in the ground truth data. GMP-MRF [13] has approximately twice the error rate of our algorithms. LOG-L1 and Nonlinear correlation are slightly better than our algorithm using 25 images. Nonlinear correlation with 9 images is the best result presented using our three pixels since the chance of a pixel being occluded increases with the number of camera angles used.

<table>
<thead>
<tr>
<th>Method</th>
<th>Errors &gt; 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct and Kanade</td>
<td>1.4</td>
</tr>
<tr>
<td>GMP-MRF [13]</td>
<td>2.8</td>
</tr>
<tr>
<td>LOG-L1 [12]</td>
<td>4.0</td>
</tr>
<tr>
<td>Nonlinear correlation [10]</td>
<td>10.0</td>
</tr>
<tr>
<td>Nakamura et al. [13] (25 images)</td>
<td>0.3</td>
</tr>
<tr>
<td>Nakamura et al. [10] (9 images)</td>
<td>0.9</td>
</tr>
</tbody>
</table>
Global Approach [Horn’81, Poggio’84, …] encode desirable properties of $d$ in $E(d)$:

$$E(d) = E \left( \begin{array}{c} d_p \\ d_q \\ d_r \\ d_s \end{array} \right)$$

$$\text{arg min}_{d} E(d) = \sum_{p \in P} M(d_p) + \sum_{\{p,q\} \in \text{Neighbors}} P(d_p, d_q)$$

- MAP-MRF
- match pixels of similar color
- most nearby pixels have similar disparity

NP-hard problem $\Rightarrow$ need approximations
Stereo as Energy Minimization

- Matching cost formulated as energy
  - “data” term penalizing bad matches
    \[ D(x, y, d) = |I(x, y) - J(x + d, y)| \]
  - “neighborhood term” encouraging spatial smoothness (continuity; disparity gradient)
    \[ V(d_1, d_2) = \text{cost of adjacent pixels with labels } d_1 \text{ and } d_2 \]
    \[ = |d_1 - d_2| \text{ (or something similar)} \]

\[
E = \sum_{(x,y)} D(x, y, d_{x,y}) + \sum_{\text{neighbors } (x_1,y_1),(x_2,y_2)} V(d_{x_1,y_1}, d_{x_2,y_2})
\]

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 its 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 $w: E \rightarrow \mathbb{R}^+$
- $\text{Cost}(C) = \sum_{\text{edges in } C} w(\text{edge})$
- 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

Boykov et al., Fast Approximate Energy Minimization via Graph Cuts,

Ground truth

<table>
<thead>
<tr>
<th>Local: Compact Window</th>
<th>Global: Expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>high texture</td>
<td></td>
</tr>
<tr>
<td>18 sec</td>
<td>75 sec, $\lambda = 5$</td>
</tr>
<tr>
<td>16% error</td>
<td>16% error</td>
</tr>
<tr>
<td>0.33% error</td>
<td>0.35% error</td>
</tr>
</tbody>
</table>

| medium texture         |                   |
| 12 sec, 3.36% error    | 32 sec, 1.86% error, $\lambda = 20$ |

Results
Difficulties

- Parameter selection

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

- 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
- 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

Late 90’s state of the art  Recent state of the art

left image  true disparities

5.23% errors  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

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

---

**Multi-Baseline Stereo**

- Okutomi and Kanade, 1991
- When images rectified, 
  
  \[
  \begin{align*}
  \text{disparity } d &= x_l - x_r = \frac{fb}{b} \\
  \Rightarrow \quad \frac{d}{b} &= \frac{f}{b}
  \end{align*}
  \]
  
  where \( f \) = focal length, \( b \) = baseline length

- For fixed scene \( d \), \( \frac{f}{b} \) = constant

  \Rightarrow \text{take multiple images from cameras w/ varying } b \text{ and combine them}

  For example,

  \[
  \begin{align*}
  k_1 & \quad k_2 & \quad k_3 & \quad k_4 \\
  b_1 & \quad b_2 & \quad b_3 & \quad b_4
  \end{align*}
  \]
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 \( 7 \times 7 \) cascaded Gaussians
   and 1 \( 9 \times 9 \) Laplacian.
   \( \nabla^2 \) approximates 25 \( \times \) 25 \( \nabla^2 \) filter

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

2.1 Rectify
   Rectify each image \( I_i \) with \( Base \) by warping and resampling
2.2 Match

For each pixel \((i,j)\) in Base
For each pixel \((k,l)\) in epipolar line \(A\) 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.,

\[
\begin{align*}
SSD_2(i,j) & = \sum_{(s,t) \in W(i,j)} \left( s + c_1(k,s) \right)^2 + c_2(l,t) \\
& - Bas (s,t)^2
\end{align*}
\]

where \(Z = (s_1, s_2)\) = unit vector in epipolar line direction in \(I\)
\[
Z = \frac{s_1}{b} = \frac{l_1}{d_z}
\]

(\(b\) of \(Z = d_z\))
Fig. 5. SSD values versus inverse distance: (a) $D = 6$, (b) $D = 12$, (c) $D = 24$.

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

Fig. 7. Combining multiple baseline stereo pairs.

Result is for each inspection image and displacement $Z$, a measure of match:

$$SSSD$$

$$SSD_1$$

$$SSD_2$$

$$SSD_3$$

2.3 Combine Evidence

Sum SSD values:

$$SSSD(i,j,Z) = \sum_i SSD_x(i,j,Z)$$
3. Estimate Depth Map

Find value of $Z$ that minimizes SSD: Fit quadratic function to data points and interpolate to estimate min $Z$.

Depth $z = Z/f$ at each pixel.

- Camera Configurations Used

- 256 x 240 images
- 30 frames per second
- Disparity range 60 pixels

---

The CMU Video-Rate Stereo Machine

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

- 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

- Project “structured” light patterns onto the object
  - simplifies the correspondence problem

Laser Scanning

- 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