Stereo Depth Estimation, Matching
CMV: Panorama Stitching is a Solved Problem
Goals

• Understand why **stereo matching** is the hard part of stereo vision.
• Know the definition and formation of the stereo **cost volume**.
• Understand the basic metrics used to compare patches (SSD, **SAD**, **NCC**)
• Understand the **plane sweep stereo** algorithm
• Understand the distinction between **local** and **global methods** for stereo correspondence.
Announcements

• P1 artifact voting results coming soon...
Camera(s) without a common COP

- With panoramas, we always assumed a common COP.
- How can we model the geometry of a camera in a separate world coordinate system?

Two important coordinate systems:
1. World coordinate system
2. Camera coordinate system

How do we project a given point \((x, y, z)\) in world coordinates?
Projection matrix

\[ \Pi q = (x, y, z, 1) \]
Intrinsic Camera Parameters

Everything you need to get from camera coordinates to pixel coordinates:

\[
K = \begin{bmatrix}
-f & 0 & 0 \\
0 & -f & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

\(\uparrow\text{cam}\)

\(\text{img}\)

\(K\) (intrinsics)

\(K\) (converts from 3D rays in camera coordinate system to pixel coordinates)

in general, \(K = \begin{bmatrix}
-f & s & c_x \\
0 & -\alpha f & c_y \\
0 & 0 & 1
\end{bmatrix}\) (upper triangular matrix)

\(\alpha\) : aspect ratio (1 unless pixels are not square)

\(s\) : skew (0 unless pixels are shaped like rhombi/parallelograms)

\((c_x, c_y)\) : principal point ((0,0) unless optical axis doesn’t intersect projection plane at origin)
Extrinsic Camera Parameters

• Everything you need to get from \textbf{world} coordinates to \textbf{camera} coordinates

\[
\begin{bmatrix}
    R & 0 & 0 \\
    0 & 0 & 1 \\
    0 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
    I_{3 \times 3} & -c \\
    0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

Step 2: Rotate by $R$  \quad \text{Step 1: Translate by $-c$}
This part converts 3D points in world coordinates to 3D rays in the camera’s coordinate system. There are 6 parameters represented (3 for position/translation, 3 for rotation).

The $\mathbf{K}$ matrix converts 3D rays in the camera’s coordinate system to 2D image points in image (pixel) coordinates.

$$\Pi = \mathbf{K} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & 0 \\ 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I}_{3 \times 3} & -\mathbf{c} \end{bmatrix}$$
Projection matrix

\[ \mathbf{q} = (x, y, z, 1) \]

(in homogeneous image coordinates)
Stereo

• Given two images from different viewpoints
  – How can we compute the depth of each point in the image?
  – Based on *how much each pixel moves* between the two images
Stereo

• Given two images from different viewpoints
  – How can we compute the depth of each point in the image?
  – Based on *how much each pixel moves* between the two images
Hypothesis generation time: what relationship do you expect to find between depth and how much a pixel moves?
Depth from Disparity

Assumptions
- 2 cameras
- Same f
- Same PP
- COP off by b in x, known

\[ C_l = (0,0,0) \]

\[ C_r = (b,0,0) \]
\[
\frac{2}{f} = \frac{x}{x_e} \quad (\text{Similar triangles})
\]

\[
\frac{(x_e + x_r)}{f} = b - \frac{2 \times x_r}{f}
\]

\[
x = b - \frac{2 \times x_r}{f} \quad (\text{Solve for } x)
\]

\[
\frac{2 \times x_e}{f} = b - \frac{2 \times x_r}{f} \quad (\text{Set equal})
\]

\[
\frac{(x_e + x_r)}{f} = b \quad (\text{Group } z's)
\]
\[ \mathcal{Z} = \frac{y}{X_l + X_r} \]

\[ \mathcal{Z} \propto \frac{1}{\text{disparity}} \]

**Note:** \( x_l, x_r \) unsigned

If signed, disparity = \( x_l - x_r \)
Depth from disparity

\[
disparity = x - x' = \frac{\text{baseline} \times f}{z}
\]
If I have rectified stereo images, then I can get depth if I can find correspondence.

Good news: only need to search same row!

Bad news: ambiguity abounds!

Matching is the hard part of stereo.
Stereo Depth Reconstruction: Algorithm

\[ C = \text{np.array}(h, w, D) \]

for \( r \) in range(h):
    for \( c \) in range(w):
        for \( d \) in range(D):
            \[ C[r, c, d] = \text{match-cost}(I_l(r, c), I_r(c, r+d)) \]

\[ \text{depth} = \text{np.argmin}(C, \text{axis}=2) \]
Notes:

- \( C \) is called the Cost Volume

- compare windows around \( I_q[r,c], I_{r,c+1} \)
The Cost Volume

\[ y = 141 \]

\[ C(x, y, d); \text{the cost volume (aka disparity space image (DSI))} \]
Window size

Effect of window size
- Smaller window
  + better detail
  - more noise
- Larger window
  + less noise
  - coarser

Better results with *adaptive window*


Metrics for Stereo Matching

- **SSD** = sum of squared differences
  \[ \sum_{i} (w_{i} - w_{2})^{2} \]

- **SAD** = sum of absolute differences
  \[ \sum_{i} |w_{i} - w_{2}| \]

- **NCC** = normalized cross-correlation
  \[ \frac{\sum_{i} (w_{i} - \bar{w})(w_{2} - \bar{w})}{\sqrt{\sum_{i} (w_{i} - \bar{w})^{2}} \sqrt{\sum_{i} (w_{2} - \bar{w})^{2}}} \]
Normalized Cross Correlation

regions $A, B$, write as vectors $a, b$

subtract the mean of each vector:

$$a \rightarrow a - \langle a \rangle, \quad b \rightarrow b - \langle b \rangle$$

cross correlation $= \frac{a \cdot b}{|a||b|}$

$$\frac{a}{|a|} \cdot \frac{b}{|b|}$$

Invariant to $I \rightarrow \alpha I + \beta$
Stereo matching based on SSD

Left

Right

scanline

SSD

Best matching disparity

$d_{min}$
Stereo with NCC: The Good Case
Stereo with NCC: The Bad Case

- Target region
- Left image band
- Right image band
- Cross correlation

1

0.5

0
Stereo results

– Data from University of Tsukuba
– Similar results on other images without ground truth
Results with window search

Window-based matching (best window size)

Ground truth
Better methods exist...

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

Ground truth

For the latest and greatest:  http://www.middlebury.edu/stereo/
Stereo as energy minimization

• What defines a good stereo correspondence?
  1. Match quality
     • Want each pixel to find a good match in the other image
  2. Smoothness
     • If two pixels are adjacent, they should (usually) move about the same amount
Stereo as energy minimization

\[ I(x, y) \quad J(x, y) \]

\[ y = 141 \]

\[ C(x, y, d); \text{the disparity space image (DSI)} \]
Greedy selection of best match