CSE473/573 – Stereo Correspondence

Presented By:
Radhakrishna Dasari
Contents

• Stereo Vision

• Depth With Stereo

• Correspondence Search

• Dynamic Programming based Stereo Matching
Stereo Vision – Goal

- The recovery of the 3D structure of a scene using two images acquired from a different viewpoint in space.
- The images can be obtained using two cameras separated by a distance or one moving camera.
Triangulation – Principle underlying Stereo Vision

• The 3D location of a point in space is restricted to a straight line that passes through the center of projection and the projection of the object point on the image plane.

• Stereo Vision determines the position of a point in space by finding the intersection of the two lines passing through the center of projection and the projection of the object point on the image planes.
Stereo Vision - Notation

- recover depth by finding image coordinate $x'$ in one image that corresponds to $x$ in the other image
Stereo Vision – Sub problems

- Calibration: How do we recover the relation of the cameras – i.e the distance between camera centers, orientation of image planes w.r.t each other (If not already known?)

- Correspondence: How do we search for the matching point $x'$?
Stereo Correspondence Problem

- We have two images taken from cameras with different intrinsic (focal length, resolution) and extrinsic (distance between cameras and orientation change) parameters.

- How do we match a point in the first image to a point in the second? How can we constrain our search?
Depth Map Estimation

Given a calibrated binocular stereo pair, can we fuse it to produce a depth image?
Binocular Fusion Problem

- The left case, there is no ambiguity, and stereo reconstruction is a simple matter.

- In the more usual case shown on the right, any of the four points in the left picture may, match any of the four points in the right one.

- Only four of these correspondences are correct. the other ones yield the incorrect reconstructions shown as small gray discs.
Binocular Fusion Problem
Epipolar Constraint - Point corresponds to a Ray

We try to estimate the position of $X$ based on $x$ and $x'$. From Geometry

- Potential matches for $x$ have to lie on the corresponding line $l'$.
- Potential matches for $x'$ have to lie on the corresponding line $l$. 

![Diagram showing epipolar constraint](image-url)
Epipolar Geometry: Notation

- **Baseline** – Line connecting camera centers $O$ and $O'$

- **Epipoles** – Intersection of baseline with the image planes, which are projections of camera centers on each other. $e$ and $e'$

- **Epipolar Plane** – Planes containing baseline
Epipolar Geometry: Notation continued..

- **Epipolar Lines** - intersections of epipolar plane with image planes (always come in corresponding pairs)

In this picture $l$ and $l'$ are the epipolar lines, for the epipolar plane $O XO'$
Image Rectification

• Arbitrary arrangement of cameras results in image planes that are not parallel.

• Complicated Epipolar representation and correspondence search. How to make it simpler?
Image Rectification

• Re-project image planes onto a common plane parallel to the line between optical centers

• Pixel search would be horizontal after this transformation

Image Rectification - Example

Original stereo pair

Stereo pair in standard form
Image Rectification - Example
Basic Stereo Matching Algorithm

- Perform Stereo Rectification

- For each pixel $x$ in the first image
  - Find corresponding epipolar scanline in the right image
  - Examine all pixels on the scanline and pick the best match $x'$
Basic Stereo Matching Algorithm

```plaintext
for x=1:w,
  for y=1:h,
    bestdist=inf;
    for i=-dr:0,
      if (dist(pix(x,y),pix(x+i,y))<bestdist)
        d(x,y)=i; best=sim(pix(x,y),pix(x+i,y)); end
    end
  end
end
```
Assumptions in Stereo Correspondence

Let’s make some assumptions to simplify the matching problem

1. Baseline is relatively small (compared to the depth of scene points)
2. Most scene points are visible in both views
3. Also, matching regions are similar in appearance
Disparity

Disparity is defined as the difference in the location of a feature point between the right and left images.

Disparity high for closer objects and low for distant objects.

\[ d = x^l - x^r \]
Depth from Disparity

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

\[ d = \frac{Bf}{Z} \]

B – Baseline
Z – Depth
d – Disparity
f – focal length
Depth Resolution from Disparity

\[
\frac{dd}{dZ} = \frac{Bf}{Z^2}
\]

\[
\Delta Z = \frac{Z^2}{Bf} \Delta d
\]
Correspondence search with similarity constraint

- Slide a window along the right scanline and compare contents of that window with the reference window in the left image

- Matching cost: SSD or normalized correlation
Correspondence search with similarity constraint
Correspondence search with similarity constraint
Effect of Window Size

- Smaller Window Size – More detail, But More Noise
- Large Window Size – Smooth disparity map, but less detail
Similarity Constraint

• Corresponding regions in two images should be similar in appearance

• ...and non-corresponding regions should be different

• When will the similarity constraint fail?
Similarity Constraint Failure

Textureless surfaces

Occlusions, repetition

Non-Lambertian surfaces, specularities
Results with Window Search

Data

Window-based matching

Ground truth
How can we improve window based matching?

- The similarity constraint is **local** (each reference window is matched independently)
- Need to enforce **non-local** correspondence constraints
Non-Local Constraints

- Uniqueness - For any point in one image, there should be at most one matching point in the other image
Non-Local Constraints

- Ordering - Corresponding points should be in the same order in both the images
Non-Local Constraints

- **Ordering** - Corresponding points should be in the same order in both the images. Does not always hold true. For instance, if we have a smaller object in front of a large object, the images violate the ordering constraint.

- **Smoothness in disparity** - Also, we expect disparity values to change slowly.

![Diagram of non-local constraints](image)
Surface as Path

surface slice

surface as a path

occlusion left

occlusion right
Dynamic Programming (Baker and Binford, 1981)

% Loop over all nodes \((k, l)\) in ascending order.
for \(k = 1\) to \(m\) do
  for \(l = 1\) to \(n\) do
    % Initialize optimal cost \(C(k, l)\) and backward pointer \(B(k, l)\).
    \(C(k, l) \leftarrow +\infty; B(k, l) \leftarrow \text{nil};\)
    % Loop over all inferior neighbors \((i, j)\) of \((k, l)\).
    for \((i, j) \in \text{Inferior} - \text{Neighbors}(k, l)\) do
      % Compute new path cost and update backward pointer if necessary.
      \(d \leftarrow C(i, j) + \text{Arc} - \text{Cost}(i, j, k, l);\)
      if \(d < C(k, l)\) then \(C(k, l) \leftarrow d; B(k, l) \leftarrow (i, j)\) endif;
      endfor;
    endfor;
  endfor;
% Construct optimal path by following backward pointers from \((m, n)\).
\(P \leftarrow \{(m, n)\}; (i, j) \leftarrow (m, n);\)
while \(B(i, j) \neq \text{nil}\) do \((i, j) \leftarrow B(i, j); P \leftarrow \{(i, j)\} \cup P\) endwhile.
Stereo Matching With Constraints

Similarity measure (SSD or NCC)

Optimal path (dynamic programming)

Constraints
- epipolar
- ordering
- uniqueness
- disparity limit

(Cox et al. CVGIP’96; Koch’96; Falkenhagen’97; Van Meerbergen, Vergauwen, Pollefeys, VanGool IJCV’02)
The role of Baseline

Small baseline: large depth error
Large baseline: difficult search problem
Problem with large baselines

- Matching with fixed-size windows will fail!
- Possible solution: adaptively vary window size
Stereo Algorithms evaluation

Middlebury Stereo Evaluation - Version 2

New features and main differences to version 1
Submit and evaluate your own results

☐ Open a new window for each link

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Avg.</th>
<th>Rank</th>
<th>Tsukuba ground truth</th>
<th>Venus ground truth</th>
<th>Teddy ground truth</th>
<th>Cones ground truth</th>
<th>Average percent of bad pixels (explanation)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>nonocc</td>
<td>all</td>
<td>disc</td>
<td>nonocc</td>
<td>all</td>
</tr>
<tr>
<td>TSGO [143]</td>
<td>10.5</td>
<td></td>
<td>0.87</td>
<td>1.13</td>
<td>4.66</td>
<td>0.11</td>
<td>0.24</td>
</tr>
<tr>
<td>JSOSP-GCP [151]</td>
<td>11.8</td>
<td></td>
<td>0.74</td>
<td>1.34</td>
<td>3.98</td>
<td>0.08</td>
<td>0.16</td>
</tr>
<tr>
<td>AD Census [82]</td>
<td>14.1</td>
<td></td>
<td>1.07</td>
<td>1.48</td>
<td>5.73</td>
<td>3.22</td>
<td>0.06</td>
</tr>
<tr>
<td>AdaptingBP [16]</td>
<td>17.9</td>
<td></td>
<td>1.11</td>
<td>1.37</td>
<td>5.79</td>
<td>2.24</td>
<td>0.10</td>
</tr>
<tr>
<td>CoopRegion [39]</td>
<td>18.4</td>
<td></td>
<td>0.87</td>
<td>1.16</td>
<td>4.61</td>
<td>0.11</td>
<td>0.24</td>
</tr>
<tr>
<td>OC RADAR [152]</td>
<td>22.5</td>
<td></td>
<td>1.15</td>
<td>1.42</td>
<td>6.23</td>
<td>3.38</td>
<td>0.15</td>
</tr>
<tr>
<td>RDP [87]</td>
<td>23.8</td>
<td></td>
<td>0.57</td>
<td>1.39</td>
<td>5.00</td>
<td>2.24</td>
<td>0.21</td>
</tr>
<tr>
<td>MultiRBF [129]</td>
<td>24.2</td>
<td></td>
<td>1.33</td>
<td>1.56</td>
<td>6.02</td>
<td>3.38</td>
<td>0.13</td>
</tr>
<tr>
<td>DoubleBP [34]</td>
<td>24.9</td>
<td></td>
<td>0.88</td>
<td>1.29</td>
<td>4.75</td>
<td>0.13</td>
<td>0.45</td>
</tr>
</tbody>
</table>

http://vision.middlebury.edu/stereo/eval/
Slide Credits

- Marc Pollefeys, Luc Van Gool, Derek Hoiem, Lazebnik, Stefano Mattoccia