Lecture 09: Stereo Algorithms

Recall: Simple Stereo System

![Diagram showing a simple stereo system with left and right cameras located at (0,0,0) and (T_x,0,0).]

Image coords of point (X,Y,Z)
Left Camera: \(x_l = \frac{X}{Z} \) \(y_l = \frac{Y}{Z}\)
Right Camera: \(x_r = \frac{X - T_x}{Z} \) \(y_r = \frac{Y}{Z}\)

Recall: Stereo Disparity

![Diagram showing the disparity equation.]

Important equation!

Stereo Example

![Images of left and right stereo views.](http://www.middlebury.edu/stereo/)

Note how disparity is larger (brighter) for closer surfaces.
Computing Disparity

- Correspondence Problem:
  - Determining which pixel in the right image
    corresponds to each pixel in the left image.
  - \( \text{Disp} = x_{\text{coord}}(\text{left}) - x_{\text{coord}}(\text{right}) \)

Recall our discussion of computing correspondences
of image patches (Lecture 7).
SSD - sum of squared difference measure
NCC - normalized cross correlation measure

Epipolar Constraint

Important Concept:

For stereo matching, we don’t have to search the
whole 2D right image for a corresponding point.
The “epipolar constraint” reduces the search space
to a one-dimensional line.

Recall: Simple Stereo System

\[
\begin{align*}
\text{Left camera:} & \quad x_l = f \frac{X}{Z} \\
\text{Right camera:} & \quad x_r = f \frac{X - T_x}{Z} \\
\end{align*}
\]

Same Y Coord!

Matching using Epipolar Lines

For a patch in left image
Compare with patches along
same row in right image

Match Score Values

Example: 5x5 windows
NCC match score

Computed disparities
Ground truth
Black pixels: bad disparity values,
or no matching patch in right image


**Occlusions: No matches**

- Left image
- Right image

**Effects of Patch Size**

- 5x5 patches: Smoother in some areas
- 11x11 patches: Loss of finer details

**Adding Inter-Scanline Consistency**

So far, each left image patch has been matched independently along the right epipolar line.

This can lead to errors.

We would like to enforce some consistency among matches in the same row (scanline).

**Disparity Space Image**

First we introduce the concept of DSI. The DSI for one row represents pairwise match scores between patches along that row in the left and right image.

**Disparity Space Image (DSI)**

- Left Image
- Right Image

Dissimilarity Values

(1-NCC) or SSD
Disparity Space Image (DSI)

Left Image

Right Image

Dissimilarity Values

(1-NCC) or SSD

Disparity Space Image (DSI)

Left Image

DSI

Enter each vector of match scores as a column in the DSI

Dissimilarity Values

Invalid entries due to constraint that disparity >= low value (0 in this case)

Invalid entries due to constraint that disparity <= high value (64 in this case)

Disparity Space Image

Left scanline

Right scanline

Disparity

Disparity Space Image

N cols in left scanline

M cols in right scanline

However, I'm going to keep the full image around, including invalid values (I think it is easier to understand the pixel coordinates involved)

Disparity Space Image

N cols in left scanline (e.g. N)

Disparity (e.g. 64)

DSI and Scanline Consistency

Assigning disparities to all pixels in left scanline now amounts to finding a connected path through the DSI

Start

End
Lowest Cost Path

We would like to choose the “best” path.
Want one with lowest “cost” (Lowest sum of dissimilarity scores along the path)

Constraints on Path

It is common to impose an ordering constraint on the path. Intuitively, the path is not allowed to “double back” on itself.

Ordering Constraint

Ordering constraint… …and its failure

Dealing with Occlusions

Left scanline Right scanline

An Optimal Scanline Strategy

- We want to find best path, taking into account ordering constraint and the possibility of occlusions.

Algorithm we will discuss now is from Cox, Hingorani, Rao, Maggs, “A Maximum Likelihood Stereo Algorithm,” Computer Vision and Image Understanding, Vol 63(3), May 1996, pp.542-567.
Three cases:
- Matching patches. Cost = dissimilarity score.
- Occluded from right. Cost is some constant value.
- Occluded from left. Cost is some constant value.

\[ C(i,j) = \min \{ C(i-1,j-1) + \text{dissimilarity}(i,j), C(i-1,j) + \text{occlusionConstant}, C(i,j-1) + \text{occlusionConstant} \}; \]

Recap: want to find lowest cost path from upper left to lower right of DSI image.

At each point on the path we have three choices: step left, step down, step diagonally.

Each choice has a well-defined cost associated with it.

This problem just screams out for Dynamic Programming! (which, indeed, is how Cox et al. solve the problem)

Every pixel in left column now is marked with either a disparity value, or an occlusion label.

Proceed for every scanline in left image.

Simple trick for filling in gaps caused by occlusion.

<table>
<thead>
<tr>
<th>Red</th>
<th>Orange</th>
<th>Purple</th>
<th>Gray</th>
<th>Black</th>
</tr>
</thead>
</table>

= left occluded

Fill in left occluded pixels with value from the nearest valid pixel preceding it in the scanline.

Similarly, for right occluded, look for valid pixel to the right.

Result of DP alg with occlusion filling.
Example

Result of DP alg with occlusion filling. Result without DP (independent pixels).

Example

Result of DP alg with occlusion filling. Ground truth.

www.middlebury.edu/stereo/

State-of-the-Art Results

Algorithm Results

Ground truth

J. Sun, Y. Li, S.B. Kang, and H.-Y. Shum.