Homework 2.A: An investigation on the stereo matching criteria
Jonathan Thomspn and Ye Ji and Davi Geiger

Introduction

If two cameras are parallel to each other (no rotation) and if the translation is simply along "X", then

\[ x^R = x^L + \frac{T_x}{Z^L(\tilde{x}^L)} \quad y^R = y^L \quad Z^R(\tilde{x}^R) = Z^L(\tilde{x}^L) \]

\[ \tilde{d}(\tilde{x}^L) = \begin{bmatrix} d_x(\tilde{x}^L) \\ d_y(\tilde{x}^L) \end{bmatrix} = \begin{bmatrix} x^R - x^L \\ y^R - y^L \end{bmatrix} = \begin{bmatrix} \frac{T_x}{Z^L(\tilde{x}^L)} \\ 0 \end{bmatrix} \]

We will be working with the two image pairs "Aloe" and "Monopoly" from the middlebury 2006 stereo data set.

Question 1: Occlusions

The first step is to detect the occlusion regions. The input is the two disparity maps provided in the web site (one is the ground truth for the left image, \( d^L \), and the other is the ground truth for the right image, \( d^R \)).

How to find the occlusion regions? According to the web site: "Occlusion maps can be generated by crosschecking the pair of disparity maps". More precisely, for each pixel \( \tilde{x}^L \), and its correspondent \( \tilde{x}^R = (x^R = x^L - d^R(\tilde{x}^L), y^R = y^L) \), check that \( d^L(\tilde{x}^L) \neq d^R(\tilde{x}^R) \). If \( d^L(\tilde{x}^L) = d^R(\tilde{x}^R) \), then no occlusion, i.e., \( O^L(\tilde{x}^L) = 0 \). Else if \( d^L(\tilde{x}^L) \neq d^R(\tilde{x}^R) \) then it is occluded, i.e., \( O^L(\tilde{x}^L) = 1 \). In this way, you construct an occlusion variable, \( O^L(\tilde{x}^L) \), that is 0 or 1 and whenever \( O^L(\tilde{x}^L) = 1 \) we have an occlusion in the left eye.

We could also construct an occlusion variable on the right eye, \( O^R(\tilde{x}^R) \), such that for \( \tilde{x}^L = (x^L = x^R + d^R(\tilde{x}^R), y^L = y^R) \) we check if \( d^L(\tilde{x}^L) \neq d^R(\tilde{x}^R) \) then \( O^R(\tilde{x}^R) = 1 \), and \( O^R(\tilde{x}^R) = 0 \) otherwise.
Create an occlusion variable for the left eye, and show the image on the left eye with pixels \( I(x^L) = 0 \) ("black") if the pixel is such that \( O^L(x^L) = 1 \).

**Question 2: Histogram of the errors**

Consider three types of error for a match of pixel \( x^L \) with disparity \( d(x^L) \).

1. \[
E_{x^L}(\bar{d}(x^L)) = \left[ \frac{I^L(x^L)}{255} - \frac{I^R(x^L + \bar{d}(x^L))}{255} \right]^2 = \left[ \frac{I^L(x^L, y^L)}{255} - \frac{I^R(x^L + d(x^L), y^L)}{255} \right]^2
\]

2. \[
E_{x^L}(\bar{d}(x^L)) = \min_{\sigma, \theta} \left| (I^L \ast \psi_{\sigma, \theta})(x^L) - (I^R \ast \psi_{\sigma, \theta})(x^L + \bar{d}(x^L)) \right|^2
\]

3. \[
E_{x^L}(\bar{d}(x^L)) = \min_{\sigma, \theta} \left| I^L \ast \psi_{\sigma, \theta}(x^L) - |(I^R \ast \psi_{\sigma, \theta})(x^L + \bar{d}(x^L))|^2 \right|
\]

Note that the magnitude of a complex number \( z = a + i b \) is \( |z| = |a + i b| = \sqrt{a^2 + b^2} \) and so, of course, \( z^2 = a^2 + b^2 \).

**Plot a histogram for each error, given the "true" disparity:** A histogram \( H(E_{x^L}(d_{\text{true}})) \) is constructed by having the "x-axis" to be the possible values for the error and the "y-axis" is the accumulation of votes for each error.

The trick is to create bins of error range. So for each image, one creates bins (or ranges for the errors) so that any pixel \( x^L \) will have its error (for the true disparity) to fall in some bin (to be within some range) for which the vote is cast (accumulating the "y-axis" value for that error). How to set the bins? it is a bit of an art. "Try your best" and make sure that your program has this parameter (the range of each bin) easy to be changed, so we can discuss in class.

For the first error, the square of intensity differences, you may get reasonable results with the bins being of size 0.0005!. Yes, bin size is \( 5 \times 10^{-4} \). By the time you are filling
the bin 20, i.e., error 0.01, your histogram values are already close to zero. Yes, you may have outliers, a few entries that the errors is large. We don’t worry about it.

So there are 6 histograms to be created, three for the image "Aloe" and three for the image "Monopoly".

**Attention: oclusions.** Exclude points at occlusions, since at these points no matching occurs.

**Compute Mean Error and Standard Deviation.** The Histograms are like probability distribution, if we normalize them. So first normalize each of the histograms, dividing each column of $H(E)$ by the total sum of the columns (total area), i.e., $H_{\text{normalized}}(E) = \frac{H(E)}{\sum E H(E)}$. Then, consider the histogram to be a probability distribution on the error. Compute the mean error and the standard deviation of the error for each Histogram, i.e.,

$$<E> = \sum E \frac{H(E)}{\sum E H(E)} \text{ and } <E^2> = \sum E^2 \frac{H(E)}{\sum E H(E)}$$

So $\sigma = \sqrt{<E^2> - <E>^2} = \sqrt{<E^2> - 2<E><E> + <E>^2} = \sqrt{<E^2> - <E>^2}$.

**Question 3: Minimum errors**

We want to investigate if indeed these errors, for the true disparity, are minimum in the sense that by varying the disparity values a "little bit", the error increases. If it decreases, then we found a better disparity match for these error criteria. So what we do in this question is: for each pixel and true disparity, change the true disparity by random value between $\{-1, 0, 1\}$. Another way to say is that we change the true disparity

$$\vec{d}^{\text{true}}(\vec{x}_L) \rightarrow \vec{d}^{\text{new}}(\vec{x}_L) = \vec{d}^{\text{true}}(\vec{x}_L) + \text{random}(\{-1, 0, 1\})$$

then for this new disparity map, compute the three error histograms. How to compare the two histograms, the one computed with $\vec{d}^{\text{true}}$ with the one computed with $\vec{d}^{\text{new}}$. 
One simple method, just compute the mean error and standard deviation and compare them. Hopefully, the range $<E> \pm \sigma$ is smaller for the true disparity map. Let us know for the three different error criteria.