Homework 2: An investigation on the stereo matching criteria

Davi Geiger

Due October 4th, 2016

Introduction

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

\[
\begin{align*}
Z_0 - f & = T_x - d_0 \\
\Rightarrow d_0 & = f \frac{T_x}{Z_0 - f}
\end{align*}
\]

Figure 1: Projection to parallel cameras. From the triangle on the right we derive \( \frac{Z_0}{Z_0 - f} = \frac{T_x}{T_x - d_0} \rightarrow d_0 = f \frac{T_x}{Z_0} \)

\[
\begin{align*}
x_{r,e}^R &= x_{l,e}^L - f \frac{T_x}{Z^L(x_{l,e}^L, y_{l,e}^L)} & y_{r,e}^R &= y_{l,e}^L & \text{and} & & Z^R(x_{r,e}^R, y_{r,e}^R) &= Z^L(x_{l,e}^L, y_{l,e}^L) \\
d(x_{l,e}^L, y_{l,e}^L) &= x_{l,e}^L - x_{r,e}^R = f \frac{T_x}{Z^L(x_{l,e}^L, y_{l,e}^L)}
\end{align*}
\] (1)
This is also the case for images that have been "rectified". We will be working with the two image pairs from the middlebury 2014 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? Occlusion maps can be generated by cross-checking the pair of disparity maps. More precisely, for each pixel \(\vec{x}_L\), and its correspondent \(\vec{x}_R = (x_R = x_L - d_L(\vec{x}_L), y_R = y_L)\), check that \(d_L(\vec{x}_L) = d_R(\vec{x}_R)\). If \(|d_L(\vec{x}_L) - d_R(\vec{x}_R)| < 0.5 \text{ pixels}\), then there is no occlusion, i.e., \(O_L(\vec{x}_L) = 0\). Else if \(|d_L(\vec{x}_L) - d_R(\vec{x}_R)| \geq 0.5 \text{ pixels}\) then it is occluded, i.e., \(O_L(\vec{x}_L) = 1\). In this way, you construct an occlusion variable, \(O_L(\vec{x}_L)\) (left eye) that is 0 or 1.

Create an occlusion variable for the left eye, and show the image on the left eye with pixels \(I(\vec{x}_L) = (0, 0, 0)\) ("black") if the pixel is such that \(O_L(\vec{x}_L) = 1\).

**Planar surfaces: linearly varying disparities**

Let us consider the surface along epipolar lines, i.e., surface \(Z^L(\vec{x}_L) = Z^L(x_{l,e}^L, y_{l,e}^L)\) with the left coordinate index \(l = 1, ..., N\) and the epipolar index \(e = 1, ..., M\). We investigate the locally planar assumption around a pixel \(\vec{x}_0^L = (x_0^L, y_0^L) = (x_{l_0,e_0}^L, y_{l_0,e_0}^L)\) (see figure 2). Note that in theory, the coordinates can be float numbers. We first investigate the assumption along a line segment, where \(x_{l,e}^L\) varies. The planar (or linear) assumption is written as \(Z^L(x_0^L + \delta x_{l,e}^L, y_0^L) \approx Z^L(x_0^L, y_0^L) + A \delta x^L\) for \(A = \frac{\partial Z^L(x_{l,e}^L, y_{l,e}^L)}{\partial x_l} |_{(x_0^L, y_0^L)}\) for an unknown slant \(A\).

In order to demonstrate that the disparities \(d(x_{l,e}^L, y_{l,e}^L)\) also vary linearly for \(x_{l,e}^L \in [x_0^L - \delta, x_0^L + \delta]\), we assume the changes in depth are small compared to the depth value itself, i.e.,

\[
\delta_{x_{l,e}^L} Z^L(\vec{x}_0^L) = A \delta x^L << Z^L(\vec{x}_0^L)
\]
Figure 2: **Stereo Tilted surface.** The right side depicts a tilted surface with the left and right cameras seeing from above. A match occurs for \( x_0 \) and \( x_R_0 \). On the left is the “matching space”, with the horizontal axis representing a left epipolar line and the vertical axis representing the corresponding right epipolar line. We observe the foreshortening of the line segment on the right image where the range \([−\delta_R, \delta_R]\) is smaller than in the left image \([−\delta, \delta]\). Note that on the right, depth is represented by the z axis. On the left it is the disparity represented by the \( 135^\circ \) diagonal axis. The larger is the disparity the smaller is z, according to equation 1. Also, we see the limitation of the integer disparities, a stair case solution. A floating point disparity is needed for more accuracy.

Thus, for a local line segment \( x^L_{l,e} \in [x^L_0 - \delta, x^L_0 + \delta] \) (see figure 2), we obtain

\[
d(x^L_{l,e}, y^L_{l,e}) = x^L_{l,e} - x^R_{r,e} = f \frac{T_x}{Z^L(x^L_{l,e}, y^L_{l,e})} \\
\downarrow \\
d(x^L_0 + \delta x^L, y^L_0) = f T_x (Z^L(x^L_0, y^L_0) + A \delta x^L)^{-1} \\
= f \frac{T_x}{Z^L(x^L_0, y^L_0)} \left( 1 + \frac{A}{Z^L(x^L_0, y^L_0)} \delta x^L \right)^{-1} \\
= d(x^L_0, y^L_0) \left( 1 - \frac{A}{Z^L(x^L_0, y^L_0)} \delta x^L \right) \\
= d(x^L_0, y^L_0) - \alpha \delta x^L
\]

where \( \alpha = f \frac{T_x A}{(Z^L(x^L_0, y^L_0))^2} = \frac{A}{f T_x} (d(x^L_0, y^L_0))^2 \) is unknown since A is unknown.
Floating point disparities

In order to provide the "truth displacement" we add a fractional displacement $t_0$ (floating point number) to a given "integer disparity" hypothesis $d(x_0^L, y_0^L) \rightarrow x_0^R = x_0^L - d(x_0^L, y_0^L)$, where $x_0^R, x_0^L, d(x_0^L, y_0^L) \in \mathbb{Z}$, i.e., we describe the true match to $x_0^L$ to be given by

$$ x_0^L \leftrightarrow x_0^R - t_0 = x_0^L - (d(x_0^L, y_0^L) + t_0) \tag{3} $$

where $x_0^R, x_0^L, d(x_0^L, y_0^L) \in \mathbb{Z}$ and $t_0 \in \mathbb{R}$.

![Diagram of matching line segments](image)

**Figure 3**: Linear match hypothesis (5). We show a left and right images, with the blue segment representing the matching line segments, with sizes $[-\delta, \delta]$ and $[-\delta^R, \delta^R]$, respectively. Points along the segments are represented by $x_0^L, x_0^L + \delta x^L$ and $x_0^R, x_0^R + t_0 + \delta x^R$. Thus, $\delta x^L = x^L - x_0^L$ and $\delta x^R = x^R - x_0^R + t_0$.

The parameters $\alpha, t_0$ are unknowns we want to recover.

**Complete Hypothesis**

Our hypothesis 3 added to the linear planar assumption gives
\[ x^L \in [x_0^L - \delta, x_0^L + \delta] \quad \leftrightarrow \quad x^R \in [x_0^R - t_0 - \delta^R, x_0^R - t_0 + \delta^R] \]  

where \( x_0^R = x_0^L - d(x_0^L, y_0^L) \), \( \delta^R = (1 + \alpha) \delta \) and \( x^R \in \mathbb{R} \) (see figure 2 and figure 3). Within the line segments the point wise match for \( x^L \) is given by

\[ x^L \leftrightarrow x^R = x^L - d(x_0^L, y_0^L) - t_0^i + \alpha (x^L - x_0^L) \]

or

\[ x^R - x_0^R = -t_0^i + (1 + \alpha) (x^L - x_0^L) \]

Thus, the center of the line segment hypothesis in the right image is at \( x_0^R - t_0 = x_0^L - d(x_0^L, y_0^L) - t_0 \) and while \( \delta x^L \in [-\delta, \delta] \) along the line segment in the left image, \( \delta x^R \in [-\delta^R, \delta^R] \) along the line segment in the right image. These quantities depend on \( \alpha, t_0 \), which are the unknowns we want to recover.

### Wavelet Transform for Matching Hypothesis

Let us consider the Morlet Wavelets

\[ \psi_{\sigma, \theta, \vec{T}}(\vec{x}) = \frac{1}{\sigma} \psi \left( R(-\theta) \frac{(\vec{x} - \vec{T})}{\sigma} \right) \]

\[ = C_1 \frac{1}{\sigma} \left( e^{i \frac{\pi}{2\sigma} [(1,0) \cdot R(-\theta)(\vec{x} - \vec{T})]} - C_2 \right) e^{-\frac{1}{2\sigma^2} \left[ R(-\theta)(\vec{x} - \vec{T}) \right]^2} \]

\[ = C_1 \frac{1}{\sigma} \left( e^{i \frac{\pi}{2\sigma} \tilde{e}_\theta \cdot (\vec{x} - \vec{T})} - C_2 \right) e^{-\frac{1}{2\sigma^2} (\vec{x} - \vec{T})^2} \]

where \( \tilde{e}_\theta = R(\theta) \begin{pmatrix} 1 \\ 0 \end{pmatrix} \). The Morlet Wavelet transform of an image is then given by

\[ \mathcal{W}[I](\sigma, \theta, \vec{T}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(\vec{x}) C_1 \frac{1}{\sigma} \left( e^{i \frac{\pi}{2\sigma} \tilde{e}_\theta \cdot (\vec{x} - \vec{T})} - C_2 \right) e^{-\frac{1}{2\sigma^2} (\vec{x} - \vec{T})^2} d^2\vec{x} \]

### Wavelets under the hypothesis matching

The hypothesis (5) in the wavelet domain implies
\[ W[I^L](\sigma, \theta_0, x^L_0 + \delta x^L, y_0) \approx W[I^R](\sigma (1 + \alpha), \theta_0, x^R_0 + \delta x^R - t_0, y^R_0) \]
\[ \delta x^L \in [-\delta, \delta] \]
\[ = W[I^R](\sigma (1 + \alpha), \theta_0, x^L_0 - d(x^L_0, y^R_0) + (1 + \alpha) \delta x^L - t_0, y^R_0) \quad (8) \]

where \( \delta x^L = x^L - x^L_0 \) and \( \delta x^R = x^R - x^R_0 + t_0 \).

**Proof:**

\[
W[I^L](\sigma, \theta, x^L_0 + \delta x^L, y_0) =
\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I^L(x, y) C_1 \frac{1}{\sigma} \left( e^{i \frac{\pi}{2\sigma} \theta \cdot (x, y) - (x^L_0 + \delta x^L, y_0)} - C_2 \right) e^{-\frac{1}{2 \sigma^2} ((x,y) - (x^L_0 + \delta x^L, y_0))^2} \, dx \, dy
\]

defining \( x' = x - x^L_0 - \delta x^L, \quad y' = y - y_0 \)

\[
= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I^L(x^L_0 + \delta x^L + x', y' + y_0) C_1 \frac{1}{\sigma} \left( e^{i \frac{\pi}{2\sigma^2} \theta \cdot (x', y') - C_2 \right) e^{-\frac{1}{2 \sigma^2} ((x', y')^2 + (y')^2} \, dx' \, dy'
\]

\[
= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I^R(x^R_0 - t_0 + (1 + \alpha) (x' + \delta x^L)) C_1 \frac{1}{\sigma} \left( e^{i \frac{\pi}{2\sigma} \theta \cdot (x', y') - C_2 \right) e^{-\frac{1}{2 \sigma^2} ((x', y')^2 + (y')^2} \, dx' \, dy'
\]

defining \( \sigma' = \sigma (1 + \alpha) \) and \( z = x^R_0 - t_0 + (1 + \alpha) (x' + \delta x^L) \) → \[
\left\{
\begin{array}{l}
dz = (1 + \alpha) \, dx' \\
x' = \frac{z - (x^R_0 - t_0 + (1 + \alpha) \delta x^L)}{(1 + \alpha)}
\end{array}
\right.
\]

\[
= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I^R(z) \frac{C_1}{\sigma'} \left( e^{i \frac{\pi}{2\sigma'} \theta \cdot ((z, y) - (x^R_0 - t_0 + \delta x^R, y_0))} - C_2 \right) e^{-\frac{1}{2 \sigma'^2} ((z, y) - (x^R_0 - t_0 + \delta x^R, y_0))^2} \, dz \, dy
\]

\[ = W[I^R](\sigma (1 + \alpha), \theta, x^R_0 - t_0 + \delta x^R, y_0) \quad (9) \]

where \( \delta x^R = (1 + \alpha) \delta x^L \). Typically, recomputing the Wavelet transform for different floating point locations \( x^R_0 - t_0 + \delta x^R \) and with floating point scales \( \sigma (1 + \alpha) \) is too expensive. Here we will use the pre-computed wavelets at the near integer location to \( x^R_0 - t_0 + \delta x^R \) and near integer pre-computed scale to \( \sigma (1 + \alpha) \).

The parameters \( \alpha, t_0 \) are unknowns we want to recover.

**Recovery of the Matching Parameters**

We will propose an iterative algorithm to estimate \( \alpha, t_0 \). Our first initial condition is that \( \alpha = 0, t_0 = 0 \) and the iterative algorithm improves its estimate.

We now describe the \( i^{th} \)-Iteration of the algorithm. We start with an estimation of the parameters \( \alpha^{i-1}, t_0^{i-1}, \) from the previous step of the algorithm and show the \( i^{th} \) step.
**$i^{th}$-Iteration**

We are given the previous estimate of $\alpha^{i-1}, t_0^{i-1}$. We derive the new estimates of parameters in two steps. We first estimate the moments and from the moments we derive the new estimate for the parameters.

**Estimation of 1$^{st}$ and 2$^{nd}$ moments**

We can estimate the 1$^{st}$ and 2$^{nd}$ moments along the line segment. More precisely, the first moment is given by

$$< x^L > = \frac{1}{\sum_{\delta x^L = -\delta} W[I^L](\sigma, \theta_0, x^L, y_0)} \sum_{\delta x^L = x^L_0 - \delta} x^L W[I^L](\sigma, \theta_0, x^L, y_0)$$

and

$$< x^R > = \frac{1}{\sum_{x^R = x^R_0 - t_0^{i-1} - \delta R} W[I^R](\sigma^{i-1}, \theta_0, x^R, y_0)} \sum_{x^R = x^R_0 - t_0^{i-1} - \delta R} x^R W[I^R](\sigma^{i-1}, \theta_0, x^R, y_0)$$

(10)

where $\sigma^{i-1} = \sigma (1 + \alpha^{i-1})$. The second moment is given by

$$< (x^L)^2 > = \frac{1}{\sum_{\delta x^L = -\delta} W[I^L](\sigma, \theta_0, x^L, y_0)} \sum_{\delta x^L = x^L_0 - \delta} (x^L)^2 W[I^L](\sigma, \theta_0, x^L, y_0)$$

and

$$< (x^R)^2 > = \frac{1}{\sum_{x^R = x^R_0 - t_0^{i-1} - \delta R} W[I^R](\sigma^{i-1}, \theta_0, x^R, y_0)} \sum_{x^R = x^R_0 - t_0^{i-1} - \delta R} (x^R)^2 W[I^R](\sigma^{i-1}, \theta_0, x^R, y_0)$$

(11)

Note that re-estimating the Wavelet transform at each iteration for different floating point locations $x^R_0 - t_0 + \delta x^R$ and floating point scales $\sigma (1 + \alpha)$ is too expensive. Here we will use the pre-computed wavelets at the near integer location to $x^R_0 - t_0^{n} + \delta x^R$ and near integer pre-computed scale to $\sigma (1 + \alpha^n)$. 


New estimate for the parameters

From equation 5 we derive the expected version of it with the new (to be estimated) parameters

\[ < x^R > - x_0^R = -t_0^i + (1 + \alpha^i) ( < x^L > - x_0^L ) \] (12)

where \( x_0^R = x_0^L - d(x_0^L, y_0^L) \) is the integer matching hypothesis. Similarly,

\[ < (x^R - x_0^R)^2 > = < (-t_0^i + (1 + \alpha^i) (x^L - x_0^L))^2 > \]
\[ = (t_0^i)^2 - 2t_0^i (1 + \alpha^i) ( < x^L > - x_0^L ) + (1 + \alpha^i)^2 < (x^L - x_0^L)^2 > \]
\[ \Downarrow \text{ using (12)} \]
\[ < (x^R - x_0^R)^2 > - ( < x^R - x_0^R > )^2 = (1 + \alpha^i)^2 ( < (x^L - x_0^L)^2 > - ( < x^L - x_0^L > )^2 ) \] (13)

We then can recover \( \alpha^i \) and \( t_0^i \) from

\[ 1 + \alpha^i = \sqrt{ < (x^R - x_0^R)^2 > - ( < x^R - x_0^R > )^2 } / < (x^L - x_0^L)^2 > - ( < x^L - x_0^L > )^2 \]
\[ t_0^i = (1 + \alpha^{i+1}) < x^L - x_0^L > - < x^R - x_0^R > \] (14)

Question 2

Note that we will work with \( \theta_0 = 0, \pi, \frac{\pi}{2} \), while \( \sigma = 1, 3, 6 \). So,

\[ W[I^L](\sigma_0, \theta_0 = 0, x_0^L, y_0) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I^L(\vec{x}) \frac{1}{\sigma_0} \left( e^{i \frac{\pi}{\sigma_0} (x-x_0^L)} - C_2 \right) e^{-\frac{1}{2\sigma_0} \left[ (x-x_0^L)^2 + (y-y_0)^2 \right]} d^2 \vec{x} \]

\[ W[I^L](\sigma_0, \theta_0 = \frac{\pi}{2}, x_0^L, y_0) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I^L(\vec{x}) \frac{1}{\sigma_0} \left( e^{i \frac{\pi}{\sigma_0} (y-y_0)} - C_2 \right) e^{-\frac{1}{2\sigma_0} \left[ (x-x_0^L)^2 + (y-y_0)^2 \right]} d^2 \vec{x} \] (15)

Thus, there are two angles and three scales (\( \sigma \)) estimates for the quantities \( < x^L > \), \( < x^R > \), \( < (x^L)^2 > \), \( < (x^R)^2 > \). We can average them before applying 14 so that we obtain more robust estimates of \( \alpha, t_0 \) Note that re-estimating the Wavelet transform at each iteration for different floating point locations \( x_0^R - t_0 + \delta x^R \) and floating point
scales $\sigma (1 + \alpha)$ is too expensive. Here we will use the pre-computed wavelets at the near integer location to $x^R_0 - t^n_0 + \delta x^R$ and near integer pre-computed scale to $\sigma (1 + \alpha^n)$.

For each integer hypothesis $x^L_0, x^R_0$ and a given window of size $[-\delta, \delta]$, we can make estimates of $\alpha^n$ and $t^n_0$ from iterating (14) for $n$ steps, starting from $\alpha^0 = 0$ and $t^0_0 = 0$. The final disparity is then, $D(x^L_0, y^L_0) = x^L_0 - x^R_0 + t^n_0$ at the center, and at the line segment it is $D(x^L_0 + \delta x^L, y^L_0) = (x^L_0, y^L_0) - \alpha^n \delta x^L + t^n_0$, with $\delta x^L \in [-\delta, \delta]$. For us, we consider $\delta = 18$ so that we can apply the previous homework wavelet filters.

Choose some locations at the image, where there is contrast data, and compare these results to the true disparity.