Solution 1

(a) Let us call the projected points on camera 1 $p_1 = [x, y]^T$ and on camera 2 $p_2 = [x', y']^T$ and the point in the world frame $P = [a, b, Z]^T$. We can then express p_1 and p_2 as the following equations. Note that $\vec{p_1}$ denotes p_1 in homoneous coordinates

$$\lambda_1 \vec{p_1} = KP = \lambda_1 [fa, fb, Z]^T \tag{1}$$

$$\lambda_2 \vec{p_2} = K(P - t) = \lambda_2 [f(a - t_x), fb, Z]^T \tag{2}$$

Converting from homogenous to regular coordinates, we find the following

$$p_1 = \begin{bmatrix} \frac{fa}{Z} \\ \frac{fb}{Z} \end{bmatrix} \tag{3}$$

$$p_2 = \begin{bmatrix} \frac{fa - ft_x}{Z} \\ \frac{fb}{Z} \end{bmatrix} \tag{4}$$

The discrepancy d in the x-coordinate is

$$d = x - x' = \frac{fa}{Z} - \frac{fa - ft_x}{Z} = \frac{ft_x}{Z}$$

(b) We can express p_1 and p_2 in terms of the point P shown below:

$$\lambda_1 \vec{p_1} = KP \tag{5}$$

$$\lambda_2 \vec{p_2} = K(P - t) \tag{6}$$

We can then find p_1 in terms of P.

$$P = \lambda_1 K^{-1} \vec{p_1} \tag{7}$$

Substituting this into the equation for p_2, we get the following

$$\lambda_2 \vec{p_2} = K(\lambda_1 K^{-1} \vec{p_1} - t)$$

$$\lambda_2 \vec{p_2} = \lambda_1 \vec{p_1} - Kt$$

Expanding the vectors we get

$$\begin{bmatrix} \lambda_2 x' \\ \lambda_2 y' \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} \lambda_1 x - Kt_x \\ \lambda_1 y \\ \lambda_1 \end{bmatrix}$$

Converting back to standard coordinates the following result is obtained.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x - \frac{ft_x}{\lambda_1} \\ y \end{bmatrix}$$

λ_1 can be found using the equation of a plane. Let $L = [\alpha, \beta, \gamma]^T$. The the equation of a plane is $L^T A = k$ where A is a point on the plane. Plugging the expressiong of P in terms of p_1 (Equation 7) equation yields:

$$\lambda_1 L^T K^{-1} \vec{p_1} = k$$

$$\lambda_1 = \frac{k}{L^T K^{-1} \vec{p_1}}$$

K^{-1} was calculated to be:

$$K^{-1} = \begin{bmatrix} 1/f & 0 & 0 \\ 0 & 1/f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Thus we can calculate λ_1 as the following:

$$\lambda_1 = \frac{k}{\frac{\alpha x}{f} + \frac{\beta y}{f} + \gamma}$$

Now $\vec{p_2}$ is:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x - \frac{f t_x (\frac{\alpha x}{f} + \frac{\beta y}{f} + \gamma)}{k} \\ y \end{bmatrix} = \begin{bmatrix} x - \frac{t_x}{k}(\alpha x + \beta y + \frac{\gamma}{f}) \\ y \end{bmatrix} \tag{8}$$

Thus the discrepancy is just

$$d = \frac{t_x \alpha}{k} x + \frac{t_x \beta}{k} y + \frac{t_x \gamma}{f}$$

Thus $a = \frac{t_x \alpha}{k}$ $b = \frac{t_x \beta}{k}$ and $c = \frac{t_x \gamma}{f}$.

(c) Proceeding the same way as the last part, we start with expressions of p_1 and p_2

$$\lambda_1 \vec{p_1} = KP$$

$$\lambda_2 \vec{p_2} = K(P - t)$$

Solving for P in terms of p_1,

$$P = \lambda_1 K^{-1} \vec{p_1}$$

Substituting into p_2,

$$\lambda_2 \vec{p_2} = K(\lambda_1 K^{-1} \vec{p_1} + t) = \lambda_1 \vec{p_1} + t$$

$$\lambda_2 \vec{p_2} = [x, y, 1]^T + [0, 0, -t_z]^T$$

$$\lambda_2 \vec{p_2} = [x, y, 1 - t_z]^T$$

$$p_2 = [\frac{x}{(1 - t_z)}, \frac{y}{(1 - t_z)}]^T$$

The discrepancy can be found as the following

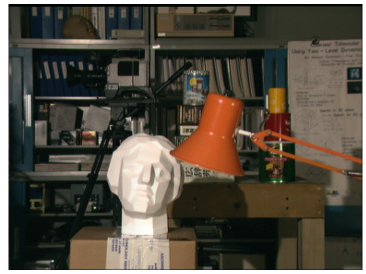
$$d = p_1 - p_2 = [x - \frac{x}{(1 - t_z)}, y - \frac{y}{(1 - t_z)}]^T = [\frac{-x t_z}{1 - t_z}, \frac{-y t_z}{1 - t_z}]^T$$

The discrepancy shows that the set of possible coordinates (x', y') is on a line passing through (x, y) in the direction of $< x, y >$.

Solution 2

All the parts of this problem use the following two images as input.

(a) Left image



(b) Right image

(a) Using a similar code for the `smatch` function, the following image was generated by simply taking argmin in the discrepancy dimension of the cost volume. Due to the lack of filtering, this disparity map is very noisy.

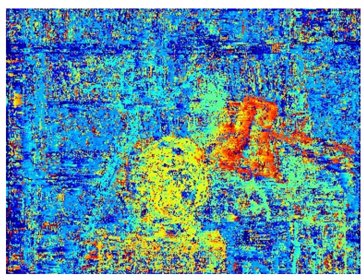


Figure 2: Disparity map using naive solution

(b) Using the bilateral filter solution code from PSET1 as a base, the disparity map is filtered, using the left image to generate the kernel. The result of this is shown below:

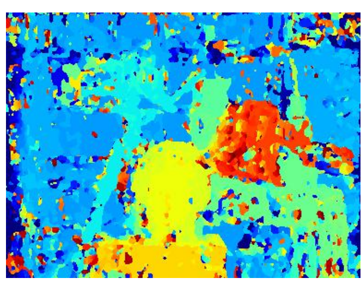


Figure 3: Disparity map bilaterally filtered

Solution 3

This problem uses the same source image shown in Figure 1a and 1b.

(a) Using the same cost volume generating function, the Viterbi algorithm was used to calculate an augmented cost volume, creating a different disparity map. This only works in the horizontal direction, hence the horizontal streaking, although much better than the naive solution.

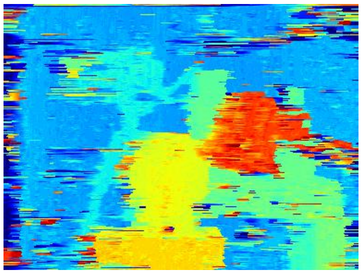


Figure 4: Disparity map using Viterbi

(b) To perform semi global matching, the same augmented cost volume is generated in four different directions, then aggregated and the `argmin` is taken to generate the disparity map. The implementation used first calculates the left-right and right-left directions in parallel, then calculates the up-down and down-up directions.

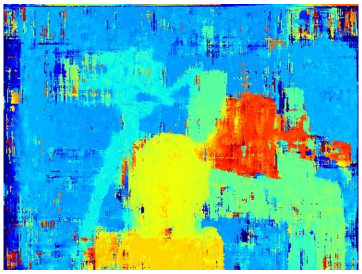


Figure 5: Disparity map using Semi-Global Matching

Solution 4

For the Lucas-Kanade method, the following equation must be solved.

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_x I_t \\ I_y I_t \end{bmatrix} \tag{9}$$

The solution for this can be found easily, and allows for much easier parallelization. The solution is shown below.

$$\begin{bmatrix} u \\ v \end{bmatrix} = -\frac{1}{\sum I_x \sum I_y - (\sum I_x I_y)^2} \begin{bmatrix} (\sum I_x I_t)(\sum I_y^2) - (\sum I_y I_t)(\sum I_x I y) \\ -(\sum I_x I_t)(\sum I_x I_t) + (\sum I_y I_t)(\sum I_x^2) \end{bmatrix} \tag{10}$$

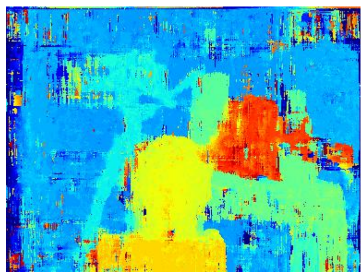


Figure 6: Quiver plot from Lucas-Kanade method of calculating optical flow.

Information

This problem set took approximately 15 hours of effort.

I discussed this problem set with:

- Myself

I also got hints from the following sources:

- Wikipedia article on matrix calculus at https://en.wikipedia.org/wiki/Matrix_calculus
- Read numpy tutorial from https://docs.scipy.org/doc/numpy-1.13.0/user/basics.broadcasting.html
- For stackings arrays multiple https://stackoverflow.com/questions/22634265/python-concatenate-or-clone-a-numpy-array-n-times