EC4216 Computer Vision
GIST
Fall 2023
Dongjun Nam

Programming Assignment 1 Report

Dongjun Nam

june6423@gm.gist.ac.kr

This assignment requires performing denoising using burst shot images of the same exposure time. Assuming that the noise in each image follows a Gaussian distribution, you can reduce the noise by averaging over several correctly aligned images. To do this, we need to take the following steps. (i) Construct Cost Volume. (ii) Derive disparity map using Semi Global Matching. (iii) Warp each image and get the aggregate image. (iv) Calculate performance.

1. *Construct Cost Volume.*

   We know that there are 7 images coming in as input, centered on image 4, with images 1,2,3 on the left and 5,6,7 on the right. To know the disparities that exist between the images, we need to find the matrix that computes the pixel-wise similarity for all disparities. Let's call this the cost volume. Since we know the positional relationship between the reference image (image 4) and the other images, we know which direction to give the disparity. Since images 1, 2, and 3 are to the left of the reference image, we can find the right cost volume, and since images 5, 6, and 7 are to the right of the reference image, we can find the left cost volume.

   I used the Brighfield-Tomasi Dissimilarity for pixel-wise similarity. Brighfield-Tomasi Dissimilarity does not compare all pixels 1:1 but instead compares pixels that are extended by their neighboring half pixels to find the l1 loss. Given the assumption that the image only moves along the x-axis, the expansion of a half-pixel is only considered along the x-axis. Since all pixels have only integer coordinates, the expansion of a half-pixel was obtained as the average of the original and adjacent coordinates. ($I_r^+(x_r) = \frac{I_r(x_r)+I_r(x_r+1)}{2}$) Also, the intensity of a pixel was calculated in greyscale without considering the RGB channel.

   We can obtain an intermediate disparity map taking argmin over the disparity axis with the cost volume. However, it is very unstable. So we used the sliding window method to obtain the average. The difference in the disparity map with and without sliding window is visualized in Figure 1. The left image is an intermediate disparity map without a sliding window and the right image uses a sliding window. We can obtain a denser intermediate disparity map without a sliding window. However, the performance after warping is higher with the sliding window. We'll discuss this later in this report.

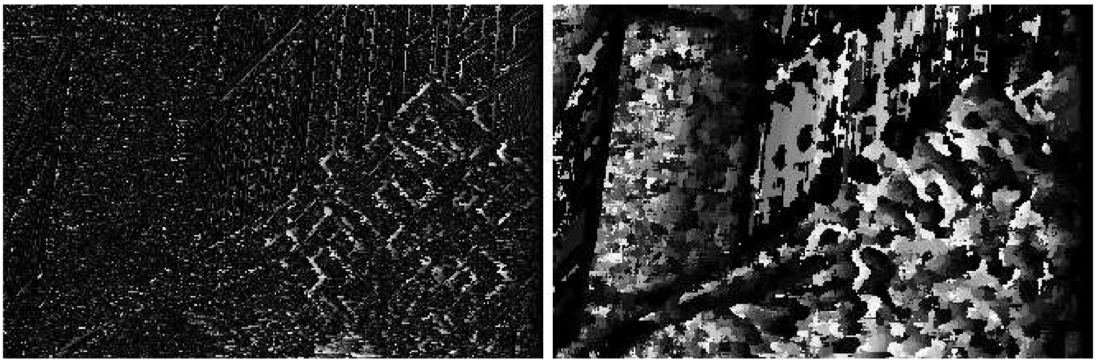

Figure 1: Intermediate disparity map with and without sliding window

2. *Semi Global Matching*

   Based on the obtained cost volume, perform semi-global matching by dynamic programming for 8 directions. The intermediate disparity map was only evident for edges, but by performing semi-global matching, a clean disparity map at the object level can be obtained. The formula for dynamic programming is as follows. Where $d^*(p,d)$ is final disparity and $D(p,d)$ is pixel-wise disparity(Cost volume).

$$d^*(p,d) = \arg\min_d S(p,d) = \arg\min_d \sum_r L_r(p,d) \tag{1}$$

$$\begin{aligned} L_r(p,d) = D(p,d) + \min\{&L_r(p-r,d), L_r(p-r,d-1) + P_1, L_r(p-r,d+1) + P_1, \\ &\min_i L_r(p-r,i) + P_2\} - \min_k L_r(p-r,k) \end{aligned} \tag{2}$$

   Figure 2 and Figure 3 visualizes the final disparity map comparing to its intermediate disparity map.

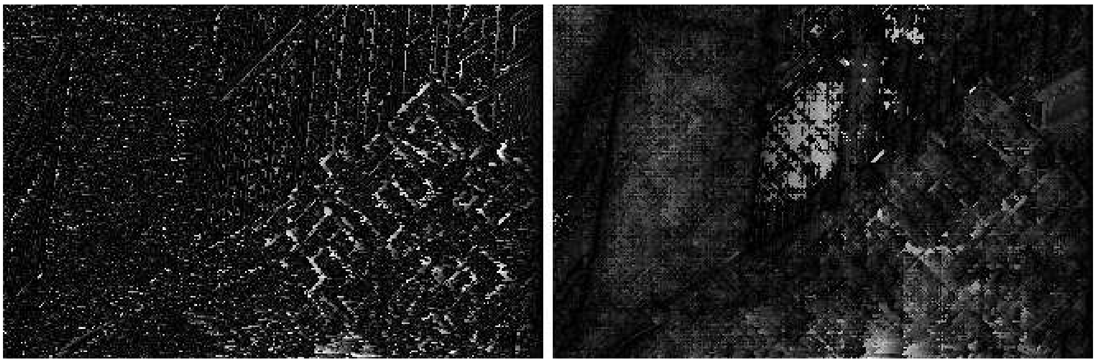

Figure 2: Intermediate disparity map(left) and Final disparity map(right) without sliding window

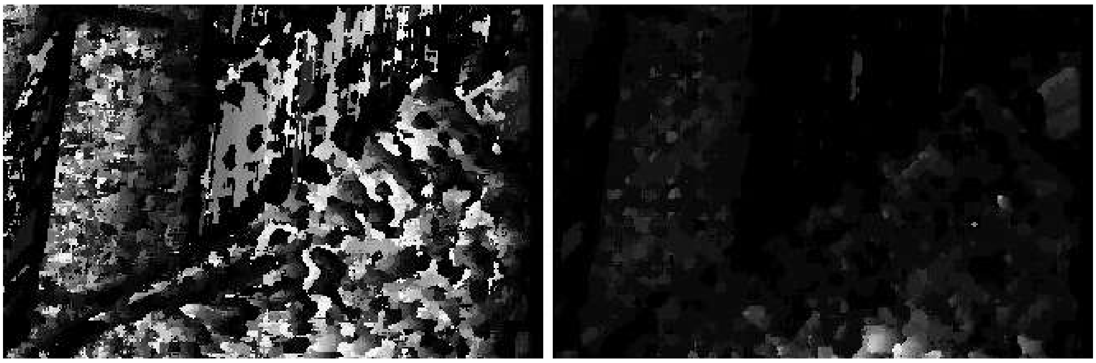

Figure 3: Intermediate disparity map(left) and Final disparity map(right) with sliding window

3. *Weighted median filter*

   I applied a weighted median filter to obtain clearer disparity map. I used a weighted median filter because a median filter can damage edges. Here's the equation for weighted median filtering.[1]

$$h(\mathbf{x}, i) = \sum_{\mathbf{x}' \in \mathcal{N}(\mathbf{x})} w(\mathbf{x}, \mathbf{x}')\delta(V(\mathbf{x}') - i) \tag{3}$$

   And We can find the value of $w(\mathbf{x}, \mathbf{x}')$ by referring to another paper.[2]

$$w(\mathbf{x}, \mathbf{x}') \propto \exp\Big\{-\frac{|x - x'|^2 + |y - y'|^2}{2\sigma_1^2} - \frac{|\mathbf{I}(x) - \mathbf{I}(x')|^2}{2\sigma_2^2}\Big\} \tag{4}$$

   x is position of the pixel and $\mathbf{I}(x)$ is the intensity of the pixel. I set $2\sigma_1^2 2 = \sigma_2^2 = 1$ because its value does not change the result. Using these weights, the weighted median value was calculated based on the neighboring 3*3 pixels.

   However, when I tried applying a weighted median filter, I found that it actually decreased performance, so I didn't use it. The results are shown in Table 1, and we'll discuss later in this report.

4. *Warping*

   Based on the obtained disparity map, we need to warp all the images to the reference image to create a single image. For every pixel in the other image, we consider the disparity and consider which pixel in the reference image it matches. We create an array count of the size of the reference image and count how many pixels are warped to each pixel in the reference image to get the pixel-wise mean. I cropped the bounds of the images that were prone to error. The cropped size is 24 pixels, the same as the maximum disparity. Warped images are shown in Figure 4.

5. *Result*

   Figure 4 shows an aggregated warped image with and without a sliding window. Warped image without sliding windows(left) has sharp edges but it is more noisy, while the image with sliding windows(right) has blurry edges but is less noisy.

Figure 4: Warped image without sliding window(left) and with sliding window(right)

   Based on the aggregated warped image(estimated image), I calculated MSE and PSNR to measure it's performance. MSE and PSNR formula are shown as below. Where I_{gt} is ground truth image and $\hat{I}$ is estimated image and R = 255.

$$MSE = \frac{\sum(I_{gt} - \hat{I})^2}{H * W * C} \tag{5}$$

$$PSNR = 10log_{10}(\frac{R^2}{MSE}) \tag{6}$$

6. *Ablation Study*

   The ablation study allowed us to understand which parameters contribute to the performance improvement: the sliding window is 7*7 and the weighted median filter is not used as shown in Table 1. Nevertheless, the performance is lower than the guideline, which we believe is due to the problem with semi-global matching. The final disparity map when the sliding window is not used is not smooth in 8 directions, which may have contributed to the performance drop.

   This may explain why the non-directional sliding window method showed higher PSNR despite the blurred edges. The weighted median filter actually worsened the performance by damaging the edges. There is room to improve performance with more sophisticated weight settings and box size adjustments, so this is an area for further research.

Noise Level	*Window size*	*Weighted median filter*	*MSE*	*PSNR*
25	1*1	X	300.72	23.35
25	1*1	3*3	372.26	22.42
25	**7*7**	**X**	**293.99**	**23.45**
25	7*7	3*3	301.95	23.33
10	7*7	X	226.32	24.58
12.5	7*7	X	232.97	24.46
15	7*7	X	232.97	24.46

Table 1: Compare performance in various parameters

My hyper parameter used in the code are listed below. (Best performance setting in Noise level 25)

- Number of frames = 7
- Maximum disparity $d = 24$ (0 to 23)
- Sliding window size = 7
- Sliding window stride = 1
- Sliding window padding = None
- Number of directions in SGM = 8
- P1 and P2 Value in SGM = 5, 150
- Cropped boundary size = 24
- Weighted median filter box size = 3
- $2\sigma_1^2 = 1$
- $2\sigma_2^2 = 1$

7. *Reference*

[1] Z. Ma et al. "Constant time weighted median filtering for stereo matching and beyond", IEEE International Conference on Computer Vision 2013

[2] D. Sun, S. Roth, and M. Black. Secrets of optical flow estimation and their principles. In CVPR, 2010.