Model fitting and multi-stereo vision

Qi Ma, 20-960-225, qimaqi@student.ethz.ch

December 3, 2021

Abstract

This assignment is most difficult one I ever had but I really learned a lot by analyzing the paper and writing the code

1 Model fitting

The first task is model fitting when a very representative of line fitting case is introduced.

Q1 Write down the ground truth, estimation from least-squares and estimation from RANSAC in the report? **A1**

- The ground truth of the line is $k = 1, b = 10$
- The estimation of the line is $k = 0.999, b = 10.084$
- The least-squares of the line is $k = 0.616, b = 8.962$

This result figure shows below:

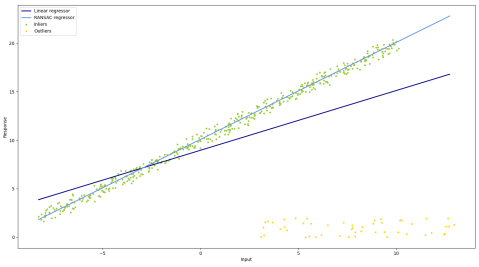

Figure 1: Result figures.

```
Estimated coefficients (true, linear regression, RANSAC):
1 10 0.6159656578755459 8.96172714144364 0.9995171354891829 10.084162309381236
```

Figure 2: exact result of ransac fitting.

Inspired by the discussion in Moodle. The straight way to calculate the number of inlier is vertical distance: $|y - kx - b|$ and since we do not only have noise in x and also y so calculating point to line space is more reasonable. $d = \frac{|Ax+By+C|}{\sqrt{A^2+B^2}}$. But the result belows shows worse. This is caused by the point to line space is smaller than vertical distance. So the threshold distance should also be more tightly.

```
Estimated coefficients (true, linear regression, RANSAC):
1 10 0.6159656578755456 8.96172714144364 1.0911794882209374 9.415441911007154
```

Figure 3: Different way to calculate the inliers.

2 Multi-view stereo vision

This tasks oriented at the advanced depth estimation which deal with 3D volume but in this case since most student need computation-light task so the way to propagate the 3D volume is actually 2D operation. But in practice the performance is proved to be good.

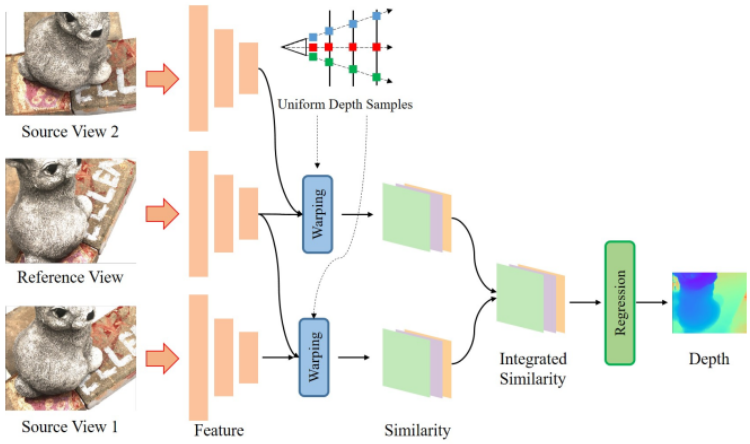

Figure 3.1: Detailed Structure of our method.

Figure 4: MVS Net structure.

The actual network process

1. feature extraction
2. ref feature warp to source feature with multiple Depth
3. group-wise correlation and result in similarity in different depth volume [2]
4. similarity aggregation
5. similarity regression and softmax to build probability volume
6. depth regression

2.1 Feature extraction

In this step we use multiple ConvBnReLu layer to extract the feature and downsample the resolution. Two thoughts here 1) We use weight-sharing convolution since we only care the feature extraction without any involving with feature from another stereo view. 2) We may use a SPP module to increase the performance of multi-scale depth estimation.

As shown above, there are some far away keypoints in the image which can well expalin the result before.

2.2 Differentiable Warping

Q1 For pixel p in the reference feature and a depth value dj, write down the equation of corresponding pixel pi,j := pi(dj) in the report, which is the projection of p in source view i with depth value as dj (the coordinates of pixels are given in homogeneous coordinates).

A1 I will solve this problem in three steps, first we list the 2D points - 3D points and depth relationship and then I transfer all the points to homogeneous coordinates and lastly I will finish the correspondence.

First I show the result and then I will show the solution step by step:

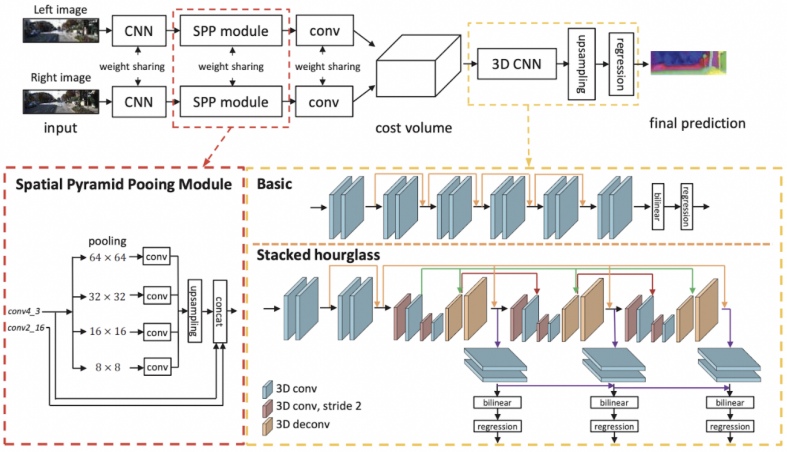

Figure 5: Possible we can use PSMNet SPP module [1]

$$\lambda_s \begin{bmatrix} u_s \\ v_s \\ 1 \end{bmatrix} = \begin{bmatrix} K_s R_{r,s} K_r^{-1} & K_s(t_s - R_{r,s} t_r) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} depth_r * u_r \\ depth_r * v_r \\ depth_r \\ 1 \end{bmatrix} \tag{1}$$

After checking the MVSNet [3] paper the equation is equal to the reverse equation form paper:

$$H_i(d_j) = K_i R_i (I - (t_1 - t_i) n_1^T / d_j) R_1^T K_1^T \tag{2}$$

Step 1) We write down the same 3D points correspondence in source view s and reference view r. Since we already known the depth value.

$$\lambda_s \begin{bmatrix} u_s \\ v_s \\ 1 \end{bmatrix} = K_s [R_{w,s} | t_{w,s}] P_w \tag{3}$$

$$depth_r \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = K_r [R_{w,r} | t_{w,r}] P_w \tag{4}$$

Since the stack matrix $[R|t]$ is a 3x4 matrix and not invertible. So we can use homogeneous coordinate and rewrite the equation like below:

$$\begin{bmatrix} \lambda_s * u_s \\ \lambda_s * v_s \\ \lambda_s \\ 1 \end{bmatrix} = M_{w,s} \begin{bmatrix} P_w \\ 1 \end{bmatrix} \tag{5}$$

$$\begin{bmatrix} depth_r * u_r \\ depth_r * v_r \\ depth_r \\ 1 \end{bmatrix} = M_{w,r} \begin{bmatrix} P_w \\ 1 \end{bmatrix} \tag{6}$$

And the M matrix is K matrix multiple into the extrinsic matrix

$$M_{w,s} = \begin{bmatrix} K_s & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{w,s} & t_{w,s} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} K_s R_{w,s} & K_s t_{w,s} \\ 0 & 1 \end{bmatrix} \tag{7}$$

where the M matrix is 4x4 homogeneous matrix and it is invertible because K is invertible so we reformulate the equation with replacing the M inverse.

$$M_{w,r}^{-1} = \begin{bmatrix} R_{w,r}^T K_r^{-1} & -R_{w,r}^T t_{w,r} \\ 0 & 1 \end{bmatrix} \tag{8}$$

$$\begin{bmatrix} \lambda_s * u_s \\ \lambda_s * v_s \\ \lambda_s \\ 1 \end{bmatrix} = M_{w,s} M_{w,r}^{-1} \begin{bmatrix} depth_r * u_r \\ depth_r * v_r \\ depth_r \\ 1 \end{bmatrix} \tag{9}$$

$$= \begin{bmatrix} K_s R_{r,s} K_R^{-1} & K_s(t_s - R_{r,s} t_r) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} depth_r * u_r \\ depth_r * v_r \\ depth_r \\ 1 \end{bmatrix} \tag{10}$$

However this is just seems to be complicated so we can extract a new "rotation matrix" and "translation" from $M_{sw}M_{rw}$ and get R_new and t_new which is from the [:3,:3] and [:3,3:4]. Then we can get the transform used in code:

$$normalizeterm \begin{bmatrix} u_s \\ v_s \\ 1 \end{bmatrix} = R_{new} \begin{bmatrix} depth_r * u_r \\ depth_r * v_r \\ depth_r \end{bmatrix} + t_{new} \tag{11}$$

2.3 Training

The training screenshot is shown below: As we can see the abs-depth-error decrease to about 15 in first epoch

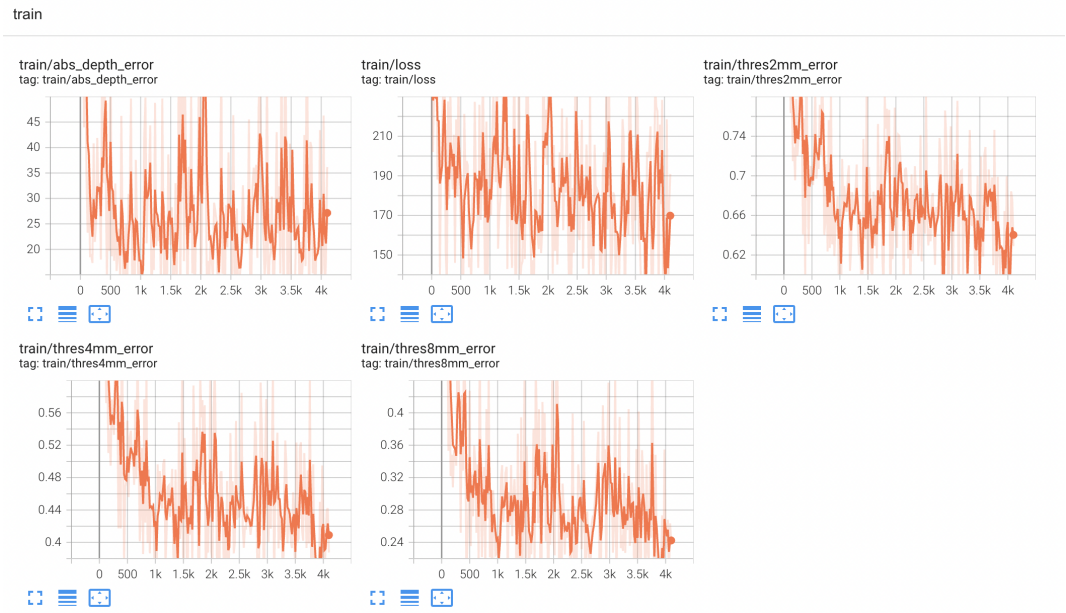

Figure 6: The train loss converges.

The validation screenshot is shown below: As we can see the abs-depth-error converges.

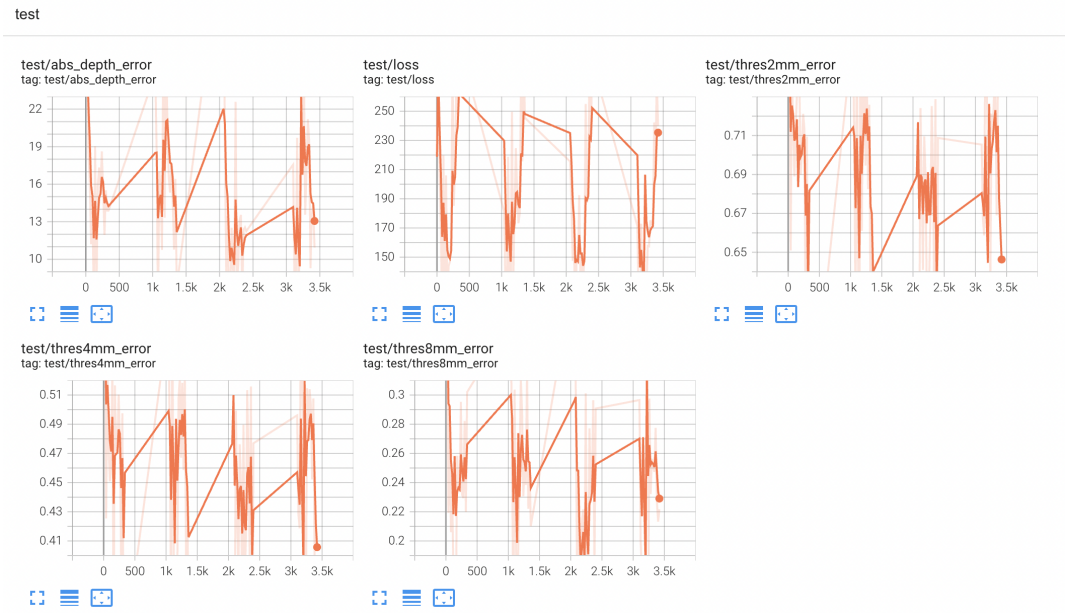

Figure 7: The validation loss converges.

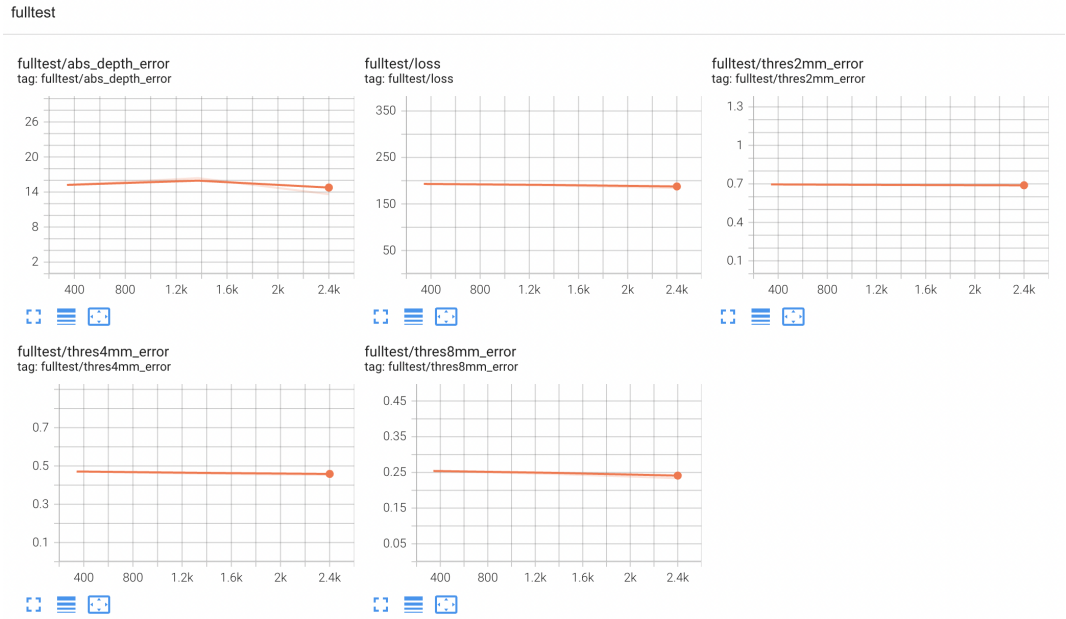

Figure 8: The validation loss converges in first epoch.

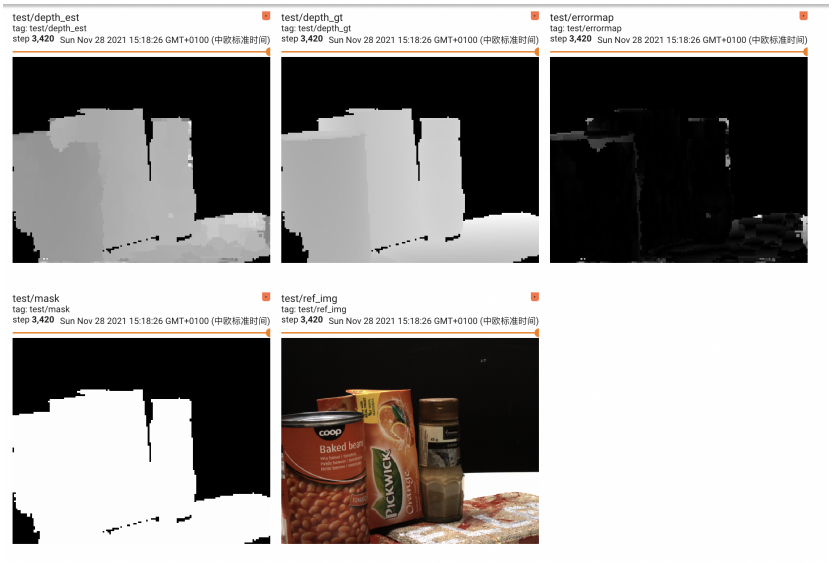

Figure 9: The validation image.

Figure 10: The validation image.

The output of validation shows great compared to ground truth

2.4 Visualization of ply file

The visualization of scan1 and scan9 shows good 3D point clouds as below:

Figure 11: visualization of scan ply.

Figure 12: visualization of scan ply..

Q2 Explain what geometric consistency filtering is doing in the report.

A2 The geometric consistency filtering is basically filter out the depths which is not consistent in multi-view depth estimation. This is achieved by first project the depth in a reference pixel p1 through its depth d1 to pixel pi in another view, then reproject pi back to the reference image by pi's depth estimation di. If the reprojected pixel is close to the reference pixel and the reprojected depth have small difference between the reference depth the pixel, then the pixel and depth estimation is geometric consistent.

Q3 In our method, we sample depth values, $d_{j}{}_{j=1}^{D}$, that are uniformly distributed in the range [DEPTH MIN, DEPTH MAX]. We can also sample depth values that are uniformly distributed in the inverse range [1/DEPTH MAX, 1/DEPTH MIN]. Which do you think is more suitable for s?

A3 The large scene will have a big variance in depths so uniform distribution will result in two questions. 1) Depth intervals will become large and so estimated depth will inevitably have gap between ground truth depth information, which may confuse the network because the depth resolution. However, increasing the D value will handle this. 2) The estimated depth may range from small number to a very large number and the network is harder to converge since some wrong prediction will lead to large error. In this case when we sample the inverse of depth will bring in a normalization effect and will be better for training.

But if we just sample this depth value in inverse range and then we inverse it back and input to the network. The new distribution is not uniform distributed in depth range but will have more depths candidate close to the $DEPTH_{Min}$ and more sparse in long range. From this point we can say such sample is also better for large-scale since the depth intervals will become bigger and bigger in large-scale scene with the range.

Q4 In our method, we take the average while integrating the matching similarity from several source views. Do you think it is robust to some challenging situations such as occlusions?

A4 Simply averaging the matching similarity is not robust to occlusion case for example if one object is occluded in one source view but not the other, then the matching similarity will be small even it matches actually good except for occlusion. So the overall matching similarity will be small. But the robustness to the occlusion is to some extend achieved by geometric consistencies filter. If certain pixel is occluded then the difference between the original depth estimation and reprojected depth estimation will be different. Moreover, the photometric consistency is also a simple but strong filter for outliers.

References

[1] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5418, 2018.

[2] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3273–3282, 2019.

[3] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018.