3D Rigid Motion Estimation

Hangting Cao, Ang Cao, Xingcheng Yuan, Dong Chen
Unversity of Michigan
{hangting, ancao, ybenny, donchen}@umich.edu

Abstract

In this project, we implement the deep rigid instance scene flow (DRISF) algorithm from scratch. This method is intended to tackle the problem of 3D scene flow estimation in the context of self-driving. It leverages strong structure priors as well as deep learning techniques such that a scene motion can be composed by the motion of the robot and the 3D motion of the actors in the scene. The problem is formulated as energy minimization in a deep structure model, which is solved efficiently using a Gaussian-Newton solver. Our experiment results show that we have successfully implemented this algorithm, despite certain differences with that in the original paper. Also, decent performance on KITTI dataset is achieved. Finally, several qualitative and quantitative analyses are performed to evaluate our work.

1. Introduction

By analogy with optical flow, scene flow problems can be interpreted as estimation of the three-dimensional motion field of points in a scene [1], which can be computed from two stereo pairs consecutive in time; in this sense, scene flow is the 3D counterpart of optical flow. Scene flow has great versatility in a wide range of real-world applications. Also, it enables obtaining insights into overall geometry and composition, and more importantly, motion of a scene. In particular, this can serve as an advantageous tool for the area of robotics systems, especially for self-driving cars. From this prospective, scene motion can be explained by the motion of an autonomous system, a.k.a. ego-car. Meanwhile, the presence of dynamic objects that typically move rigidly can be exploited as strong priors. These facts are commonly used by early structure prediction approaches to fit a piece-wise rigid representations of scene motion [2, 3, 4]. Although these methods have decent performance in estimation of motion fields, they mostly require minutes for processing each scene; hence, they are not suitable for practical traffic scenarios.

As aforementioned, scene flow estimation generally involves a few low-level tasks, e.g. stereo estimation [5] and optical flow prediction [6]. Despite that deep learning has been proved to achieve impressive real-time results in such tasks, their output is not structured and thus fail to capture relationships between estimated variables. For example, they are not able to guarantee consistency of estimates produced by the pixels of a given object. This undoubtedly would bring security risks.

To improve the previous algorithms, we implement a novel deep rigid instance scene flow model, in which we leverage 3D structure relationships as strong priors and propose a Gaussian-Newton solver to minimize the well designed energy function. Our major work is:

- We re-implement the entire pipeline from scratch and achieve decent performance, given the reference paper does not provide source code or implementation details.
- We make extensive experiments to demonstrate the effectiveness of the structure optimization solver, and analyze the contribution of each energy function term by ablation studies.
- We explore the pros and cons of this algorithm, and analyze the gaps of performance between original paper and our implementation.

2. Related Work

2.1. Stereo

Traditional approaches [7, 8] for stereo problems are characterized by three components: computation of patch-wise features, construction of cost volumes, and final post-processing. Although modern methods that make use of CNNs for predicting whether two patches are a match manifest desirable performance for a variety of challenging benchmarks, they have limited usefulness due to expensive computation involved.

Luo et al. [8] tackle this problem by using a correlation layer to extract marginal distributions over all possible disparities; however, due to the need of smoothing estimations, such remedy still involves the component of post processing, thus posing significant limitation for computation

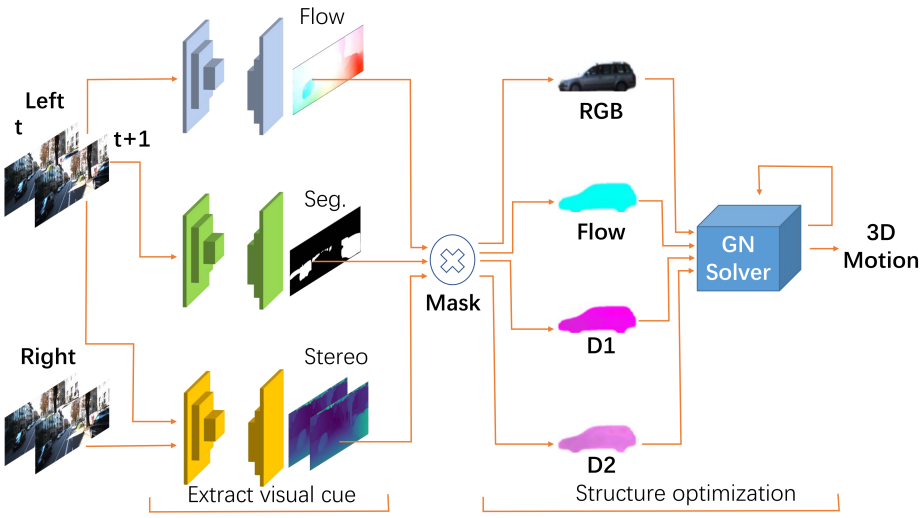

Figure 1. **Overview of our approach:** Given two consecutive stereo images, we first estimate the flow, stereo, and segmentation (Sec. 3.1). The visual cues of each instance are then encoded as energy functions (Sec. 3.2) and passed into the Gaussian-Newton (GN) solver to find the best 3D rigid motion (Sec. 3.3).

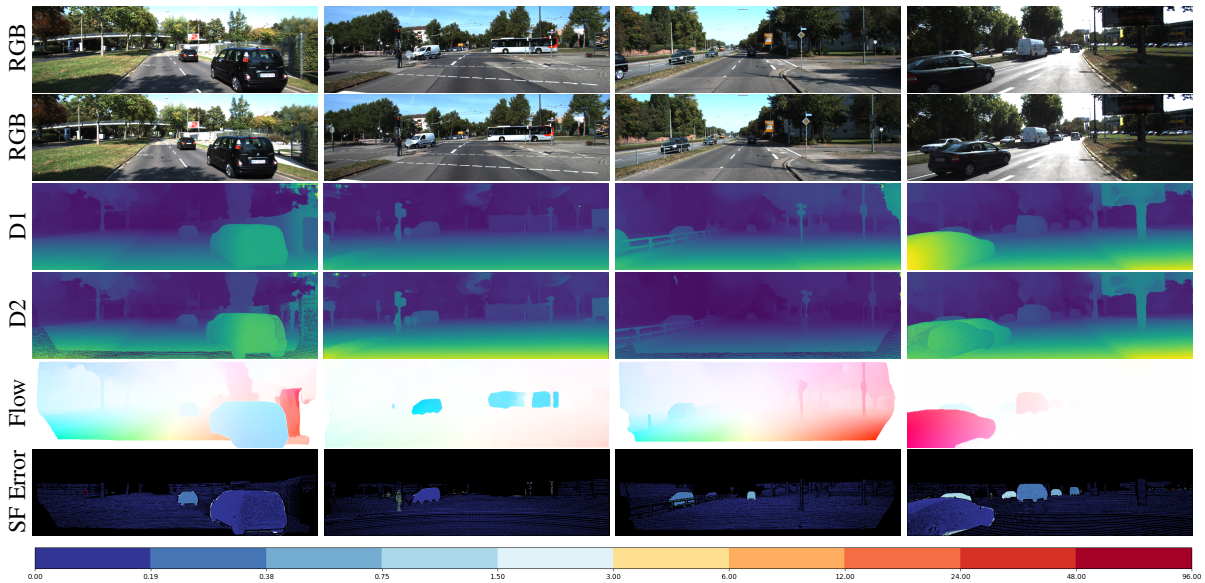

Figure 2. **Qualitative results:** Pipeline of our model and the performance. Images mostly have a small error of scene flow estimation on both foreground objects and background. Certain limitations on the edges objects possibly caused by small misalignment of the ground truth from the dataset.

efficiency. Subsequent improvements thereby focused on regression of subpixel disparities directly from the image pair. For example, in DispNet network [5], a 1D correlation layer is used to approximate cost volumes while later layers are designed for implicit aggregation. Furthermore, Kendall et al. [9] propose an even more complicated approach that exploits 3D convolution for regularization and includes a differentiable soft argument of the minimum to obtain subpixel disparities. The up-to-date algorithm PSMNet [10], which features the incorporation of Pyramid spatial pooling and stacked hourglass [11], has state-of-the-art computation performance for stereo tasks. Therefore, we use this network for the task of stereo estimations.

2.2. Optical flow

As a classical task in computer vision, optical flow dates back a couple of decades ago. Such task is generally based on the minimization of energy functions (also known as cost functions). Originally, when optical flow was first introduced by Horn and Schunck [12], the energy cost is defined

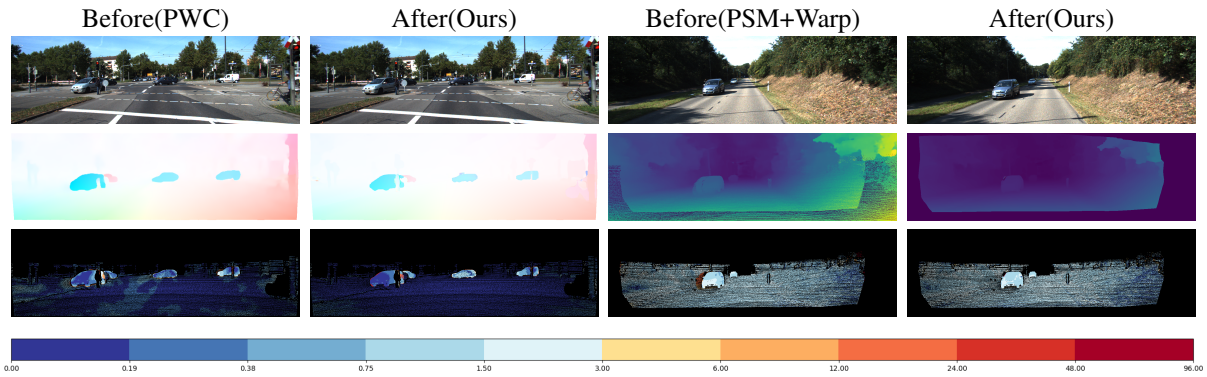

Figure 3. Comparisons of our proposed architecture

as a combination of only a data term and a smoothness term, which is solved based on variational inference. Recently, the well-known technique of deep learning, has been employed to improve performance in matching accuracy significantly. As in early algorithms used for aforementioned stereo problems, the post processing component [13] is still indispensable because of the sparsity of matching results; hence, this is prone to expensive computation.

Flownet [6] was first introduced to improve computation efficiency for optical flow problems. In its successive algorithm, Flownet2 [14], multiple networks are stacked for iterative refinement of estimated flows and a differentiable warping operation is proposed for compensating for large displacements, at the cost of a resulting large network. Although SpyNet [15] is capable of reducing network size via the replacement of warping operation with spatial pyramid network, they have a slightly worse performance. However, recent improvements of faster and smaller model size induced by PWC-Net [16] as well as Lite-Flownet [17] have successfully tackled these problems by combining previous ideas with pyramid processing technique and the concept of cost volume. Specifically, the latest PWC-Net is adopted for our optical flow task.

2.3. Scene flow

Scene flow can be interpreted as a combination of stereo and 2D optical flow described above in the sense that it gives information about 3D motion of a given point in space. In this regard, scene flow problems are also originally solved based on variational inference.

One primary limitation of this traditional method is that large motions can cause significant errors in real world applications. The family of algorithms based on slanted planes [18, 2, 19, 3] are thus proposed to improve robustness by decomposing the scene into a number of small rigidly- moving planes. Later proposed algorithms tend to incorporate various visual cues. Specifically, at the advantage of recognition cues, a network built by Behl et al. [20] are capable of establishing decent object correspondences across a wide range of scenarios. Another even advanced method is proposed by Ren et al. [21], which features the use of a variety of visual cues and conditional random fields. As with many algorithms in the area of computer vision, this method also has the shortcoming of demanding computation. The model scene flow model that we use in this work improves it substantially via formulating a less complicated optimization task, which desirably yields significant speed-up in computation in comparison with any previous algorithms, e.g. scene slow can be computed in less than a second.

3. Method

We will describe the algorithm and method we implemented in this section. The method starts with three different visual cues-namely segmentation, stereo and flow-and then formulate the scene flow by minimizing the energy functions on different prospective. We first describe how the three types of cues are obtained, then how the task of scene flow is composed of energy minimization problems, and finally how the optimization is performed.

3.1. Visual Cues

3.1.1 Segmentation: Mask R-CNN

Mask R-CNN is a proposal-based classification network based on Faster-RNN [22]. The first stage is a Region Proposal Network (RPN) that proposes several candidate objects bounding boxes. The second stage is to extract the features from these proposals and to output the class, the box offset and a binary mask for all possible classes in parallel. Mask R-CNN, in our method, will provide object bounding boxes on the image to facilitate the scene flow estimation.

3.1.2 Stereo: PSM-Net

PSM-Net is a neural network used for depth estimation. It contains a convolution layer module for feature extraction,

a novel Spatial Pyramid Pooling (SPP) module to determine the context relationship and a 3D cost volume module for implicit cost volume aggregation and regularization [10]. After that, a stacked hourglass networks, an encoder-decoder architecture, is implemented to help learn more context information. PSM-Net outperforms other methods on the object boundaries and is able to produce disparity images with very small error.

3.1.3 Optical Flow: PWC-Net

PWC-Net is an effective CNN model used for flow estimation [16]. The network works with three classical principles: pyramidal processing, warping, and the use of a cost volume, similar to PSM-Net for disparity estimation. Pyramidal processing helps encode image features with large context, the progressive warping reduces the cost of building cost-column through a coarse-to-find scheme and cost column can better improve flow estimation by sharpening the boundaries. PWC-Net produces the flow estimation on the two consecutive 2D images.

3.2. Energy Formulation

Once obtained the visual cues mentioned above, we calculate the 3D motion of each instance by minimizing the energy functions. Let $\mathcal{L}^0$, $\mathcal{R}^0$, $\mathcal{L}^1$, $\mathcal{R}^1$ denote the input stereo pairs captured from two consecutive time steps t_0 and t_1. Let $\mathcal{D}^0$, $\mathcal{D}^1$ be the estimated disparity map obtained from PSM-Net, $\mathcal{F}_\mathcal{L}$, $\mathcal{F}_\mathcal{R}$ be the estimated flow obtained from PWC-Net, and $\mathcal{S}^0_\mathcal{L}$ be the instance segmentation computed on the left image $\mathcal{L}^0$. We use $\boldsymbol{\xi} \in se(3)$, the Lie-algebra associated with $SE(3)$, to represent the 3D rigid motion. For each instance $i \in \mathcal{S}^0_\mathcal{L}$, we find $\boldsymbol{\xi}$ minimizing the weighted combination of photometric error, rigid fitting and flow consistency. Denote $\mathcal{I} = \{\mathcal{L}^0, \mathcal{R}^0, \mathcal{L}^1, \mathcal{R}^1, \mathcal{D}^0, \mathcal{D}^1, \mathcal{F}_\mathcal{L}, \mathcal{F}_\mathcal{R}\}$ as the input stereo pairs and visual cues. We denote the set of pixels belonging to instance i as $P_i = \{\mathbf{p} | \mathcal{S}^0_\mathcal{L}(\mathbf{p}) = i\}$. Then, the optimization problem is formulated as follows:

$$\min_{\boldsymbol{\xi}} \lambda_{\text{photo},i} E_{\text{photo},i}(\boldsymbol{\xi}; \mathcal{I}) + \lambda_{\text{rigid},i} E_{\text{rigid},i}(\boldsymbol{\xi}; \mathcal{I}) + \lambda_{\text{flow},i} E_{\text{flow},i}(\boldsymbol{\xi}; \mathcal{I}), \quad (1)$$

where $\lambda_{\cdot,i}$ are the weights. It finds a balance between three terms and next we introduce each term separately.

Photometric Error. This term encodes the fact that one point should have similar appearance at different time steps. For each pixel $\mathbf{p} \in P_i$ in $\mathcal{L}^0$, we compare the photometric value with that of the corresponding pixel in $\mathcal{L}^1$:

$$E_{\text{photo},i}(\boldsymbol{\xi}; \mathcal{I}) = \sum_{\mathbf{p} \in P_i} \alpha_\mathbf{p} \rho(\mathcal{L}^0(\mathbf{p}) - \mathcal{L}^1(\mathbf{p}')), \quad (2)$$

where $\alpha_\mathbf{p} \in \{0, 1\}$ is the indicator function denoting which pixel is an outlier. $\mathbf{p}'$ is the corresponding pixels in $\mathcal{L}^1$, obtained by inverse depth warping followed by a rigid transform $\boldsymbol{\xi}$:

$$\mathbf{p}' = \pi_\mathbf{K}(\boldsymbol{\xi} \circ \pi^{-1}_\mathbf{K}(\mathbf{p}, \mathcal{D}(\mathbf{p}))), \quad (3)$$

where $\pi_\mathbf{K}(\cdot)$: $\mathbb{R}^3 \rightarrow \mathbb{R}^2$ is the perspective projection function given known intrinsic matrix $\mathbf{K}$ and $\pi^{-1}_\mathbf{K}(\cdot, \cdot)$: $\mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}^3$ is the inverse depth warping that gets a 3D point utilizing a 2D point and its associated disparity; $\boldsymbol{\xi} \circ \mathbf{x}$ transforms a 3D point with transformation $\exp(\boldsymbol{\xi})\mathbf{x}$, where $\exp(\boldsymbol{\xi})$: $\mathbb{R}^6 \rightarrow \mathbb{R}^{4\times4}$ is the exponential map. ρ is a robust error function and we use the generalized Charbonnier function $\rho(x) = (x^2 + \epsilon^2)^\alpha$ for all the energy terms, where $\alpha = 0.45$ and $\epsilon = 10^{-5}$.

Rigid Fitting. Given correspondences $\{(\mathbf{p}, \mathbf{q} = \mathbf{p} + \mathcal{F}_\mathcal{L}(\mathbf{p})) | \mathbf{p} \in P_i\}$, this term encourages that the transformed 3D points calculated using the estimated 3D rigid motion $\boldsymbol{\xi}$, should be similar to the 3D points obtained by applying the inverse depth warping on $\mathbf{q}$. Specifically, it is formulated as

$$E_{\text{rigid},i}(\boldsymbol{\xi}; \mathcal{I}) = \sum_{(\mathbf{p},\mathbf{q})} \alpha_\mathbf{p} \rho(\boldsymbol{\xi} \circ \pi^{-1}_\mathbf{K}(\mathbf{p}, \mathcal{D}^0(\mathbf{p})) - \pi^{-1}_\mathbf{K}(\mathbf{q}, \mathcal{D}^1(\mathbf{q}))).$$

Flow Consistency. This term encourages the projection of the 3D rigid motion to be similar to the original flow estimation got from PWC-Net. We compare the optical flow and the structured rigid flow:

$$E_{\text{flow},i}(\boldsymbol{\xi}; \mathcal{I}) = \sum_{\mathbf{p} \in P_i} \rho(\mathbf{p}' - \mathbf{p} - \mathcal{F}_\mathcal{L}(\mathbf{p})), \quad (4)$$

where $\mathbf{p}'$ is the projected image coordinate on $\mathcal{L}^1$, defined in Eq. (3).

3.3. Optimization

3.3.1 Initialization

Since the energy model is highly non-convex with lots of local minimums, a good initialization is important for the algorithm to achieve good performance. In our experiments, we utilize RANSAC to simply solve the rigid fitting problem and obtain the 3D rigid motion for initialization.

3.3.2 Gaussian Newton(GN) Solver

Same as the original paper, we apply Gaussian Newton algorithm to solve the optimization problem, as shown in Eq. (1). For each iteration, the optimization problem of each instance i can be written as a weighted sum of squares:

$$\boldsymbol{\xi}^{(n+1)} = \arg\min_{\boldsymbol{\xi}} E_{\text{total},i}(\boldsymbol{\xi}) = \arg\min_{\boldsymbol{\xi}} \sum_{\text{Eng}} w_i(\boldsymbol{\xi}^{(n)}) r_i^2(\boldsymbol{\xi}^{(n)}),$$

where r is the residual function, w is weights and Eng refers to summing over the energy terms. Employing the Gaussian Newton algorithm, we have

$$\Delta\boldsymbol{\xi} = -(\sum_{\mathbf{p},\text{Eng}} \mathbf{J}_\mathbf{p}^\text{T}\mathbf{W}_\mathbf{p}\mathbf{J}_\mathbf{p})^{-1} \sum_{\mathbf{p},\text{Eng}} \mathbf{J}_\mathbf{p}^\text{T}\mathbf{W}_\mathbf{p} r(\mathbf{p}, \boldsymbol{\xi}^{(n)}; \mathcal{I}), \quad (5)$$

$$\boldsymbol{\xi}^{(n+1)} = \boldsymbol{\xi}^{(n)} \circ \Delta\boldsymbol{\xi}, \quad (6)$$

where $\circ$ is a pose composition operator and $\mathbf{J}_\mathbf{p} = \frac{\partial r(\mathbf{p},\boldsymbol{\xi};\mathcal{I})}{\partial \boldsymbol{\xi}}$ is the Jacobian matrix, $\mathbf{W}_\mathbf{p}$ is a diagonal matrix with diagonal elements equal to $\frac{\partial \tau(L_\mathbf{p})}{\partial L_\mathbf{p}}$. Define $\mathbf{x} = [x \quad y \quad z]^\text{T} = \boldsymbol{\xi} \circ \pi_{\mathbf{K}^{-1}}(\mathbf{p}, \mathcal{D}(\mathbf{p}))$ as the 3D coordinate of the pixel $\mathbf{p}$ after inverse depth warping and applying rigid transform $\boldsymbol{\xi}$, we can obtain the Jacobian matrices of three energy terms separately as follows:

$$\mathbf{J}_\text{photo} = \begin{bmatrix} (\nabla_x \mathcal{L}^1)^\text{T} \\ (\nabla_y \mathcal{L}^1)^\text{T} \end{bmatrix}^\text{T} \begin{bmatrix} \frac{f_x}{z} & 0 & -\frac{xf_x}{z^2} \\ 0 & \frac{f_y}{z} & -\frac{yf_y}{z^2} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & z & -y \\ 0 & 1 & 0 & -z & 0 & x \\ 0 & 0 & 1 & y & -x & 0 \end{bmatrix},$$

$$\mathbf{J}_\text{rigid} = \begin{bmatrix} 1 & 0 & 0 & 0 & z & -y \\ 0 & 1 & 0 & -z & 0 & x \\ 0 & 0 & 1 & y & -x & 0 \end{bmatrix},$$

$$\mathbf{J}_\text{flow} = \begin{bmatrix} \frac{f_x}{z} & 0 & -\frac{xf_x}{z^2} \\ 0 & \frac{f_y}{z} & -\frac{yf_y}{z^2} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & z & -y \\ 0 & 1 & 0 & -z & 0 & x \\ 0 & 0 & 1 & y & -x & 0 \end{bmatrix}.$$

The whole algorithm is shown in **Algorithm** 1.

Variables: $\boldsymbol{\xi}$, the 3D rigid motion
Inputs: $\mathcal{I} = \{\mathcal{L}^0, \mathcal{R}^0, \mathcal{L}^1, \mathcal{R}^1, \mathcal{D}^0, \mathcal{D}^1, \mathcal{F}_\mathcal{L}, \mathcal{F}_\mathcal{R}\}$, P_i
Result: $\boldsymbol{\xi}^*$, the optimal 3D rigid motion
Initialization: $\boldsymbol{\xi}^0 = \boldsymbol{\xi}_0$, $t = 0, T$;
while $t < T$ **do**
  $\mathbf{x} = \boldsymbol{\xi}^{(t)} \circ \pi_{\mathbf{K}^{-1}}(\mathbf{p}, \mathcal{D}(\mathbf{p}))$;
  Calculate $\mathbf{J}_\text{photo}, \mathbf{J}_\text{rigid}, \mathbf{J}_\text{flow}$;
  Calculate $\Delta\boldsymbol{\xi}$ according to Eq. 5;
  Update $\boldsymbol{\xi}$: $\boldsymbol{\xi}^{(t+1)} = \boldsymbol{\xi}^{(t)} \circ \Delta\boldsymbol{\xi}$;
end

Algorithm 1: Gaussian Newton Solver

3.4. Learning

We tune three neural networks differently. We fine-tune PSM-Net using KITTI 2015 dataset based on the one pretrained using Flyingthings3D. Due to lack of computation resources we are unable to train Mask R-CNN and PWC-Net for their large scale but instead using pretrained networks.

4. Experiments

4.1. Dataset and Implementation Details

Data: We use the training set of KITTI dataset for scene flow evaluation, which consists 200 sets of training images captured on real world driving scenarios and the corresponding flow and disparity information. We are unable to use test images of the KITTI dataset for evaluations, since we do not have corresponding ground truths.

Implementation details: In our implementation, the weights of each energy function are set to 1. We use all three energy functions for foreground objects, but only use photometric term for background. To obtain a good initialization for GN solver, we run RANSAC five times initially and only consider the rigid fitting term as the cost function. We iteratively update $\boldsymbol{\xi}$ through GN algorithm.

4.2. Qualitative Results

Fig. 3 illustrates the performance of the GN solver and compares results between the disparity and flow estimation before and after applying the solver. Some foreground objects are not very accurately estimated solely by the neural networks so errors around the vehicles can be observed. However, by projecting the 3D motion to disparity and flow the model is able to address such an issue to provide rigid estimation on the edges. Moreover, the solver is able to eliminate the occlusion error in estimation. This suggests that the inclusion of environmental information and prior knowledge can facility a better motion estimation.

4.3. Quantitative Results

4.3.1 Outliers Ratio

We calculate the outliers ratio of the refined optical flow and disparity as the performance measurement of the scene flow, which is the standard evaluation protocol used in KITTI and reference paper. Specifically, outliers ratio is often computed individually for background and foreground, which are obtained from a instance segmentation network. Therefore, the performance of a scene flow method is evaluated based on outliers ratios of background and of foreground for optical flow, disparity, and scene flow, respectively.

4.3.2 Comparison of Scene Flow Methods

A number of scene flow methods are compared with ours in terms of outliers ratio. As shown in Tab. 1, the model that we implement outperforms all others listed by a significant margin. Surprisingly, it achieves state-of-the-art performance on every entry, which, however, turns out to be more decent than its counterpart in the original paper. We believe that such spurious advantage is due to the fact that we have to use only training images to feed the model. The primary reason why we forgo using test images is that

	Disparity 1			Disparity 2			Optical Flow			Scene Flow		
Methods	bg	fg	all	bg	fg	all	bg	fg	all	bg	fg	all
CSF[19]	4.57	13.04	5.98	7.92	20.76	10.06	10.40	25.78	12.96	12.21	33.21	15.71
OSF[3]	4.54	12.03	5.79	5.45	19.41	7.77	5.62	18.92	7.83	7.01	26.34	10.23
SSF[21]	3.55	8.75	4.42	4.94	17.48	7.02	5.63	14.71	7.14	7.18	24.58	10.07
ISF[20]	4.12	6.17	4.46	4.88	11.34	5.95	5.40	10.29	6.22	6.58	15.63	8.08
Origin[23]	2.16	4.49	2.55	2.90	9.73	4.04	3.59	10.40	4.73	4.39	15.94	6.31
Ours	**1.02**	**0.46**	**0.95**	**0.76**	**1.39**	**0.94**	**2.81**	**7.41**	**3.49**	**3.29**	**8.12**	**4.06**

Table 1. Comparison against top 5 published approaches

Employed energy			Background outliers(%)				Foreground outliers(%)			
E_{pho}	E_{flow}	E_{rigid}	D1	D2	Fl	SF	D1	D2	Fl	SF
✓			1.08	1.68	9.96	10.64	0.48	1.52	14.83	15.21
	✓	✓	1.08	1.71	9.61	10.31	0.48	1.52	9.39	9.80
✓	✓	✓	1.08	1.71	9.60	10.30	0.48	1.53	11.06	11.47

Table 2. Contribution of each energy

Methods	D1-all	D2-all	Fl-all	SF-all
PSM+PWC	0.95	16.35	5.88	18.33
Deep+RANSAC	0.95	1.58	7.31	8,07
Our Full	**0.95**	**1.57**	**6.26**	**7.03**

Table 3. Improvement over original flow/stereo estimation

ground truths of the test set in KITTI scene flow dataset are inaccessible. It is true that a remedy would be to divide the original training set into a new training set and a validation set, and retrain each component network with the former, and then feed the latter to the model to output results. However, we have limited computational resources, and thus end up using only training images here.

4.3.3 Contribution of Energy Terms

In order to explore the effects of individual energy terms for the motion estimation performance, outlier ratio results of background and foreground objects are computed for three combinations of energy terms, as listed in Tab. 2.

From Tab. 2, one can see that the SF(scene flow) outliers ratios for background are significantly close among the three combinations, while a large margin can be seen among those for foreground; this is inconsistent with their counterpart results in the original paper. We believe this is because we only feed 30 sets of images to our model for the quantitative analysis here, due to limited time for this project. In the original paper, best performance is achieved for foreground objects when all energy terms are included, while the error for background is lowest when using only photometric term,

4.3.4 Contribution of structure optimization

To gain insights into potential improvements of model, certain estimation results are compared in Tab. 3: direct output from PSM and PWC, the result obtained by RANSAC and the one after GN solver. Tab. 3 demonstrates that our structure optimization algorithm could significantly improve the performance and eliminate the errors in deep neural network. This implies that it is very crucial to incorporate prior knowledge in this task.

5. Conclusion

We have successfully re-implemented the whole model proposed in the reference paper and conducted several experiments to evaluate the performance of our implementation. The results show that we achieve good performance on the estimation of the 3D rigid motion. And in some experiments, we even outperform the reference paper since we can only test on the training data. However, there are also some unreasonable results. For the future improvement, one important thing is to do more data pre-processing. In the reference paper, they spent lots of efforts pruning out the outliers, which will significantly effect the fitting problem. Furthermore, our problem has a highly non-convex structure, so it is more sensitive to the outliers and the initialization.

References

[1] Sundar Vedula, Simon Baker, Peter Rander, Robert Collins, and Takeo Kanade. Three-dimensional scene flow. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 722–729. IEEE, 1999.

[2] Christoph Vogel, Konrad Schindler, and Stefan Roth. Piecewise rigid scene flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1377–1384, 2013.

[3] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3061–3070, 2015.

[4] Michael J Black and Paul Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer vision and image understanding*, 63(1):75–104, 1996.

[5] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016.

[6] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.

[7] William Hoff and Narendra Ahuja. Surfaces from stereo: Integrating feature matching, disparity estimation, and contour detection. *IEEE transactions on pattern analysis and machine intelligence*, 11(2):121–136, 1989.

[8] Takeo Kanade and Masatoshi Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE transactions on pattern analysis and machine intelligence*, 16(9):920–932, 1994.

[9] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 66–75, 2017.

[10] 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.

[11] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.

[12] Berthold KP Horn and Brian G Schunck. Determining optical flow. In *Techniques and Applications of Image Understanding*, volume 281, pages 319–331. International Society for Optics and Photonics, 1981.

[13] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1164–1172, 2015.

[14] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017.

[15] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4161–4170, 2017.

[16] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018.

[17] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8981–8989, 2018.

[18] Koichiro Yamaguchi, David McAllester, and Raquel Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *European Conference on Computer Vision*, pages 756–771. Springer, 2014.

[19] Zhaoyang Lv, Chris Beall, Pablo F Alcantarilla, Fuxin Li, Zsolt Kira, and Frank Dellaert. A continuous optimization approach for efficient and accurate scene flow. In *European Conference on Computer Vision*, pages 757–773. Springer, 2016.

[20] Aseem Behl, Omid Hosseini Jafari, Siva Karthik Mustikovela, Hassan Abu Alhaija, Carsten Rother, and Andreas Geiger. Bounding boxes, segmentations and object coordinates: How important is recognition for 3d scene flow estimation in autonomous driving scenarios? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2574–2583, 2017.

[21] Zhile Ren, Deqing Sun, Jan Kautz, and Erik Sudderth. Cascaded scene flow prediction using semantic segmentation. In *2017 International Conference on 3D Vision (3DV)*, pages 225–233. IEEE, 2017.

[22] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[23] Wei-Chiu Ma, Shenlong Wang, Rui Hu, Yuwen Xiong, and Raquel Urtasun. Deep rigid instance scene flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3614–3622, 2019.