Deep Equilibrium Model for Memory Efficient Stereo Matching on the High Resolution Argoverse Dataset

Antyanta Bangunharcana Soohyun Kim Kyung-Soo Kim
Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea
{antabangun, soohyun, kyungsookim}@kaist.ac.kr

1. Introduction

The estimation of depth from stereo image pairs is a longstanding computer vision task with applications in robotics [12, 13]. In this report, we are interested in building a stereo matching solution for a high resolution images in the Argoverse dataset [8] that runs within 200*ms* in modern GPUs. To allow for flexible design of the model in satisfying the time requirements, we adopt a RAFT based model in our solution [16, 18] which has gained much attention [6, 14] in recent years, wherein an update operation is iteratively performed to refine the disparity predictions as well as the hidden states. Through the use of such design, the number of unrolled iterations can be adjusted depending on the desired model latency.

However, this class of models also introduces huge memory consumption necessary for backpropagation through time (BPTT) during training. To address this issue, We introduce the following contributions. First, we adopt deep equilibrium (DEQ) formulation [2, 3] into our RAFT-based stereo model. Additionally, to improve the representation power, we follow the canonical volumetric based deep stereo models [7, 15] and use 3D convolutions to extract geometric features from the cost volume. This 3D features are utilized during the iterative updates which supplies the network with geometric knowledge. We also utilize the 3D features to regress an initial disparity estimate, allowing for better convergence of the model [4].

2. Method

The overall model contains two principal components, the volumetric submodule and the iterative updates to refine the disparity predictions. We illustrate the model in Figure 1.

2.1. Geometric stereo

We begin by extracting feature maps for both the input images $I \in \mathbb{R}^{H \times W \times 3}$. We follow RAFT and pass the images into encoder sub-networks, with the objective of extracting matching and context feature maps. Both the left and right input images are passed into the matching sub-network to give the matching feature maps $I_L^m, I_R^m \in \mathbb{R}^{h \times w \times c}$. The context sub-network is only applied to the reference image, which in this case is the left image, and it gives us the initial hidden states $h^{[0](s)}$ and the context features $q^{(s)}$ at multiple scales s.

We build a cost volume $\mathbf{C} \in \mathbb{R}^{h \times w \times D \times c_o}$, following previous stereo matching works by computing pairwise correlations [17] using the extracted matching feature maps, where D is the maximum disparity. A 3D UNet is then applied on the correlation volume giving us 3D geometric features $\mathbf{C}^{(\mathbf{s})} \in \mathbb{R}^{h/2^s \times w/2^s \times D/2^s \times c_s}$ at multiple scales. At the highest scale $s = 0$ of the aggregated volume, we compute an initial disparity estimate as a weighted sum of the candidate disparities [15]

$$\hat{d_{init}} = \sum_{d=0}^{D} d \times Softmax(q_d). \tag{1}$$

Alternatively, we use the weighted sum of only the top matching candidates [5].

2.2. Deep Equilibrium Stereo

Our iterative updates to refine $\hat{d^{[0]}}$ and the hidden states $h^{[0](s)}$ follows the update operations in RAFT-Stereo. A GRU-based [9] convolutional layers are used to update the hidden states using the image contexts $q^{(s)}$ following the Slow-fast GRU updates. At the highest scale, we additionally sample values from the cost volume that is used as additional input into the update layers. However, instead of sampling just the correlation values from the volumes, we trilinearly sample the 3D features at the disparity of interests, giving our model more representational power that is aware of the 3D contexts.

These update steps, when implemented naively, may consume a massive amount of memory during training, especially with the addition of the 3D contexts on a high resolution input images. Therefore, we adopt a deep equilibrium formulation into our model. Specifically, we solve for the fixed point $h^{*(s)}$ and $\hat{d}^*$ and turn off the autograd functionality of Pytorch in all the previous update steps. We also use Anderson solver [1] to accelerate convergence to the fixed

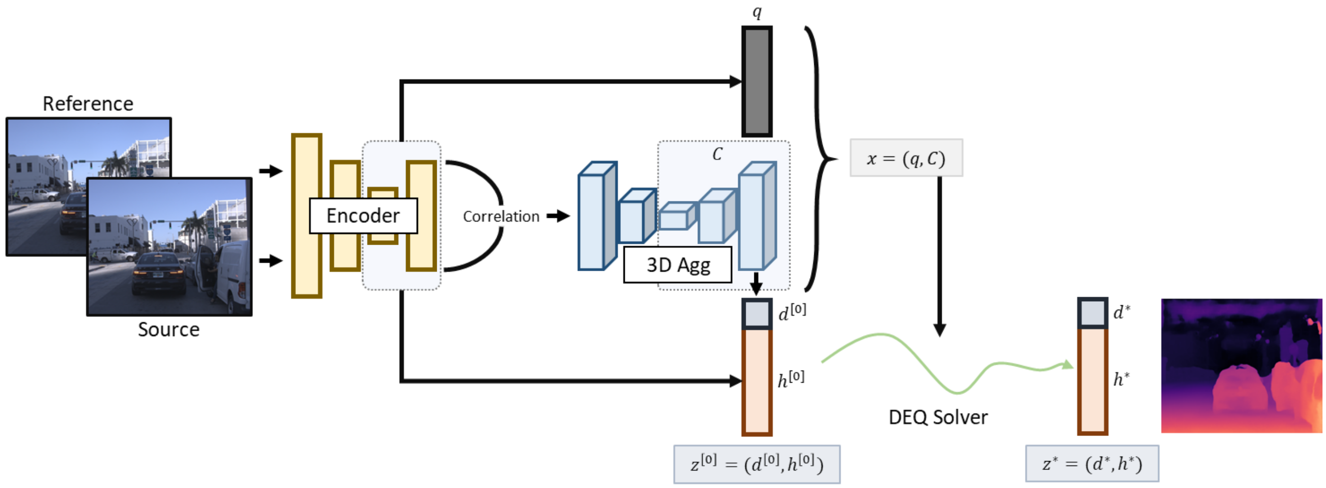

Figure 1. Overall architecture of the model.

point. As the autograd is only turned on at the fixed point, the memory consumption for the iterative updates is reduced to almost free.

The other concern that we need to address is the real-time requirement of the model, and a quick convergence to the fixed point is desirable. The initial disparity predicted by the 3D volumetric sub-network helps with this objective, but we also want to reduce the inference time for the iterative updates. To achieve this, we train a hyper-anderson solver [4] to substitute the traditional anderson solver.

3. Experiments

We first train our model on large synthetic datasets before finetuning it to the Argoverse dataset. We use the SceneFlow dataset [17], and also capture a dataset from CARLA driving simulator [10, 11], providing us with a large data in urban driving scenario with sensor configuration which resemble that of Argoverse. We train our model on the synthetic data randomly cropped with size 512×960 and a batch size of 4 for 50 epochs. We set the maximum disparity of interest to be 384.

To also improve the generalization ability towards the real world data, we perform color augmentation following [19] by adding random gaussian noise with standard deviation 0.02 and gaussian blur with standard deviation range sampled randomly from $[0, 1]$ to the right image. We also apply color jitters to the right image with brightness, contrast, saturation, and hue ranges of $0.2, 0.2, 0.2$ and 0.01. Additionally, we perform random zooming and stretching of the image to allow generalization to varying disparity distribution.

We then finetune the model on the Argoverse dataset that is randomly cropped into 1920×1920 with a batch size of 2. Training on such a large resolution is made possible due to the DEQ implementation. A hyper anderson solver is then trained with this model as a fixed point target.

Our model is trained on a single Nvidia RTX 3090 GPU. Based on the $200ms$ and the latency comparison between RTX 3090 and V100, we require our model to run within $200/1.29 = 155ms$. In our experiments, we found that we can allow for $4 \sim 5$ update iterations using the hyper anderson solver. In our submitted result, we only used 4 iterations to ensure our model satisfies the real-time requirements. At the time of submission, this model ranked 2nd on the Argoverse Stereo Competition 2022 Leaderboard.

4. Conclusion

We presented a design of stereo matching network that is flexible to the requirements. The model is inspired by the canonical volumetric design of stereo matching models and the iterative refinements design that is recently gaining attention. By combining both concepts, our model benefits from the geometric knowledge obtained from the volumetric design and the flexibility of iterative refinements. However, more thorough experiments still needs to be done to explore the potential of the proposed model.

The codes will be made available on `https://github.com/antabangun/ges`

References

[1] Donald G Anderson. Iterative procedures for nonlinear integral equations. *Journal of the ACM (JACM)*, 12(4):547–560, 1965.

[2] Shaojie Bai, Zhengyang Geng, Yash Savani, and J Zico Kolter. Deep equilibrium optical flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 620–630, 2022.

[3] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. *Advances in Neural Information Processing Systems*, 32, 2019.

[4] Shaojie Bai, Vladlen Koltun, and J Zico Kolter. Neural deep equilibrium solvers. In *International Conference on Learning Representations*, 2021.

[5] Antyanta Bangunharcana, Jae Won Cho, Seokju Lee, In So Kweon, Kyung-Soo Kim, and Soohyun Kim. Correlate-and-excite: Real-time stereo matching via guided cost volume excitation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3542–3548. IEEE, 2021.

[6] Antyanta Bangunharcana, Soohyun Kim, and Kyung-Soo Kim. Revisiting the receptive field of conv-gru in droid-slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1906–1916, 2022.

[7] 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.

[8] Ming-Fang Chang, John W Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3d tracking and forecasting with rich maps. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[9] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[10] Jean-Emmanuel Deschaud. Kitti-carla: a kitti-like dataset generated by carla simulator. *arXiv preprint arXiv:2109.00892*, 2021.

[11] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[13] Sungchul Hong, Antyanta Bangunharcana, Jae-Min Park, Minseong Choi, and Hyu-Soung Shin. Visual slam-based robotic mapping method for planetary construction. *Sensors*, 21(22):7715, 2021.

[14] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9772–9781, 2021.

[15] 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.

[16] Lahav Lipson, Zachary Teed, and Jia Deng. Raft-stereo: Multilevel recurrent field transforms for stereo matching. In *2021 International Conference on 3D Vision (3DV)*, pages 218–227. IEEE, 2021.

[17] 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.

[18] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020.

[19] Jamie Watson, Oisin Mac Aodha, Daniyar Turmukhambetov, Gabriel J Brostow, and Michael Firman. Learning stereo from single images. In *European Conference on Computer Vision*, pages 722–740. Springer, 2020.