RAFT-3D: Supplementary Material

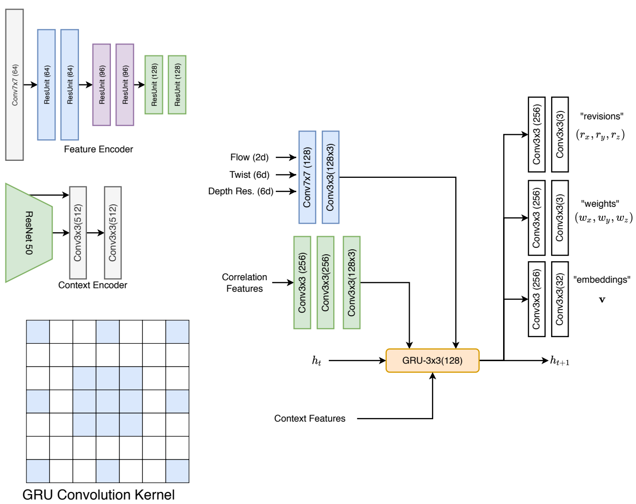

Figure 1. Network architecture. The components include (1) a feature encoder (2) a context encoder with a ResNet50 backbone, and (3) and GRU-based updated operator. The GRU uses a dilated convolution pattern as shown. In contrast to RAFT[3] where features are concatenated before being passed to the GRU, we perform elementwise addition of the context features, correlation features, and motion features.

1. Network Architecture

Details of the network architecture, including feature encoders and the GRU-based update operator are shown in Figure 1.

2. bi-Laplacian Optimization Layer Gradients

This layer minimizes an objective function in the form

$$||D_x \mathbf{u}||^2_{\mathbf{w}_x} + ||D_x \mathbf{u}||^2_{\mathbf{w}_y} + ||\mathbf{u} - \mathbf{v}||^2 \tag{1}$$

where D_x and D_y are linear finite difference operators, and $\mathbf{v}$ is the flattened feature map.

First consider the case of single channel, $\mathbf{v} \in \mathbb{R}^{HW}$. Let $W_x = \text{diag}(\mathbf{w}_x), W_y = \text{diag}(\mathbf{w}_y) \in \mathbb{R}^{HW \times HW}$. We can solve for $\mathbf{u}^*$

$$(\mathbf{I} + D_x^T W_x D_x^T + D_y^T W_y D_y^T)\mathbf{u}^* = \mathbf{v} \tag{2}$$

We perform sparse Cholesky factorization and backsubstition to solve for $\mathbf{u}^*$ using the Cholmod library[2].

Gradients: In the backward pass, given the gradient $\frac{\partial L}{\partial \mathbf{u}^*}$, we need to find the gradients with respect to the boundary weights $\frac{\partial L}{\partial \mathbf{w}_x}$ and $\frac{\partial L}{\partial \mathbf{w}_y}$.

Given the linear system $\mathbf{H}\mathbf{u} = \mathbf{v}$, the gradients with respect to $\mathbf{H}$ and $\mathbf{v}$ can be found by solving the system in the backward direction [1]

$$\frac{\partial L}{\partial \mathbf{v}} = \mathbf{H}^{-T} \frac{\partial L}{\partial \mathbf{u}^*} \tag{3}$$

$$\frac{\partial L}{\partial \mathbf{H}} = \mathbf{u}^* \mathbf{d}_v^T \tag{4}$$

$$\mathbf{d}_v = \mathbf{H}^{-T} \frac{\partial L}{\partial \mathbf{u}^*} \tag{5}$$

Here the column vector $\mathbf{d}_v$ is defined for notational convenience. Since $\mathbf{H}$ is positive definite, $\mathbf{H}^{-T} = \mathbf{H}^{-1}$ so we can reuse the factorization from the forward pass.

To compute the gradients with respect to $\mathbf{w}_x$ and $\mathbf{w}_x$

$$\frac{\partial L}{\partial \mathbf{w}_x} = \text{diag}\left(\frac{\partial L}{\partial \mathbf{H}} \frac{\partial \mathbf{H}}{\partial W_x}\right) \tag{6}$$

$$= \text{diag}\left((D_x \mathbf{u}^*)(D_x \mathbf{d}_v)^T\right) \tag{7}$$

giving

$$\frac{\partial L}{\partial \mathbf{w}_x} = (D_x \mathbf{u}^*) \odot (D_x \mathbf{d}_v) \tag{8}$$

where $\odot$ is elementwise multiplication. Similarly

$$\frac{\partial L}{\partial \mathbf{w}_y} = (D_y \mathbf{u}^*) \odot (D_y \mathbf{d}_v) \tag{9}$$

Multiple Channels: We can easily extend Eqn. 2 to work with multiple channels. Since the matrix $\mathbf{H}$ does not depend on $\mathbf{v}$, it only needs to be factored once. We can solve Eqn. 2 for all channels by reusing the factorization, treating $\mathbf{v}$ as a $HW \times C$ matrix. The gradient formulas can also be updated by summing the gradients over the channel dimensions.

References

[1] Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 136–145. JMLR. org, 2017. 1

[2] Yanqing Chen, Timothy A Davis, William W Hager, and Sivasankaran Rajamanickam. Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. *ACM Transactions on Mathematical Software (TOMS)*, 35(3):1–14, 2008. 1

[3] 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. 1