LSM: Learning Subspace Minimization for Low-level Vision – Supplementary Material –

A. Derivatives of Various Data Terms $D(\boldsymbol{x})$

In Sec 3.4, we introduced two categories of tasks. Now, we show the first-order and the (approximated) second-order derivatives of the data terms, which compose the vector $\boldsymbol{d}$ and the (block) diagonal matrix $\boldsymbol{D}$ at each iteration.

Binary Image Labeling Recall that the first category is binary image labeling (interactive segmentation and video segmentation) as:

$$D(\boldsymbol{x}) = \sum_{\boldsymbol{p}} \alpha_{\boldsymbol{p}} \|\tau(\boldsymbol{x}_{\boldsymbol{p}}) - 1\|_2^2 + \beta_{\boldsymbol{p}} \|\tau(\boldsymbol{x}_{\boldsymbol{p}}) + 1\|_2^2, \quad \text{(A.1)}$$

where $\boldsymbol{p} = [x, y]^\top$ is a pixel coordinate, τ is an activation function to relax the binary label $\tau(\boldsymbol{x}_{\boldsymbol{p}})$ between $(+1, -1)$, and $\alpha_{\boldsymbol{p}}$ and $\beta_{\boldsymbol{p}}$ are the probabilities that $\tau(\boldsymbol{x}_{\boldsymbol{p}}) = +1$ or -1. Therefore, the first-order and the second-order derivatives at an intermediate solution $\boldsymbol{x}$ are:

$$\begin{aligned} \frac{\partial D}{\partial \boldsymbol{x}_{\boldsymbol{p}}} &= [(\alpha_{\boldsymbol{p}} + \beta_{\boldsymbol{p}})\tau(\boldsymbol{x}_{\boldsymbol{p}}) + (\beta_{\boldsymbol{p}} - \alpha_{\boldsymbol{p}})][\frac{\partial \tau(\boldsymbol{x}_{\boldsymbol{p}})}{\partial \boldsymbol{x}_{\boldsymbol{p}}}], \\ \frac{\partial^2 D}{\partial \boldsymbol{x}_{\boldsymbol{p}}^2} &= (\alpha_{\boldsymbol{p}} + \beta_{\boldsymbol{p}})[\frac{\partial \tau(\boldsymbol{x}_{\boldsymbol{p}})}{\partial \boldsymbol{x}_{\boldsymbol{p}}}]^2, \end{aligned} \quad \text{(A.2)}$$

where we ignore the scale factor 2 for simplicity, and $\frac{\partial \tau(\boldsymbol{x}_{\boldsymbol{p}})}{\partial \boldsymbol{x}_{\boldsymbol{p}}}$ can be $1 - \tau^2(\boldsymbol{x}_{\boldsymbol{p}})$ for $tanh$ activation function.

Dense Correspondence Estimation The second category is the dense correspondence estimation (stereo matching and optical flow) where the data term is:

$$D(\boldsymbol{x}) = \sum_{\boldsymbol{p}} \|F_S(\boldsymbol{p} + \boldsymbol{x}_{\boldsymbol{p}}) - F_T(\boldsymbol{p})\|_2^2. \quad \text{(A.3)}$$

For stereo matching, the derivatives are derived as:

$$\begin{aligned} \frac{\partial D}{\partial \boldsymbol{x}_{\boldsymbol{p}}} &= \nabla_x F_S(\boldsymbol{p} + \boldsymbol{x}_{\boldsymbol{p}})^\top [F_S(\boldsymbol{p} + \boldsymbol{x}_{\boldsymbol{p}}) - F_T(\boldsymbol{p})], \\ \frac{\partial^2 D}{\partial \boldsymbol{x}_{\boldsymbol{p}}^2} &= \|\nabla_x F_S(\boldsymbol{p} + \boldsymbol{x}_{\boldsymbol{p}})\|_2^2, \end{aligned} \quad \text{(A.4)}$$

where ∇_x is the gradient operator along the horizontal direction. $\nabla_x F_S(\boldsymbol{p} + \boldsymbol{x}_{\boldsymbol{p}})$ and $[F_S(\boldsymbol{p} + \boldsymbol{x}_{\boldsymbol{p}}) - F_T(\boldsymbol{p})]$ are vectors, so $\frac{\partial D}{\partial \boldsymbol{x}_{\boldsymbol{p}}}$ and $\frac{\partial^2 D}{\partial \boldsymbol{x}_{\boldsymbol{p}}^2}$ are scalars, which is also an one-dimensional problem and can be unified with the binary image label tasks with the same network and the parameters.

For optical flow, $\boldsymbol{x}_{\boldsymbol{p}} = [u, v]^\top$ is a 2D vector and the derivatives are:

$$\begin{aligned} \frac{\partial D}{\partial \boldsymbol{x}_{\boldsymbol{p}}} &= \nabla F_S(\boldsymbol{p} + \boldsymbol{x}_{\boldsymbol{p}})^\top [F_S(\boldsymbol{p} + \boldsymbol{x}_{\boldsymbol{p}}) - F_T(\boldsymbol{p})], \\ \frac{\partial^2 D}{\partial \boldsymbol{x}_{\boldsymbol{p}}^2} &= \nabla F_S(\boldsymbol{p} + \boldsymbol{x}_{\boldsymbol{p}})^\top \nabla F_S(\boldsymbol{p} + \boldsymbol{x}_{\boldsymbol{p}}), \end{aligned} \quad \text{(A.5)}$$

where ∇ is the gradient operator along both the horizontal and vertical direction. Therefore, $\frac{\partial D}{\partial \boldsymbol{x}_{\boldsymbol{p}}}$ is a 2×1 vector, and $\frac{\partial^2 D}{\partial \boldsymbol{x}_{\boldsymbol{p}}^2}$ is a 2×2 matrix, which makes unification with other one-dimensional tasks difficult. To address this problem, we apply Cramer's rule [4] as follows:

- First, we compute the determinant of $\frac{\partial^2 D}{\partial \boldsymbol{x}_{\boldsymbol{p}}^2}$ as $det_{\boldsymbol{p}}$.
- Next, we replace the first column of $\frac{\partial^2 D}{\partial \boldsymbol{x}_{\boldsymbol{p}}^2}$ with $\frac{\partial D}{\partial \boldsymbol{x}_{\boldsymbol{p}}}$, and denote the determinant of the modified matrix as $det_{\boldsymbol{p}}^x$. Similarly, $det_{\boldsymbol{p}}^y$ is computed by replacing the second column of $\frac{\partial^2 D}{\partial \boldsymbol{x}_{\boldsymbol{p}}^2}$ with $\frac{\partial D}{\partial \boldsymbol{x}_{\boldsymbol{p}}}$.
- Finally, we collect $det_{\boldsymbol{p}}^x$ and $det_{\boldsymbol{p}}$ at all pixel locations as the minimization context, concatenate it with the image context to generate the subspace $\mathcal{V}_x$ for the horizontal component of the flow field. Similarly, the $det_{\boldsymbol{p}}^y$ and the $det_{\boldsymbol{p}}$ are collected as the minimization context for the vertical subspace $\mathcal{V}_y$. Thus the subspace generation for optical flow is unified with other one-dimensional tasks by generating the subspace for the horizontal and the vertical components of flow individually.

B. Model Efficiency

Our LSM model is efficient in terms of model size, training time, and inference time, which are contributed by integrating data terms explicitly.

B.1. Model Size

We implement our LSM framework with the aforementioned settings, which contains about 15M parameters and costs 57.26 MB in memory. As shown in Fig. A.1, our LSM model maintains a relatively small model size when compared with other CNN based methods. But our LSM model

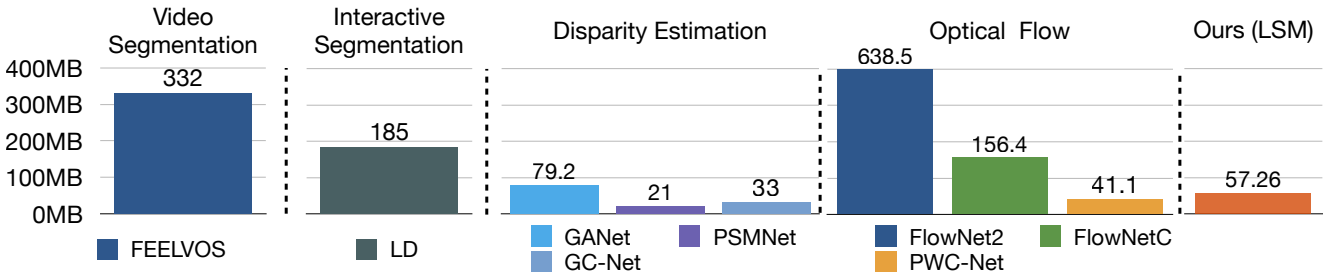

Figure A.1: Our LSM model handles multiple tasks in a relatively small model.

handles multiple tasks within the same parameters while others are designed specifically for single tasks.

B.2. Training Efficiency

We train our model with 143.2K iterations for all the experiments, which tasks roughly 20 hours and is relatively faster compared to existing CNN based methods. For example, training FlowNet2 [5] tasks more than 14 days and PWC-Net [6] takes 4.8 days. We initialize the backbone DRN-22 from the ImageNet pre-trained model, which also helps the training converges faster [3].

B.3. Inference Efficiency

Our LSM framework is also efficient during inference. Since we unify different tasks into a single network, the inference times for various tasks are roughly the same, which consume about 25ms for 512×384 images. The computation is dominated by the feature pyramid construction, the subspace generation and the minimization.

C. Zero-shot Interactive Segmentation

Similar to the other zero-shot generalization tests in Sec. 4.3, we also leave the interactive segmentation out for testing and train on the other tasks. When interact only once, the average IoU is 0.802 for our LSM model learned on the other tasks and tested on the interactive segmentation. Which is still superior than the conventional method [2, 1] as shown in Fig. A.2.

Figure A.2: Our zero-shot generalized LSM model performs better than GSC [2] for interactive segmentation.

References

[1] L. Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(11):1768–1783, 2006. 2

[2] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 2

[3] Kaiming He, Ross Girshick, and Piotr Dollar. Rethinking imagenet pre-training. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2

[4] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, second edition, 2002. 1

[5] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[6] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2