Supplementary

1. Implementation Details

We provide addition implementation details in this part, including auxiliary flow regression, affine estimation and general training settings.

1.1. Intermediate Flow Regression

We inherit formulation of auxiliary flow regression from ASpanFormer [1], where the flows are treated as random variables in 2D gaussian distribution. Concretely, for a pixel in source image, the probability that it correspond to (x, y) in target image is given by

$$P(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp(-\frac{(x-u_x)^2}{2{\sigma_x}^2} - \frac{(y-u_y)^2}{2{\sigma_y}^2}) \quad (1)$$

The mean u_x, u_y and standard deviation σ_x, σ_y is predicted by a network introduced in main paper Sec. 3.2.1. During training, we use log liklihood loss to supervise flow prediction. More specifically, for each pixel the flow loss is given by

$$L_{flow} = log[\frac{1}{2\pi\sigma_x\sigma_y}\exp(-\frac{(x_{gt}-u_x)^2}{2\sigma_x^2} - \frac{(y_{gt}-u_y)^2}{2\sigma_y^2})] \quad (2)$$

$$= log2\pi + log\sigma_x + log\sigma_y + \frac{(x_{gt}-u_x)^2}{2\sigma_x^2} + \frac{(y_{gt}-u_y)^2}{2\sigma_y^2} \quad (3)$$

where $[x_{gt}, y_{gt}]$ is ground truth correspondences coordinates. After Denoting $w_x = log\sigma_x, w_y = log\sigma_y$ and dropping $log2\pi$, then

$$L_{flow} = w_x + w_y + \frac{1}{2}e^{-2w_x}(x_{gt}-u_x)^2 + \frac{1}{2}e^{-2w_y}(y_{gt}-u_y)^2 \quad (4)$$

First two terms encourage the network to decrease prediction uncertainty. Last two terms use inverse exponential to weight l2-distance between ground truth and predicted coordinates, which encourages prediction with lower accuracy to have a higher standard deviation (or lower confidence). As a whole, the flow loss encourages lower flow estimation error and adjust uncertainty accordingly.

1.2. Regularizing Affine Estimation

As mentioned in main paper, the accuracy of estimated affine matrices can be affected by noisy intermediate flow. To alleviate this issue, we further apply regularization upon each affine matrix.

Concretely, we pick top 50% points w.r.t. flow uncertainty within each local patch to estimate affine. The estimated A will be decomposed into scale, rotation and shearing factor as following,

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} 1 & m \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \quad (5)$$

which yields,

$$\theta = \textbf{arctan}(\frac{a_{21}}{a_{11}}) \quad (6)$$

$$s_x = \sqrt{(a_{11}^2 + a_{21}^2)} \quad (7)$$

$$s_y = |a_{22}cos\theta - a_{12}sin\theta| \quad (8)$$

$$m = \frac{a_{12}cos\theta + a_{22}sin\theta}{s_y} \quad (9)$$

We constrain the scale s_x, s_y to be in range of [0.5, 4], the shearing factor m to be in range [-0.5, 0.5] and the rotation angle θ to be in range $[-\frac{\pi}{3}, \frac{\pi}{3}]$. We further deprecate local message from border grids (setting the corresponding selective fusion score as 0). In Tab. 1, we provide ablation on affine estimation w. and w/o regularization.

Design	Pose Estimation AUC		
	@5	@10	@20
w. affine regularization	27.1	47.5	64.8
w/o affine regularization	26.5	46.9	63.9

Table 1. Effect of affine regularization on ScanNet dataset [2].

1.3. Training Settings

We inherit data splits from LoFTR [6] and its following works [1, 7, 9] to train our model on both ScanNet [3] and MegaDepth [5]. ScanNet model is trained with batch size 32 and initial learning rate $3e^{-3}$, where a linear warm-up us applied for the first epoch. Learning rate is halved at epoch [3, 6, 9, 12, 15, 18, 21, 24, 27]. Megadepth model is trained with batch size 8 and initial learning rate $1e^{-3}$, where a linear warm-up us applied for the first 3 epoches. Learning rate is halved at epoch [8,12,16,20,24].

1.4. Light-weight variant

We reduce our full model on 3 points to obtain the light-weight version: (1) channel number for all attention layers is cut from 256 to 128, (2) number of interleaved attention blocks is cut from 4 to 2, (3) CNN backbone is reduced from ResNet-18 to a light vgg style network, of which the structure can be seen in Fig 1.

2. More Visualizations

We provide additional visualizations on matches and affine attention span in Fig. 2

3. Limitations and Future Works

For a fair comparison with previous works, our network is based on a relatively old fashion, including ResNet backbone and absolute sinusoidal positional encoding, which can be replaced by advanced feature extractor [4, 8] and rotary positional encoding [8]. Furthermore, although our method is effective in general scenarios, we find the affine-based deformation may not hold in some special settings, such as non-rigid scenes or objects with very fine-grained geometry. Potential future works include modernize our network with recent progress in backbone/Transformer designs and further improve the robustness of local deformation modeling.

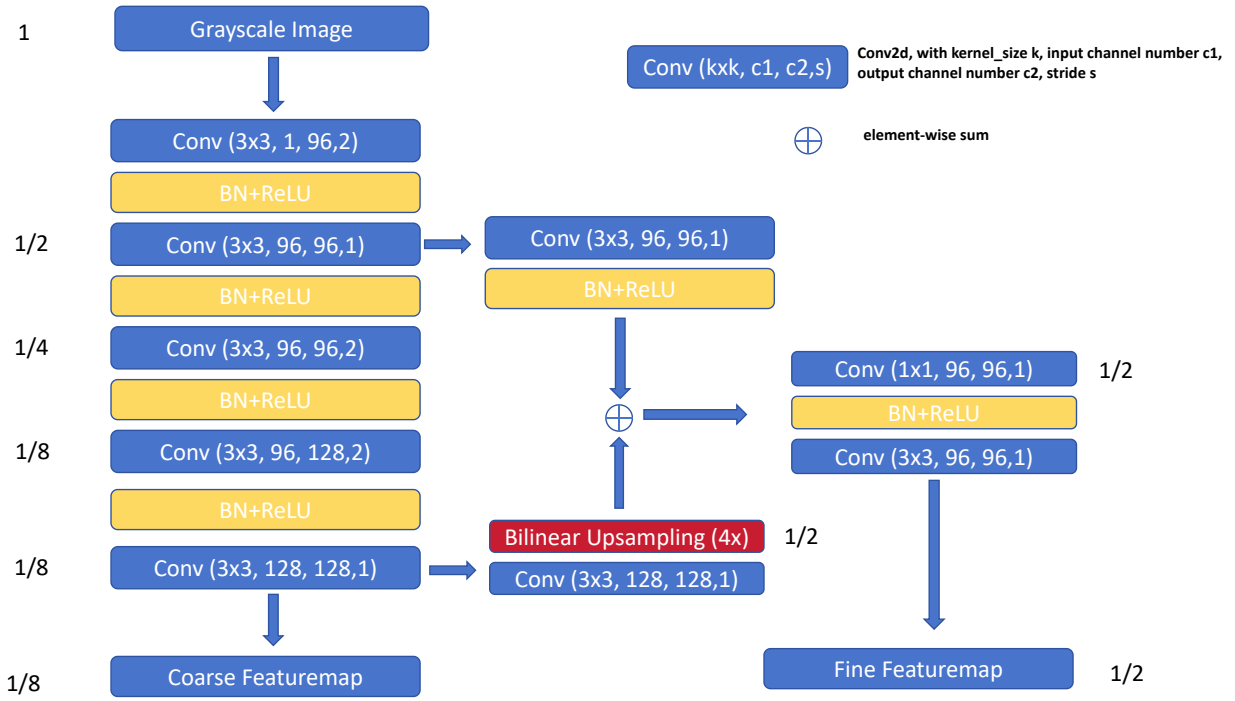

Figure 1. Network structure for backbone of light-weight variant. The backbone takes a full scale grayscale image as input and outputs coarse/fine feature map in $\frac{1}{8}, \frac{1}{2}$ respectively. The coarse feature is fed into our attention-based cross-view network for feature update, while the fine level network is used for match coordinate refinement.

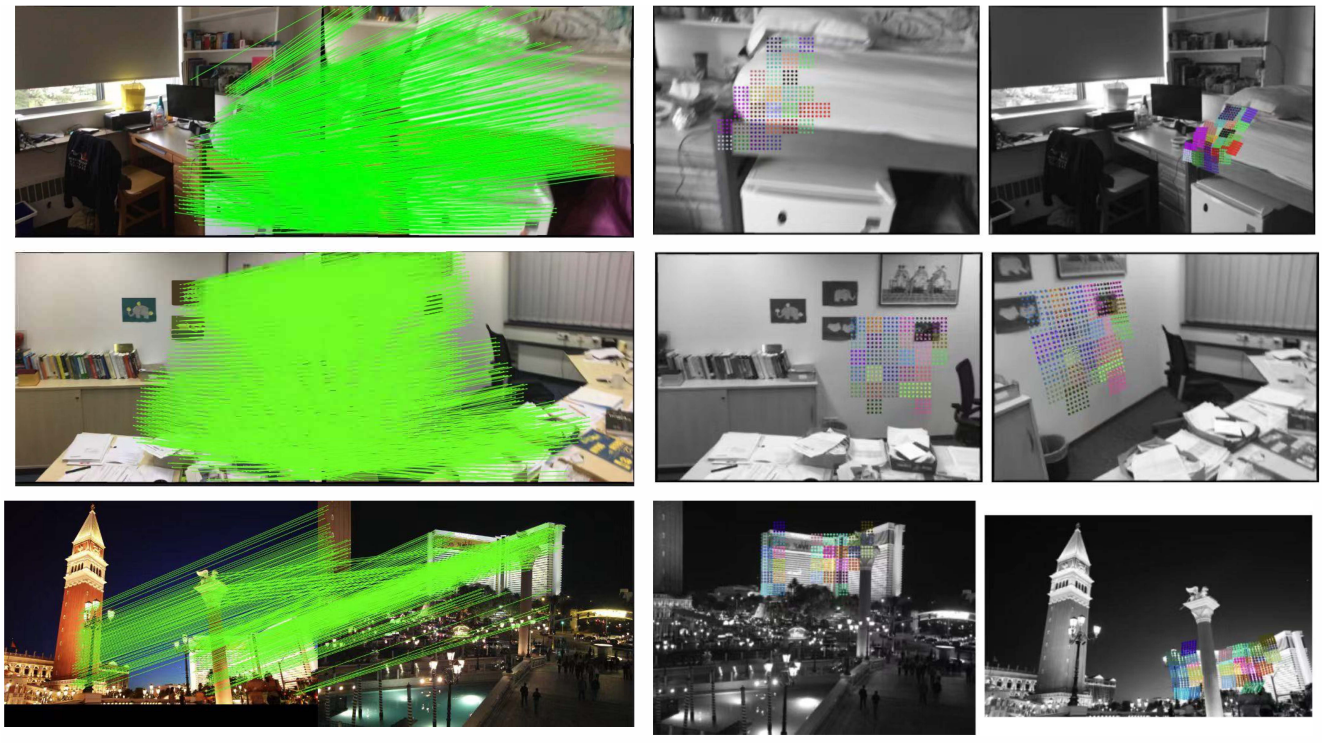

Figure 2. Matching results and estimated affine filed (top 20 according to uncertainty).

References

[1] Hongkai Chen, Zixin Luo, Lei Zhou, Yurun Tian, Mingmin Zhen, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. Aspanformer: Detector-free image matching with adaptive span transformer. In *ECCV*, 2022. 1

[2] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 1

[3] Angela Dai, Matthias Nießner, Michael Zollöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics 2017 (TOG)*, 2017. 1

[4] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Revisiting robust losses for dense feature matching. *arXiv preprint arXiv:2305.15404*, 2023. 2

[5] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 1

[6] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, 2021. 1

[7] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. In *ICLR*, 2021. 1

[8] Yifan Wang, Xingyi He, Sida Peng, Dongli Tan, and Xiaowei Zhou. Efficient LoFTR: Semi-dense local feature matching with sparse-like speed. In *CVPR*, 2024. 2

[9] Jiahuan Yu, Jiahao Chang, Jianfeng He, Tianzhu Zhang, Jiyang Yu, and Wu Feng. ASTR: Adaptive spot-guided transformer for consistent local feature matching. *CVPR*, 2023. 1