Supplementary Material: Enhancing Human Optical Flow via 3D Spectral Prior

Shiwei Mao, Mingze Sun and Ruqi Huang†

Tsinghua Shenzhen International Graduate School, China

In this supplementary material, we first provide more technical details on our rendering component and a brief overview of functional maps framework in Sec.1. Then we introduce implementation details in Sec.2, followed by more comprehensive qualitative results in Sec.3.

1. Technical Details

Rendering Details: During mesh rendering, camera rays will inevitably hit the face of a triangle mesh, instead of the vertices. Thus, we use barycentric coordinates to calculate the eigenbasis. In general, a point with barycentric coordinates (u,v,w) is inside (or on) the triangle if and only if $0 \le u,v,w \le 1$, or alternatively if and only if $0 \le v \le 1, 0 \le w \le 1$, and $v+w \le 1$. With v and w arbitrary, we can reformulate $P = uA+vB+wC, u+v+w = 1$ as:

$$\begin{aligned} P &= A + v(B-A) + w(C-A) \\ &= (1-v-w)A + vB + wC. \end{aligned} \tag{1}$$

To solve for the barycentric coordinates, Eqn.1 can be written as $v\mathbf{v}_0 + w\mathbf{v}_1 = \mathbf{v}_2$, where $\mathbf{v}_0 = B-A$, $\mathbf{v}_1 = C-A$, and $\mathbf{v}_2 = P-A$. Now, a 2×2 system of linear equations can be formed by taking the dot product of both sides with both $\mathbf{v}_0$ and $\mathbf{v}_1$:

$$\begin{aligned} v(\mathbf{v}_0\cdot\mathbf{v}_0) + w(\mathbf{v}_1\cdot\mathbf{v}_0) &= \mathbf{v}_2\cdot\mathbf{v}_0, \\ v(\mathbf{v}_0\cdot\mathbf{v}_1) + w(\mathbf{v}_1\cdot\mathbf{v}_1) &= \mathbf{v}_2\cdot\mathbf{v}_1. \end{aligned} \tag{2}$$

Functional Maps: Functional Maps is an alternative representation of point-wise maps, which is formulated primarily upon the eigenbasis of the Laplace-Beltrami operator. Given a pair of shapes S_1, S_2 , we first compute the first k eigenfunctions with respect to the smallest k eigenvalues and stack them as matrices $\Phi_i \in \mathbb{R}^{n_i\times k}, i = 1,2$. Given a point-wise map encoded as a permutation matrix $\Pi_{21} \in \mathbb{R}^{n_2\times n_1}$, the functional representation is:

$$C_{12} = \Phi_2^{\dagger}\Pi_{21}\Phi_1 \in \mathbb{R}^{k\times k}, \tag{3}$$

where † denotes the Moore Penrose pseudo-inverse. Regarding the inverse conversion, one can compute point-wise map from S_2 to S_1 via searching the nearest neighborhood of each row of Φ_2 among the rows of $\Phi_1 C_{12}^T$.

One of the key properties of functional maps is that, by introducing the spectral embedding, i.e., Φ_1, Φ_2 , we can express global map priors in simple algebraic forms in terms of C_{12}. For instance, area-preserving maps are supposed to correspond to orthogonal functional maps [OBCS*12]. In other words, one can add $\left\| C_{12}^T C_{12} - I \right\|_2$ as regularization to promote such property, which is also the unsupervised orthogonal loss in our pretraining method. Similarly, we can encourage the map to be isometric by promoting Laplacian commutativity, *i.e.,* minimizing $\|\Delta_2 C_{12} - C_{12}\Delta_1\|$. Taking maps in both direction, we can further enhance bijectivity by minimizing $\|C_{21}C_{12} - I\|$.

Computation of Functional Map: We provide further explanation and deduction for the computation of functional map. In the main article, we discuss about the following formulation:

$$C_{\text{opt}} = \min_{\mathbf{C}_{ij}} \|\Phi_j \mathbf{C}_{ij} - \Pi_{ji}\Phi_i\|^2 + \lambda \left\|\mathbf{C}_{ij}\Delta_i - \Delta_j \mathbf{C}_{ij}\right\|^2, \tag{4}$$

Let $\frac{\partial}{\partial C_{ij}} C_{\text{opt}} = 0$ then we have:

$$\begin{aligned} &\frac{\partial}{\partial C_{ij}} (\|\Phi_j \mathbf{C}_{ij} - \Pi_{ji}\Phi_i\|^2 + \lambda \left\|\mathbf{C}_{ij}\Delta_i - \Delta_j \mathbf{C}_{ij}\right\|^2) \\ &\quad = 2\Phi_j^{\dagger}(\Phi_j C_{ij} - \Pi_{ji}\Phi_i) + 2\lambda\Delta \cdot C_{ij} = 0, \end{aligned} \tag{5}$$

where the operation $\cdot$ represents the element-wise multiplication, and $\Delta_{mn} = (\mu_n^j - \mu_m^i)^2$, where μ_l^j and μ_l^i respectively correspond to the l^{th} eigenvalues of Δ_j and Δ_i. Inspired by [DSO20], for every row c_r of $\mathbf{C_{ij}}$:

$$\left(\Phi_j^{\dagger}\Phi_j + \lambda \operatorname{diag}\left(\left(\mu_m^j - \mu_n^i\right)^2\right)\right) c_r = \Phi_i^T b_r, \tag{6}$$

where b_r stands for r^{th} row of $\Pi_{ji}\Phi_i$. Since solving a linear system is differentiable in Pytorch, this allows us to estimate the functional map during training.

2. Implementation Details

Dataset: We use D-FAUST [BRPMB17] as the source dataset to render the pretraining dataset. We randomly choose 10 male and 10 female sequences with different poses. During the rendering, in order to ensure diversity and difficulty, we randomly rotate the meshes and the region of the angles is [-72, 60]. We also calculate the percentage of pixels occupied by the human body in the entire

image, with an average value of 7.74%. Thanks to the specialty of DFM, we can handle large deformation and large intervals between the two input frames. Thus, we can easily enlarge the dataset by enlarging the intervals between two frames.

Training Settings: For RAFT, during the pretraining stage, the batch size is 8 and the total steps are 100,000, the learning rate is 4e-4. While finetuning on SHOF, the batch size is 30 and the steps are 100,000 with a learning rate equal to 4e-4. We further finetune on MHOF with another 50,000 steps and the learning rate is 1.25e-4. For GMA, all the learning rates are the same as RAFT. During the pretraining stage, the batch size is 8 and the total steps are 120,000. While finetuning on SHOF, the batch size is 24 with 120,000 steps and we further finetune on MHOF with another 60,000 steps.

Network Structure: As mentioned in the main article, we can divide the common optical flow network into an image feature extractor and optical flow estimator. As is shown in Fig. 1, for RAFT [TD20], we view the part highlighted in the blue box as the image feature extractor. In the pretraining stage, we use the whole network to get the estimated optical and use DFM to enhance the image feature extractor. During the finetuning stage, we just load the parameters of the image feature extractor and freeze the batch normalization information, while the flow estimator is trained from raw and the feature extractor will also be updated. For GMA [JCL*21], we do the same process, and the image feature extractor is also highlighted in the blue box, as is shown in Fig. 2.

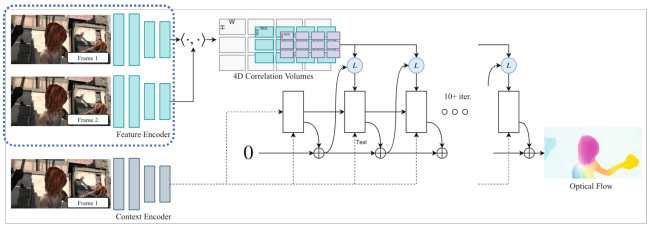

Figure 1: *The whole pipeline image is from RAFT [TD20]. We pretrain the feature extractor highlighted in the blue box.*

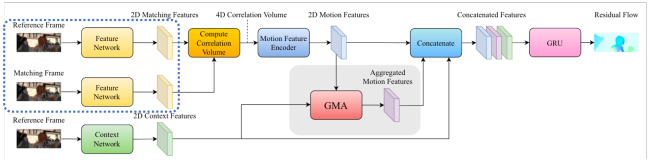

Figure 2: *The whole pipeline image is from GMA [JCL*21]. We pretrain the feature extractor highlighted in the blue box.*

3. Qualitative Results

Pretraining Dataset: To demonstrate the connection between shape matching in DFM and optical flow estimation between two frames, we further visualize both the point cloud matching and estimated optical flow, as is shown in Fig. 4. For each frame, we have its corresponding point cloud. We can visualize the shape matching results constrained by DFM through a color map. By mapping the colors of the source shape onto the target shape, we can find the error parts clearly and directly. The better the shape matching result is, the more consistent the colors on the target shape are with those

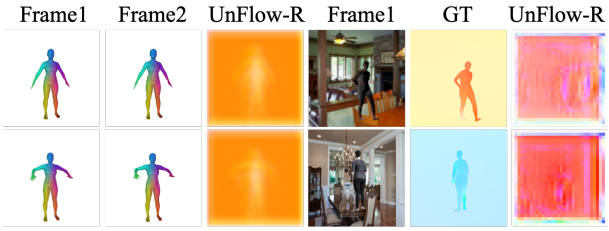

Figure 3: *We visualize the results on both D-FAUST and SHOF dataset.*

on the source shape. For RAFT, we use the given Sintel checkpoint and we compare our pretrained model with it. The interval between the two frames is 20. Though RAFT-S can estimate a quite good optical flow, when translating the flow into point cloud correspondence, it will fail at legs or hands, as is highlighted in the red box. Our pretrained model can fix such errors and provide both accurate point cloud matching and optical flow estimation, further demonstrating the strong combination between 3D shape matching and 2D optical flow estimation.

UnFlow-RAFT: We provide qualitative results for UnFlow-RAFT, which we use the unsupervised losses in [MHR18] at the pretraining stage. Although the network can converge quickly during the training process, it will produce the same homogeneous output results, as is shown in Fig.3. One reason is that unsupervised flow estimation mainly depends on the similarity between two frames, while DFM can inherently solve complicated cases with large deformations or non-rigid motions. It is also insensitive to transformation. Therefore, during pretraining, we included frames with up to 20 intervals and transformation augmentation, which are challenging for unsupervised flow estimation methods. Another reason is the domain gap between pretraining dataset and SHOF. For the pretraining dataset, we aim to leverage geometric information and thus focus less on human appearance. To maintain color consistency, we use a UV map to texture meshes based on the original correspondences during data preparation. However, SHOF leverage the details and textures of human appearance. Thus, Since the feature extractor based on UnFlow can not learn useful and generalizing information from the pretraining task, it easily fails in the fine-tuning step of SHOF, indicating that our pretraining task can indeed leverage 3D information and help improve the optical flow estimation.

SHOF with large intervals: We visualize the optical flow between two frames with large gaps, for instance, 20 frames. As is shown in Fig. 5, with our pretraining task, both RAFT and GMA can better handle human motions and effectively distinguish human motion from the background. Moreover, the motion directions of different body parts are recognized, such as arms and head. Note that the regarding GT cannot be naively composed by the GTs regarding consecutive frames. Simply accumulating the optical flow between frames will result in the superposition of intermediate actions. Therefore, we can just compare these methods from qualitative aspect.

DAVIS Dataset: To further compare the generalization and performance, we test the MHOF models on DAVIS [PPTM*16], a real-world dataset for video object segmentation. As is shown in Fig. 6,

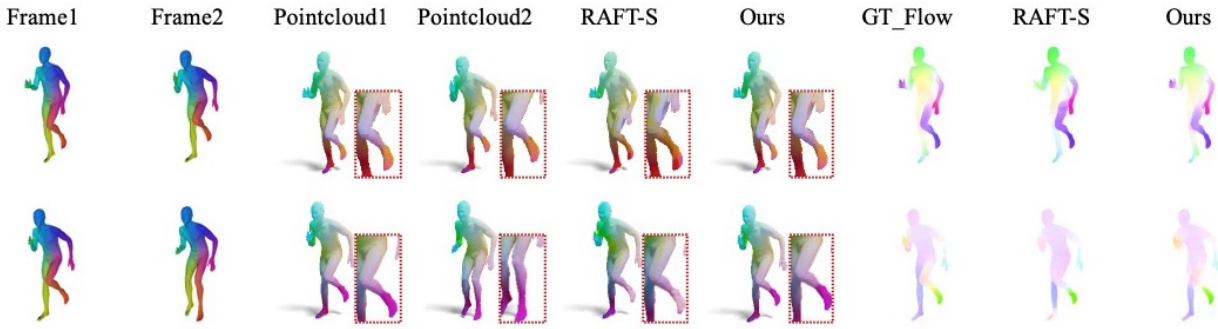

Figure 4: *Visualization of pretraining dataset. '-S' means using the given Sintel checkpoint.*

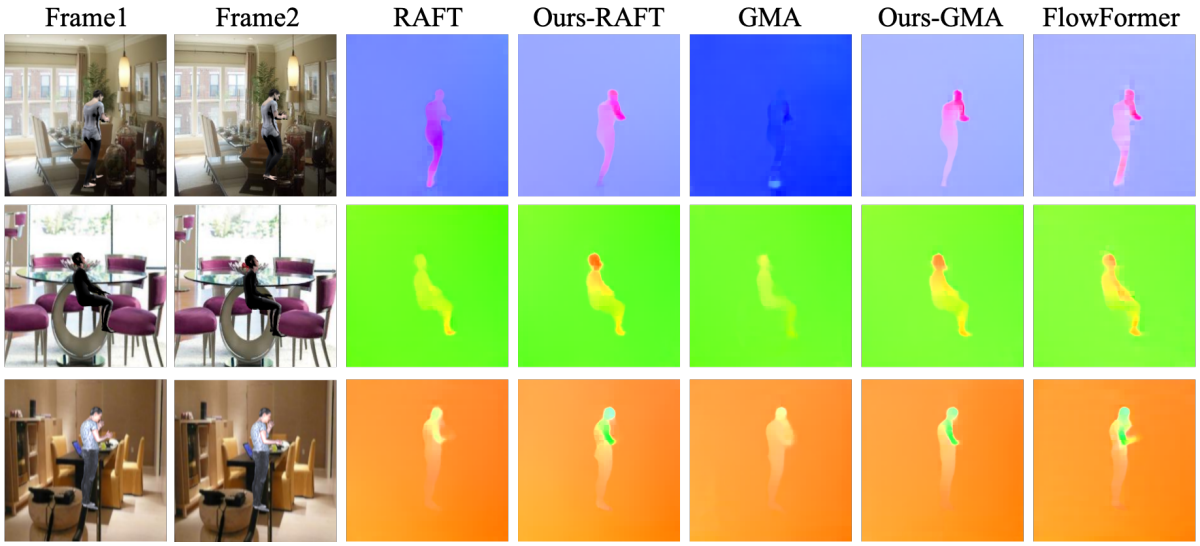

Figure 5: *Visualization of SHOF with the interval of 20 frames.*

for both single human and multi-human, our pretraining task can help improve the estimation of human parts like legs and provide more precise identification of the human contour, leading to optical flow with better granularity and accuracy. Moreover, with our pretraining, some errors can be fixed for both two backbones.

DeepDeform Dataset: We provide several visual results for DeepDeform [BZTN20] with different categories, including non-rigid and rigid objects, as is shown in Fig. 7. For non-rigid situations, such as humans, we can provide more precise and accurate optical flow estimation compared with the two strong baseline and backbone methods. While for nearly rigid things, all methods tend to view it as a whole part and are insensitive to rotation. However, the results still demonstrate that our pretraining method can be applied to various scenarios and is not only limited to human optical flow estimation.

References

[BRPMB17] BOGO F., ROMERO J., PONS-MOLL G., BLACK M. J.: Dynamic faust: Registering human bodies in motion. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 6233–6242. 1

[BZTN20] BOZIC A., ZOLLHOFER M., THEOBALT C., NIESSNER M.: Deepdeform: Learning non-rigid rgb-d reconstruction with semi-supervised data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 7002–7012. 3

[DSO20] DONATI N., SHARMA A., OVSJANIKOV M.: Deep geometric functional maps: Robust feature learning for shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 8592–8601. 1

[JCL*21] JIANG S., CAMPBELL D., LU Y., LI H., HARTLEY R.: Learning to estimate hidden motions with global motion aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 9772–9781. 2

[MHR18] MEISTER S., HUR J., ROTH S.: Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *Proceedings of the AAAI conference on artificial intelligence* (2018), vol. 32. 2

[OBCS*12] OVSJANIKOV M., BEN-CHEN M., SOLOMON J., BUTSCHER A., GUIBAS L.: Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (ToG) 31*, 4 (2012), 1–11. 1

[PPTM*16] PERAZZI F., PONT-TUSET J., MCWILLIAMS B., VAN GOOL L., GROSS M., SORKINE-HORNUNG A.: A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition* (2016). 2

[TD20] TEED Z., DENG J.: Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Confer-*

Figure 6: *Visualization of DAVIS dataset. The main differences are highlighted in red boxes.*

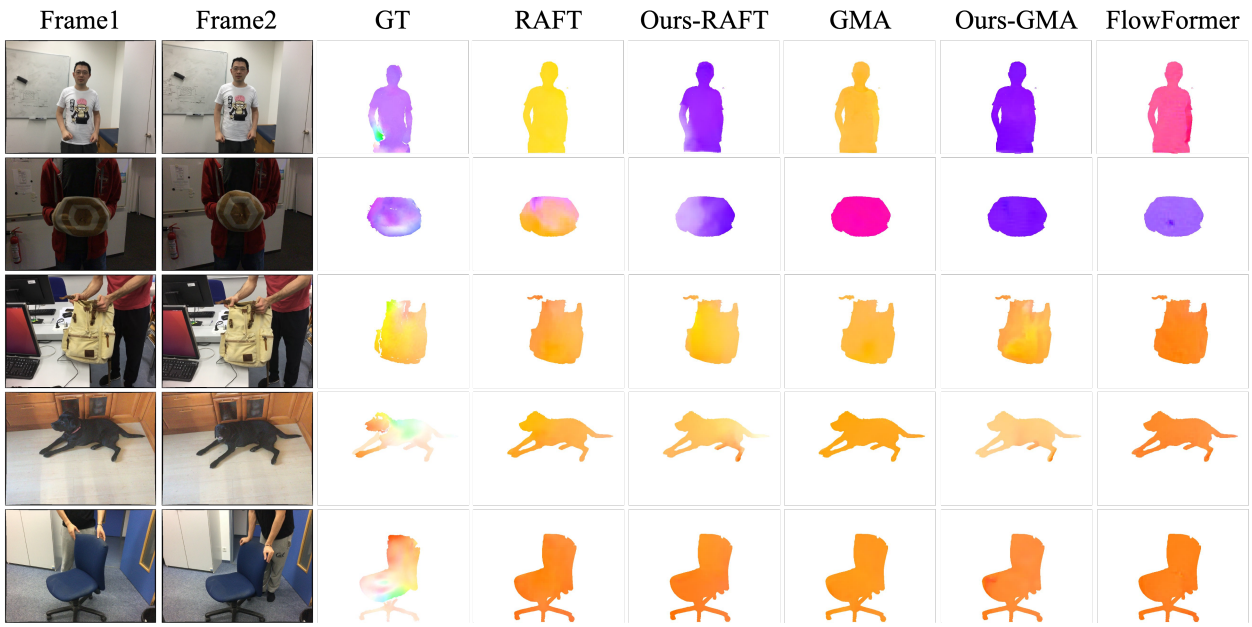

Figure 7: *Visualization of DeepDeform dataset with different categories.*

ence, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16 (2020), Springer, pp. 402–419. 2