Unsupervised Deep Multi-Shape Matching – SUPPLEMENTARY DOCUMENT –

Dongliang Cao[1,2] and Florian Bernard[1]

[1] Technical University of Munich, 80333 Munich, Germany
[2] University of Bonn, 53115 Bonn, Germany

In this supplementary document we first introduce the implementation details of our method. Subsequently, we provide details on the unsupervised loss for partial shape matching. Afterwards, we discuss our network fine-tuning. Eventually, we also show additional qualitative results of our method.

S1 Implementation details

We implemented our method in PyTorch. Our feature extractor takes 352-dimensional pre-computed SHOT descriptors [40] as inputs. We use DiffusionNet [42] composed of 4 diffusion blocks with width 128 as the network architecture for both our feature extractor and universe classifier. In the context of the FM solver, we set $\lambda = 100$ in Eq. (9) and $\gamma = 0.5$ in Eq. (10) for our partial shape matching (for complete shape matching we use $\lambda = 0$). For the basis functions for functional maps computation, we choose the number to be 80 for the FAUST and SCAPE datasets for full shape matching. For partial shape matching, we choose the number to be 50 and 30 for the CUTS and HOLES subsets of the SHREC'16, respectively, to be consistent with DPFM [2]. We apply Sinkhorn normalisation with the number of iterations equal to 10 and the temperature parameter τ equal to 0.2.

If a dataset provides ground truth correspondences based on a reference shape, we set the number of universe vertices to the number of vertices of the reference shape. Otherwise, we set the number of universe vertices to the largest number of vertices among the given shapes. For network training, we use $w_{\text{bij}} = 1.0, w_{\text{orth}} = 1.0, w_{\text{lap}} = 10^{-3}$ for $\mathcal{L}_{\text{ft}}$ in Eq. (12) (for partial shape matching we use $w_{\text{lap}} = 0$, since in this case we already enforce Laplacian commutativity regularisation in our regularised FM solver). The final loss is a linear combination of $\mathcal{L}_{\text{ft}}$ and $\mathcal{L}_{\text{cls}}$, where we set $\lambda_{\text{cls}} = 0.01$ for complete shape matching. The loss for the universe classifier $\mathcal{L}_{\text{cls}}$ is slightly different from Eq. (14) for partial shape matching, for which we provide the details in Sec. S2. We train our network with a batch size of 1 for all datasets. We use the ADAM optimiser with a learning rate of 10^{-3} for all experiments. The total number of training iterations for each dataset is 20000. During the first 4000 training iterations, when computing $\mathcal{L}_{\text{cls}}$ defined in Eq. (14), we detach the gradient for C_{yx} and only regularise it based on its structural properties defined in $\mathcal{L}_{\text{ft}}$. Afterwards, we will use the gradients for both C_{yx} and Π_{xy} to optimise our network. In this way, it can lead to faster convergence and better network performance.

S2 Unsupervised loss for partial shape matching

In the context of partial-to-complete shape matching, we can assume that the complete shape plays the role of the universe shape, since it is guaranteed that each point in the partial shapes is in correspondence with exactly one point in the complete shape. We modify the unsupervised loss for universe classifier based on it. For $\mathcal{X}$ being the complete shape and $\mathcal{Y}$ being the partial shape, the loss term can be expressed in the form

$$\mathcal{L}_{\text{cls}} = \mathcal{L}_{\text{ce}}^{\text{smooth}}(\Pi_x, \mathbf{I}_d) + \mathcal{L}_{\text{ce}}^{\text{smooth}}(\Pi_y, \hat{\Pi}_y), \tag{S1}$$

where $\mathbf{I}_d$ is the identity matrix of size d, $\hat{\Pi}_y$ is the partial-to-complete correspondences obtained by nearest neighbour search between $\Phi_y C_{xy}$ and Φ_x, and $\mathcal{L}_{ce}^{\text{smooth}}$ is the cross entropy loss with label smoothing, where we set the smoothing factor equal to 0.1. The first term of the equation encourages the correspondences between the complete shape and the (virtual) universe shape to be identical, while the second term regularises the predicted partial-to-universe correspondence based on functional map regularisation. Similar to complete shape matching, the total unsupervised loss is a linear combination of $\mathcal{L}_{\text{ft}}$ and $\mathcal{L}_{\text{cls}}$, where we set $\lambda_{\text{cls}} = 1.0$.

S3 Network fine-tuning

We observe that the generalisation ability of our method across different datasets can be improved by network fine-tuning, as shown in Fig. 3 (main paper). In order to achieve this, we first train our network on the training dataset in the ordinary way, and afterwards we use an unsupervised fine-tuning of the pre-trained network for the test dataset. Specifically, during fine-tuning, we update the network weights for each shape pair independently. To this end, we use the same loss defined in Eq. (15) to optimise the network with a fixed number of five forward/backward passes (for each shape pair individually). The advantage of network fine-tuning compared to post-processing techniques is that it directly optimises the network itself, thus leading to better performance.

S4 Additional qualitative results

We show additional qualitative results on the FAUST dataset in Fig S1, on the SCAPE dataset in Fig S2, as well as on the SHREC'16 datset in Fig S3. Our method predicts shape-to-universe correspondences for each shape to obtain cycle-consistent multi-shape matchings among a collection of shapes.

Fig. S1: Qualitative multi-matching results using our method on the FAUST dataset.

Fig. S2: Qualitative multi-matching results using our method on the SCAPE dataset.

Fig. S3: Qualitative partial-to-partial multi-matching results using our method on the SHREC'16 dataset. The full shape is shown merely for visualisation purposes (colour code).

S5 Inter-class shape matching

We evaluate our method for the challenging inter-class multi-shape matching on the TOSCA dataset.

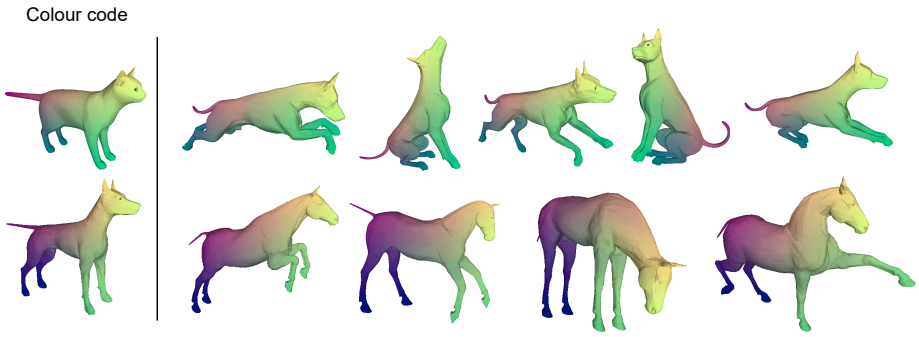

Fig. S4: Qualitative inter-class multi-matching results using our method on the TOSCA dataset.

S6 Shape matching on SHREC'19 dataset

We evaluate our method on the more challenging SHREC'19 dataset. Furthermore, we randomly remesh each shape to different resolution to evaluate the robustness of our method with respect to different meshings.

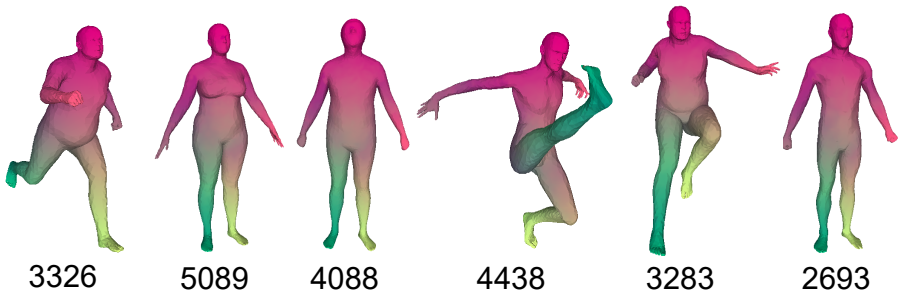

Fig. S5: Qualitative shape matching results using our method on the SHREC'19 dataset with different resolution (numbers refer to #vertices).