Quasi-Newton Solver for Robust Non-Rigid Registration
(Supplementary material)

Yuxin Yao[1]   Bailin Deng[2]   Weiwei Xu[3]   Juyong Zhang[1*]

[1]University of Science and Technology of China   [2]Cardiff University   [3]Zhejiang University

yaoyuxin@mail.ustc.edu.cn   DengB3@cardiff.ac.uk   xww@cad.zju.edu.cn   juyong@ustc.edu.cn

1. The choice of sampling radius

The number of graph nodes and edges will influence the memory footprint and computational cost for the solver. The farthest point sampling method [1] is to repeatedly add the farthest point to the graph until the geodesic distance between graph nodes and the farthest point is smaller than the given radius parameter R. Compared with farthest point sampling method(Fig. 1), our adopt method can obtain fewer nodes and is faster to converge with the similar accuracy. In our method, the radius R can be used to balance the speed and accuracy. A smaller R leads to more nodes in the deformation graph, which increases the number of variables and accuracy while requires more computational time. We show the comparison in Fig. 2. Our method does not vary the sampling density based on curvature. In our experiments, such uniform density is sufficient to generate good results and curvature-adaptive sampling can be a future work.
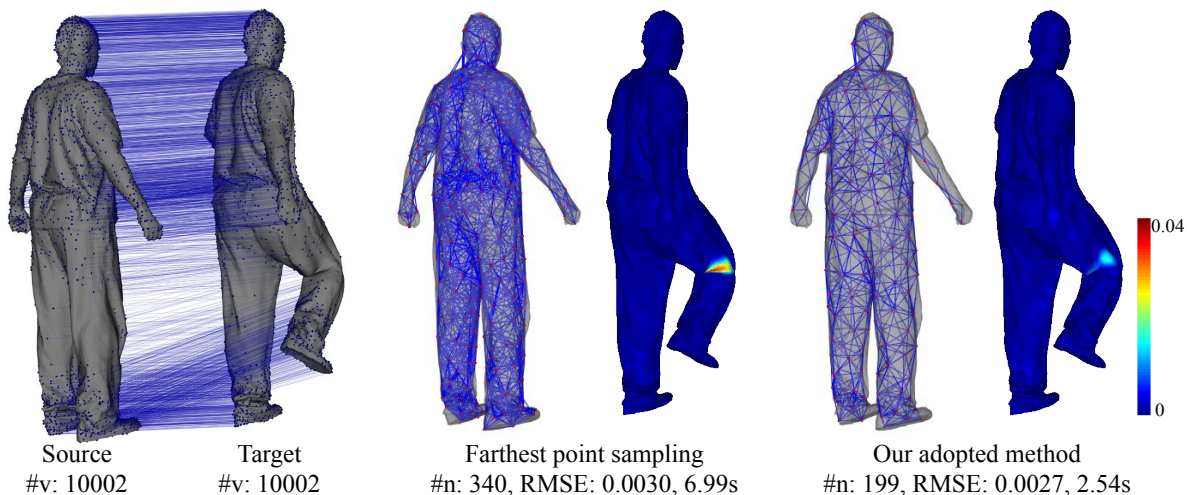

Source	Target	Farthest point sampling	Our adopted method
#v: 10002	#v: 10002	#n: 340, RMSE: 0.0030, 6.99s	#n: 199, RMSE: 0.0027, 2.54s

Figure 1. Comparison with the farthest sampling method for the given radius $R = 5\bar{l}$, where $\bar{l}$ is the average edge length on the source surface. The farthest point sampling method can obtain more graph nodes.($k_\alpha = 0.001, k_\beta = 0.1$). The RMSE and the color-coded registration errors are in the unit of meters.

2. The comparison with fixed parameters

In our method, the ν_a and ν_r values will influence the registered result, and discussion on this part is given in "Choosing ν_a and ν_r"(Sec 4.2) of the paper. In Fig. 3, we show the comparison between our dynamic adjustment strategy and the strategy by fixing ν_a, ν_r, and we can see our method can get higher accuracy.

---

*Corresponding author
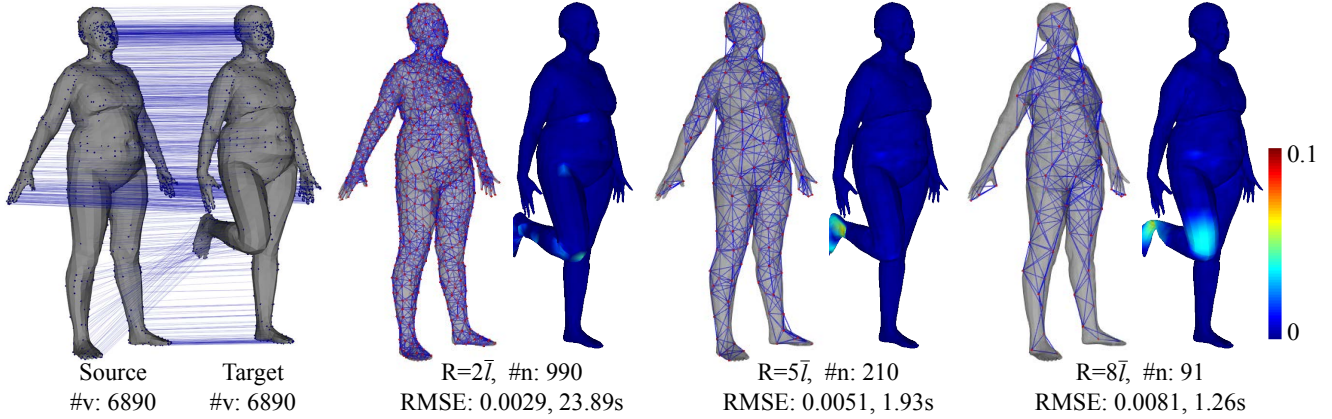
Figure 2. Comparison with different sampling radius with $k_\alpha = 0.001, k_\beta = 0.1$. More nodes can get more accurate results. The RMSE and the color-coded registration errors are in the unit of meters.
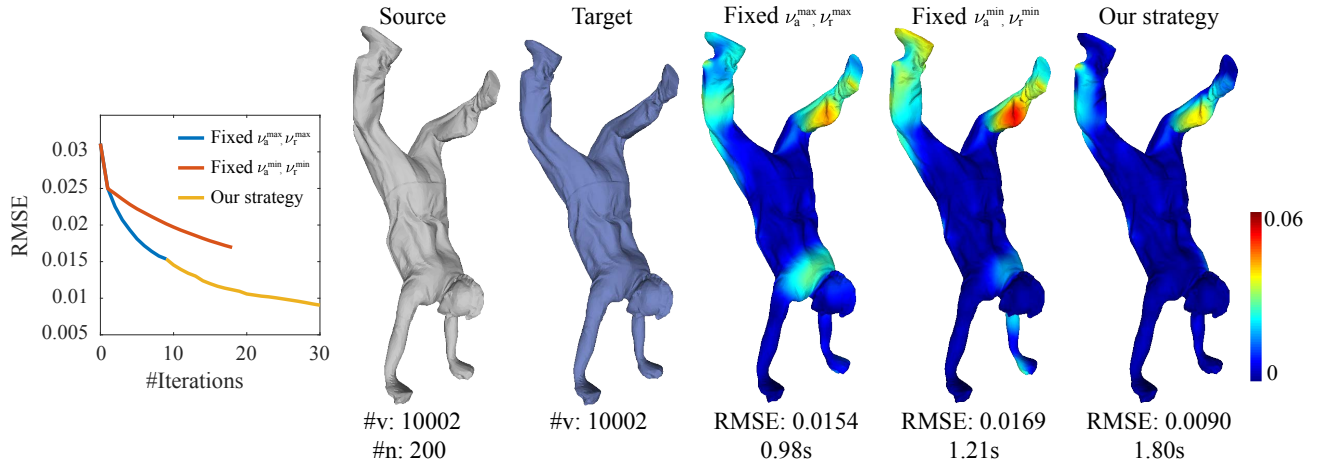

Figure 3. Comparison with fixed ν_a, ν_r on 42-th to 40-th mesh in "handstand" with $k_\alpha = 100$ and $k_\beta = 50$. Here ν_r^{min} is the value when ν_a reaches ν_a^{min}. The RMSE and the color-coded registration errors are in the unit of meters.

3. Experiment on clean data

We show more results on five models "crane", "march1", "samba", "squat1" and "swing" in Human-motion datasets. For each model, we use the closest points to construct the correspondences for small deformation, and use the SHOT with diffusion pruning method for big deformation. For each method, we search the parameter setting for best performance. The results are shown in Fig. 4 and Fig. 5, and we can see that our method is faster than other methods and achieves similar or better accuracy.

4. Experiment on partially overlapping data

We show more results and comparisons on partially overlapping data from "bouncing" datasets. We choose $\alpha = 10$ for N-ICP, $\alpha = 1, \beta = 100$ for RPTS, $\alpha = 0.1, \beta = 100$ for SVR-ℓ_0, and $k_\alpha = 1, k_\beta = 100, \nu_a^{max} = 30\bar{d}, \nu_r^{max} = 100\bar{l}$ for our method and $\theta = 45°$ for all method in these examples. The results are shown in Fig. 6, and we can see that our methods is robust to partially overlapping data, and faster than other methods.

References

[1] Carsten Moenning and Neil A Dodgson. Fast marching farthest point sampling. Technical report, University of Cambridge, Computer Laboratory, 2003. 1
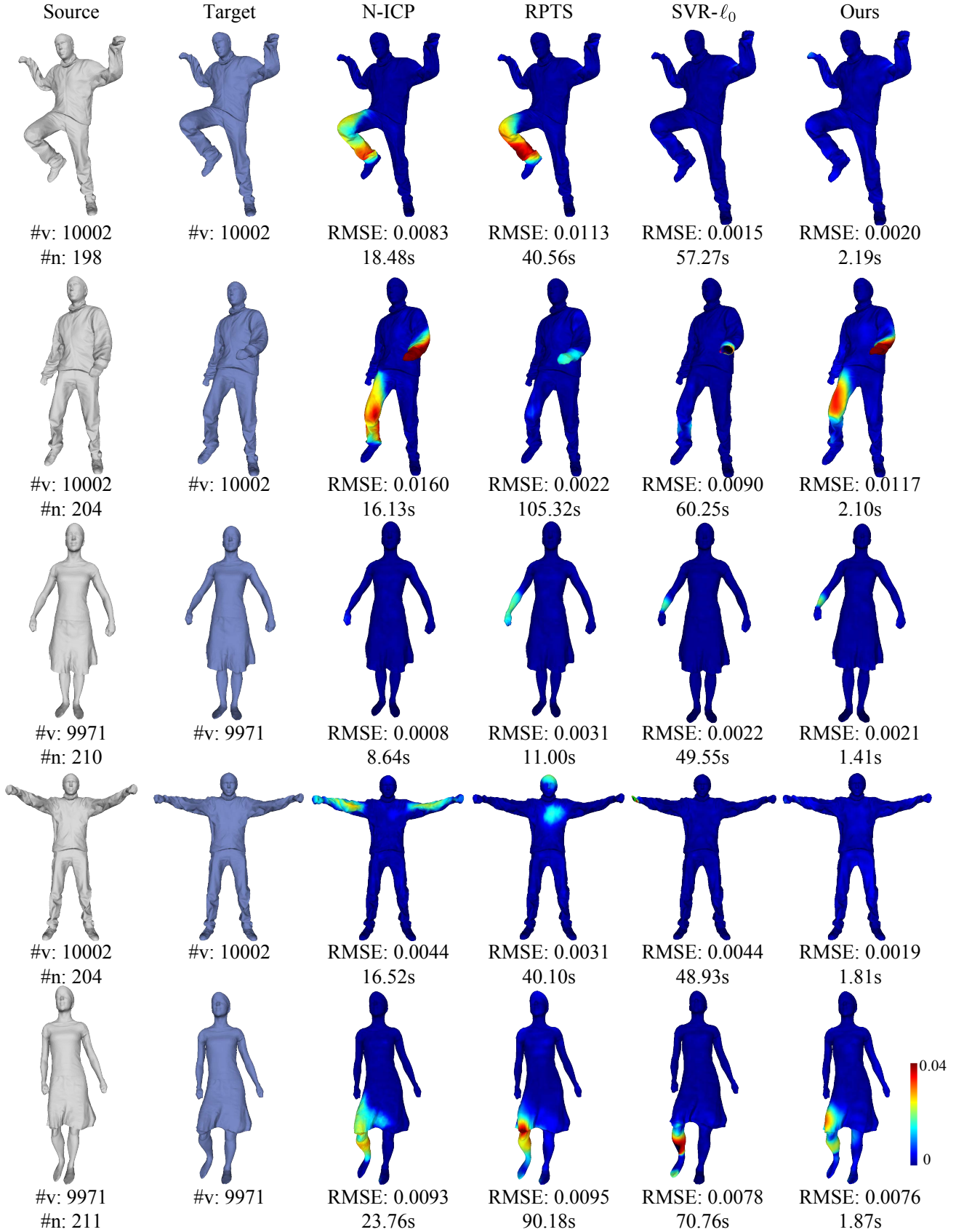
Source	Target	N-ICP	RPTS	SVR-ℓ_0	Ours

#v: 10002   #v: 10002   RMSE: 0.0083   RMSE: 0.0113   RMSE: 0.0015   RMSE: 0.0020
#n: 198   18.48s   40.56s   57.27s   2.19s

#v: 10002   #v: 10002   RMSE: 0.0160   RMSE: 0.0022   RMSE: 0.0090   RMSE: 0.0117
#n: 204   16.13s   105.32s   60.25s   2.10s

#v: 9971   #v: 9971   RMSE: 0.0008   RMSE: 0.0031   RMSE: 0.0022   RMSE: 0.0021
#n: 210   8.64s   11.00s   49.55s   1.41s

#v: 10002   #v: 10002   RMSE: 0.0044   RMSE: 0.0031   RMSE: 0.0044   RMSE: 0.0019
#n: 204   16.52s   40.10s   48.93s   1.81s

#v: 9971   #v: 9971   RMSE: 0.0093   RMSE: 0.0095   RMSE: 0.0078   RMSE: 0.0076
#n: 211   23.76s   90.18s   70.76s   1.87s

Figure 4. Comparison with N-ICP, RPTS and SVR-ℓ_0 on "crane", "march1", "samba", "squat1" and "swing" datasets with small deformation. We set $\alpha = 10$ for N-ICP, $\alpha = 10$ and $\beta = 1$ for RPTS, $\alpha = 0.1$ and $\beta = 100$ for SVR-ℓ_0, and $k_\alpha = 1, k_\beta = 10^3, \nu_a = 30\bar{l}$ for our method in these examples. The RMSE and the color-coded registration errors are in the unit of meters.
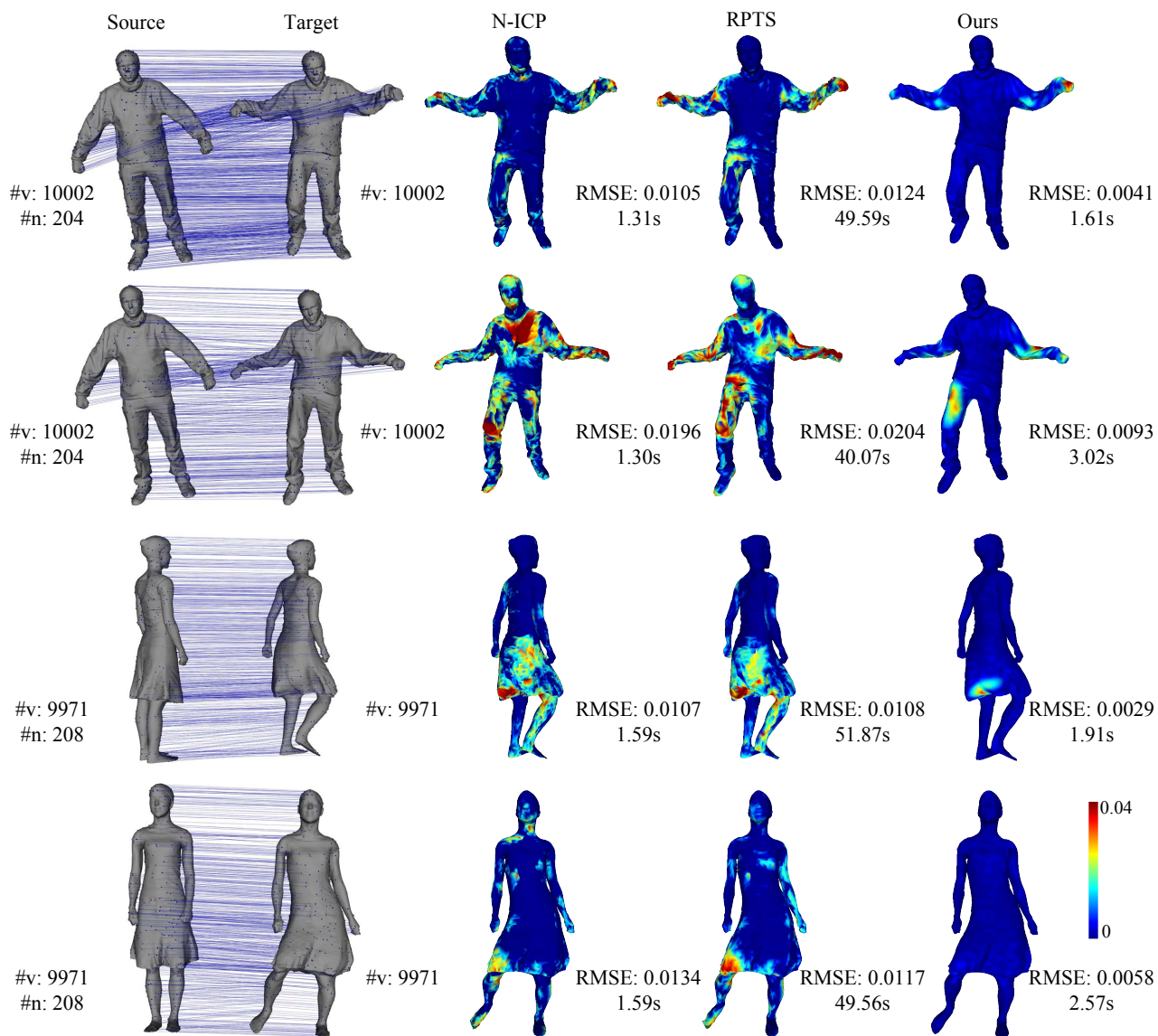
Figure 5. Comparison with N-ICP, RPTS and SVR-ℓ_0 on "crane",and "swing" datasets with big deformation. We set $\alpha = 0.01$ for N-ICP, $\alpha = 0.01$ and $\beta = 1$ for RPTS, and $k_\alpha = 0.01$ and $k_\beta = 1$ for our method in these examples. The RMSE and the color-coded registration errors are in the unit of meters.
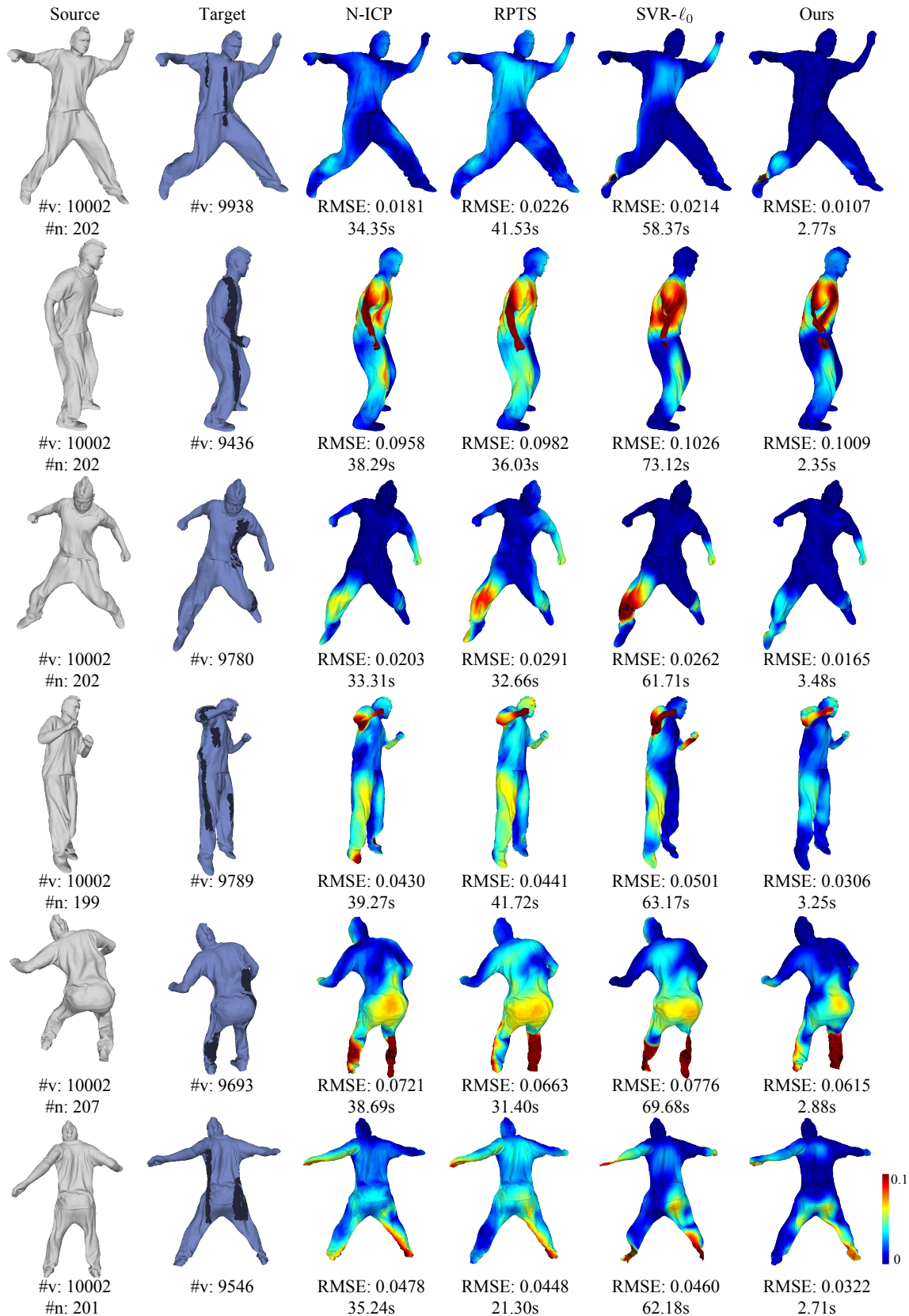
Figure 6. Comparison with N-ICP, RPTS and SVR-ℓ_0 on "bouncing" datasets with partially overlapping data. The RMSE and the color-coded registration errors are in the unit of meters.