Best Buddies Registration for Point Clouds - Supplementary Material

Amnon Drory, Tal Shomer, Shai Avidan, and Raja Giryes

Tel Aviv University, Israel

1 Overview

This document contains supplemental material for the paper *Best Buddies Registration for Point Clouds*. It includes:

1. Diagram of implementation of BBR as a Neural Network (figure 1).
2. Additional experimental results
3. Definitions of angular and translational error

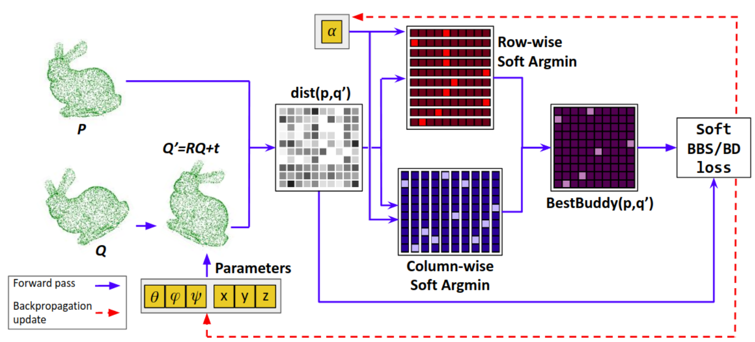

Fig. 1: **I**mplementation of Best-Buddies Registration (BBR) as a Neural Network. Registration of a pair of point clouds is equivalent to "training" this neural network. Unlike in a typical neural network setting, the weights are not learned from a training set. Instead, performing "training" (optimization) on a pair of input point clouds P and Q is equivalent to a gradient-descent search for the optimal rigid transformation between them (equation 1 in main paper). Each forward pass calculates the loss for the current value of $R = R(\theta, \phi, \psi)$ and $t = (x, y, z)$. The back propagation step updates the parameters to improve the match between P and Q. The network's weights hold the result of the optimization: the 6 parameters of the transformation, and the temperature parameter α of the soft-argmin function.

Fig. 2: **TUM-RGB-D pair used in experiment in main paper:** Left - RGB-D image from the TUM-RGBD data set [1]. This data is captured by the Kinect sensor, and exhibits warp and scanning noise. Right - Sampled point clouds of 1000 points of the same scene, emphasizing the small area of overlap.

2 Additional Experimental Results

Run-time. Table 1 shows the time it takes to run a single iteration of the different variants of our loss function, as a function of the number of points (measured with a PyTorch implementation running on a GTX 980 Ti GPU). All of the algorithm variants typically converge in few hundreds of iterations, depending on the learning rate.

Points	*BBR-softBBS*	*BBR-softBD*	*BBR-N*	*BBR-F*
200	4	4	4	6
500	4	4	6	6
1000	8	8	13	8
5000	140	140	240	24
30000	-	-	-	125

Table 1: Average running time of a single gradient descent iteration in milliseconds, as a function of the number of points in the cloud. Due to memory limitations, only BBR-F is able to handle 30000 points.

Convergence. In the accuracy section of the paper, we demonstrated the ability of *BBR-softBD*, *BBR-N* and *BBR-F* to handle cases where the initial error is of the order of up to 10°. In this section we show that *BBR-softBBS* can be used to register in situations where the initial error is much larger. This is due to its large basin of convergence. In this section we use the same Bunny, Horse and Dragon models that were used in the accuracy test, this time with a large random rotation in range $\theta_{rot} \in [30, 60]$ degrees, and run *BBR-softBBS*. We set $\Delta_{trans} = 0.005m$ and $T = 20$. As can be seen in Figure 3, *BBR-softBBS* manages to reduce the rotation error significantly, to below 3°, in all experiments.

Measurement noise. In the main paper we presented an experiment using a pair of scans from the TUM RGB-D dataset [1], shown in figure 2. Here we present registration results on an additional pair of scans, 1305031794.813436 and 1305031794.849048 of the freiburg1_xyz sequence from the TUM RGB-D (Figure 4). We sample 1500 points from each, and experiment with adding a

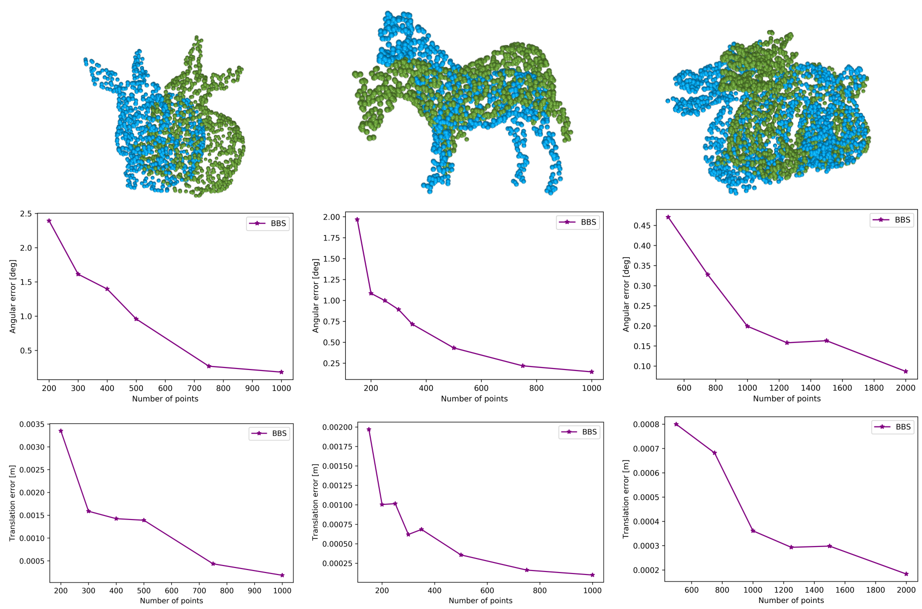

Fig. 3: **C**onvergence test: Top - Point clouds examples as used in the accuracy test. In this experiment, θ_{rot} was randomized in range $= [30, 60]$ degrees. In the point cloud visualizations above, the Bunny (left) is rotated by $50°$. The Horse (middle) is rotated by $30°$. The Dragon (right) is rotated by $45°$. Center and Bottom - angular and translation error as a function of the number of points. In all cases, *BBR-softBBS* (labeled **BBS**) manages to reduce the large initial rotation error significantly.

random rotation of up to 5° around a random axis. We repeat this 50 times and perform registration for each, showing the cumulative distribution of the final errors in Figure 5. *BBR-softBD* (labeled **BBR**) performs considerably better than either Sym-ICP [2] or CPD [3], showing its robustness to realistic measurement noise and occlusions.

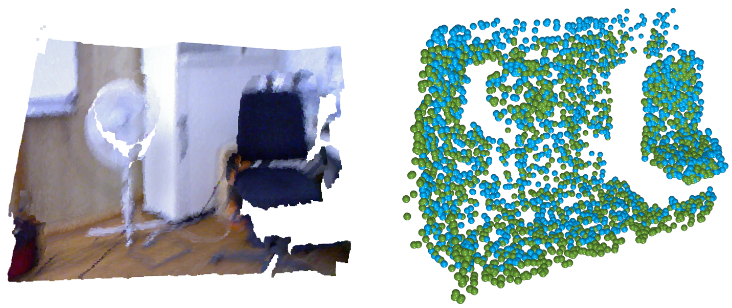

Fig. 4: **A**dditional TUM-RGB-D pair: Left - RGB-D image from the TUM-RGBD data set [1]. This data is captured by the Kinect sensor, and exhibits warp and scanning noise. Right - Sampled point clouds of 1500 points of the same scene.

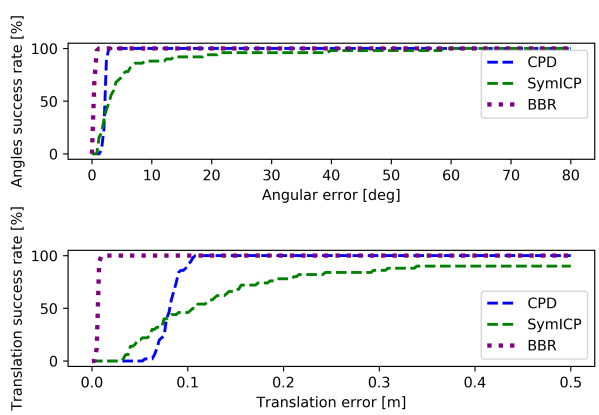

Fig. 5: **C**onvergence Analysis (TUM): The cumulative distribution of errors over 50 repeats. The x-axis is the error threshold and the y axis is the fraction of results that achieved an error smaller than this threshold. *BBR-softBD* is labeled **BBR**

3 Definitions of Angular and Translational Error

In all our experiments, we follow Lu [4] in defining angular distance as *chordal distance* [5], and translational distance as the Euclidean norm of the difference between two translation vectors.

References

1. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, Vilamoura, Algarve, Portugal, October 7-12, 2012. (2012) 573–580
2. Rusinkiewicz, S.: A symmetric objective function for ICP. ACM Transactions on Graphics (Proc. SIGGRAPH) **38** (2019)
3. Myronenko, A., Song, X.: Point set registration: Coherent point drift. IEEE Trans. Pattern Anal. Mach. Intell. **32** (2010) 2262–2275
4. Lu, W., Wan, G., Zhou, Y., Fu, X., Yuan, P., Song, S.: Deepvcp: An end-to-end deep neural network for point cloud registration. In: The IEEE International Conference on Computer Vision (ICCV). (2019)
5. Hartley, R.I., Trumpf, J., Dai, Y., Li, H.: Rotation averaging. International Journal of Computer Vision **103** (2012) 267–305