Eurographics Symposium on Geometry Processing 2019
D. Bommes and H. Huang
(Guest Editors)

Volume 38 (2019), Number 5

On evaluating consensus in RANSAC surface registration: Supplementary material

L. Hruda[1], J. Dvořák[1] and L. Váša[2]

[1]NTIS - New Technologies for the Information Society, Faculty of Applied Sciences, University of West Bohemia, Pilsen, Czech Republic
[2]Department of Computer Science and Engineering, Faculty of Applied Sciences, University of West Bohemia, Pilsen, Czech Republic

1. Composed metrics - Order of operations

A rigid transformation can be equivalently defined as either a rotation followed by a translation, i.e. $T(\mathbf{x}) = R\mathbf{x} + \mathbf{t}$, or as a translation followed by a rotation, i.e. $T'(\mathbf{x}) = R'(\mathbf{x} + \mathbf{t}')$. If T and T' describe the same transformation, following must hold:

$$R\mathbf{x} + \mathbf{t} = R'(\mathbf{x} + \mathbf{t}'). \tag{1}$$

This equation holds when $R = R'$ and $\mathbf{t}' = R^T\mathbf{t}$, because then $R'(\mathbf{x} + \mathbf{t}') = R(\mathbf{x} + R^T\mathbf{t}) = R\mathbf{x} + \mathbf{t}$. Applying transformation T to an object corresponds to first changing its orientation to the target orientation by rotation and then translating the object to the target location. Applying T' corresponds to first translating the object to such a position that the subsequent rotation, which enforces the target orientation, also moves the object to the target location. The change of orientation is the same in both cases. Therefore, Equation 1 can be rewritten as follows:

$$R\mathbf{x} + \mathbf{t} = R(\mathbf{x} + \mathbf{t}') \tag{2}$$

$$R\mathbf{x} + \mathbf{t} = R\mathbf{x} + R\mathbf{t}' \tag{3}$$

$$\mathbf{t} = R\mathbf{t}' \Rightarrow R^T\mathbf{t} = \mathbf{t}' \tag{4}$$

Suppose there are two arbitrary rigid transformations T_1 and T_2 expressed in the rotation-first form, i.e. $R_1\mathbf{x} + \mathbf{t}_1$, $R_2\mathbf{x} + \mathbf{t}_2$, and translation-first form, i.e. $R_1'(\mathbf{x} + \mathbf{t}_1')$, $R_2'(\mathbf{x} + \mathbf{t}_2')$. Since $R_1 = R_1'$ and $R_2 = R_2'$, $d_R(R_1, R_2) = d_R(R_1', R_2')$ for any rotation metric d_R. The translation metrics $\|\mathbf{t}_1 - \mathbf{t}_2\|$ and $\|\mathbf{t}_1' - \mathbf{t}_2'\|$ are, however, different. The first metric can be expanded as shown in Equation 5.

$$\|\mathbf{t}_1 - \mathbf{t}_2\| = \sqrt{(\mathbf{t}_1 - \mathbf{t}_2)^T(\mathbf{t}_1 - \mathbf{t}_2)} = \sqrt{\mathbf{t}_1^T\mathbf{t}_1 - 2\mathbf{t}_1^T\mathbf{t}_2 + \mathbf{t}_2^T\mathbf{t}}. \tag{5}$$

From Equation 4 it follows that $\|\mathbf{t}_1' - \mathbf{t}_2'\| = \|R_1^T\mathbf{t}_1 - R_2^T\mathbf{t}_2\|$ and the second metric can, therefore, be expanded as follows:

$$\begin{aligned}\|\mathbf{t}_1' - \mathbf{t}_2'\| = \|R_1^T\mathbf{t}_1 - R_2^T\mathbf{t}_2\| = \sqrt{(R_1^T\mathbf{t}_1 - R_2^T\mathbf{t}_2)^T(R_1^T\mathbf{t}_1 - R_2^T\mathbf{t}_2)} = \\ = \sqrt{\mathbf{t}_1^T R_1 R_1^T \mathbf{t}_1 - 2\mathbf{t}_1^T R_1 R_2^T \mathbf{t}_2 + \mathbf{t}_2^T R_2 R_2^T \mathbf{t}_2} = \\ = \sqrt{\mathbf{t}_1^T\mathbf{t}_1 - 2\mathbf{t}_1^T R_1 R_2^T \mathbf{t}_2 + \mathbf{t}_2^T\mathbf{t}_2}.\end{aligned} \tag{6}$$

The only case when $\|\mathbf{t}_1 - \mathbf{t}_2\| = \|\mathbf{t}_1' - \mathbf{t}_2'\|$ is when $R_1 = R_2$. The result of the translation metric and, therefore, of the composed transformation distance metric, depends on the order of rotation and translation, which we choose for describing the transformations. Furthermore, the difference between $\|\mathbf{t}_1 - \mathbf{t}_2\|$ and $\|\mathbf{t}_1' - \mathbf{t}_2'\|$ grows with the increasing difference between R_1 and R_2.

2. Composed metrics - Dependence on position

Consider an input object Q_{in} and an arbitrary rigid transformation T that transforms Q_{in} into an output object $Q_{out} = T(Q_{in})$. Translating both Q_{in} and Q_{out} by the same arbitrary vector $\mathbf{t}_0$ results in two objects Q_{in}' and Q_{out}' respectively, with different absolute position but the same mutual position as Q_{in} and Q_{out}. To transform Q_{in}' into Q_{out}' a new transform T' must be defined such that $T'(Q_{in}') = Q_{out}'$. Since translating Q_{out}' by $-\mathbf{t}_0$ results in Q_{out}, $T(\mathbf{x}) = T'(\mathbf{x} + \mathbf{t}_0) - \mathbf{t}_0$ must hold, which can be expanded as

$$R\mathbf{x} + \mathbf{t} = R'(\mathbf{x} + \mathbf{t}_0) + \mathbf{t}' - \mathbf{t}_0. \tag{7}$$

The change of orientation is equal for T and T', and therefore $R = R'$, which implies

$$\begin{aligned} R\mathbf{x} + \mathbf{t} = R(\mathbf{x} + \mathbf{t}_0) + \mathbf{t}' - \mathbf{t}_0 \\ \mathbf{t}' = \mathbf{t} - R\mathbf{t}_0 + \mathbf{t}_0 \end{aligned} \tag{8}$$

Consider two different rigid transformations T_1 and T_2 and two corresponding transformations T_1' and T_2', where $\mathbf{t}_1' = \mathbf{t}_1 - R_1\mathbf{t}_0 + \mathbf{t}_0$ and $\mathbf{t}_2' = \mathbf{t}_2 - R_2\mathbf{t}_0 + \mathbf{t}_0$, according to Equation 8, and some arbitrary vector $\mathbf{t}_0$. Since $R_1 = R_1'$ and $R_2 = R_2'$, it follows that $d_R(R_1, R_2) = d_R(R_1', R_2')$, but the translation metrics $\|\mathbf{t}_1 - \mathbf{t}_2\|$ and $\|\mathbf{t}_1' - \mathbf{t}_2'\|$ are generally different. The first one expands as shown in Equation 5, while the second metric expands as

$$\|\mathbf{t}_1' - \mathbf{t}_2'\| = \|\mathbf{t}_1 - R_1\mathbf{t}_0 + \mathbf{t}_0 - \mathbf{t}_2 + R_2\mathbf{t}_0 - \mathbf{t}_0\| \tag{9}$$

$$= \|\mathbf{t}_1 - R_1\mathbf{t}_0 - \mathbf{t}_2 + R_2\mathbf{t}_0\|, \tag{10}$$

and thus $\|\mathbf{t}_1 - \mathbf{t}_2\| = \|\mathbf{t}_1' - \mathbf{t}_2'\|$ only when $R_1 = R_2$ or when $\mathbf{t}_0 = \mathbf{0}$.

Suppose a different scenario, where only the input object Q_{in} is translated by $\mathbf{t}_0$, creating Q_{in}', and Q_{out} stays the same, i.e. $T(Q_{in}) = T'(Q_{in}') = Q_{out}$. Now $T(\mathbf{x}) = T'(\mathbf{x} + \mathbf{t}_0)$ must hold, and it holds when $R = R'$ and $\mathbf{t}' = \mathbf{t} - R\mathbf{t}_0$, the derivation is analogical to Eq. 8. If we now again consider two transformations T_1, T_2 and corresponding T_1', T_2' it is easy to see that $\|\mathbf{t}_1' - \mathbf{t}_2'\| = \|\mathbf{t}_1 - R_1\mathbf{t}_0 - \mathbf{t}_2 + R_2\mathbf{t}_0\|$, which is the same expression as Eq. 10.

In a scenario where only Q_{out} is translated by $\mathbf{t}_0$, yielding Q'_{out}, and Q_{in} stays the same ($T(Q_{in}) = Q_{out}$, $T'(Q_{in}) = Q'_{out}$). Now $T(\mathbf{x}) = T'(\mathbf{x}) - \mathbf{t}_0$ must hold and it is easily proven that it holds when $R = R'$ and $\mathbf{t}' = \mathbf{t} + \mathbf{t}_0$ because then $R'\mathbf{x} + \mathbf{t}' - \mathbf{t}_0 = R\mathbf{x} + \mathbf{t} + \mathbf{t}_0 - \mathbf{t}_0 = R\mathbf{x} + \mathbf{t}$. For two transformations T_1, T_2 and corresponding T'_1, T'_2 we get

$$\|\mathbf{t}'_1 - \mathbf{t}'_2\| = \|\mathbf{t}_1 + \mathbf{t}_0 - \mathbf{t}_2 - \mathbf{t}_0\| = \|\mathbf{t}_1 - \mathbf{t}_2\|. \quad (11)$$

This has profound consequences in context of rigid surface registration (and possibly in other applications as well). When the space of rigid transformations is being sampled and transformations are created by fitting points of Q onto points of P, the value of the Euclidean metric applied on the translation components of the transformations depends on the position of Q (Eq. 10), but does not depend on the position of P (Eq. 11). Having a general rigid transformation $T_1(\mathbf{x}) = R_1\mathbf{x} + \mathbf{t}_1$ and $\mathbf{x} = \mathbf{0}$, $\mathbf{t}_1$ exactly represents the change of position of the point caused by the transformation and if another transformation $T_2(\mathbf{x}) = R_2\mathbf{x} + \mathbf{t}_2$ is defined then $\|\mathbf{t}_1 - \mathbf{t}_2\|$ exactly represents the difference between the position change of T_1 and the position change of T_2. However, if $\mathbf{x}$ changes by a non-zero vector $\mathbf{t}_0$, then the value $\|\mathbf{t}_1 - \mathbf{t}_2\|$ starts to deviate from the difference of the position changes of T_1 and T_2 and, according to Eq. 10, the deviation grows infinitely with the distance of $\mathbf{x}$ from the origin (the length of $\mathbf{t}_0$) and with the difference between the two rotations R_1 and R_2 (the angle between $R_1\mathbf{t}_0$ and $R_2\mathbf{t}_0$). This implies that generally the farther a point is from the origin, the worse the translation component $\mathbf{t}$ of a rigid transformation describes the change of the point's position after applying the transformation, and the worse the Euclidean metric of the translation components describes the difference of these changes between the translation components of two arbitrary rigid transformations.

Using the composed transformation metric, which uses the Euclidean metric on the translation components, is therefore only meaningful when the transformed object is approximately centered at the origin. Otherwise the composed metric can behave unpredictably, having a great negative impact on the registration results.

3. Composed metrics - Dependence on orientation

Consider an input object Q_{in} and an arbitrary rigid transformation T that transforms Q_{in} into an output object $Q_{out} = T(Q_{in})$. Now imagine rotating both Q_{in} and Q_{out} using the same arbitrary rotation matrix R_0 resulting in two objects Q'_{in} and Q'_{out} respectively, with different absolute orientation but the same mutual orientation as Q_{in} and Q_{out}. A transform T' transforms Q'_{in} into Q'_{out}, i.e. $T'(Q'_{in}) = Q'_{out}$. Since rotating Q'_{out} by R_0^T results in Q_{out}, $T(\mathbf{x}) = R_0^T T'(R_0\mathbf{x})$ must hold, which can be expanded as follows:

$$R\mathbf{x} + \mathbf{t} = R_0^T(R'R_0\mathbf{x} + \mathbf{t}') \quad (12)$$

$$R\mathbf{x} + \mathbf{t} = R_0^T R' R_0\mathbf{x} + R_0^T\mathbf{t}'. \quad (13)$$

This equation holds when $R' = R_0 R R_0^T$ and $\mathbf{t}' = R_0\mathbf{t}$.

Suppose we have two different rigid transformations T_1 and T_2 and two corresponding transformations T'_1 and T'_2, where $\mathbf{t}'_1 = R_0\mathbf{t}_1$, $\mathbf{t}'_2 = R_0\mathbf{t}_2$, $R'_1 = R_0R_1R_0^T$ and $R'_2 = R_0R_2R_0^T$, and R_0 is some arbitrary rotation matrix. Obviously, $\|\mathbf{t}_1 - \mathbf{t}_2\| = \|\mathbf{t}'_1 - \mathbf{t}'_2\|$ because $\|\mathbf{t}_1 - \mathbf{t}_2\| = \|R_0\mathbf{t}_1 - R_0\mathbf{t}_2\|$. For $d_R(R_1, R_2) = d_R(R'_1, R'_2)$ it must be that $d_R(R_1, R_2) = d_R(R_0R_1R_0^T, R_0R_2R_0^T)$ which holds when the rotation metric d_R is bi-invariant.

Suppose a different scenario, where only the input object Q_{in} is rotated by R_0, creating Q'_{in}, and Q_{out} stays the same, i.e. $T(Q_{in}) = T'(Q'_{in}) = Q_{out}$. Now $T(\mathbf{x}) = T'(R_0\mathbf{x})$ must hold, which expands as $R\mathbf{x} + \mathbf{t} = R'R_0\mathbf{x} + \mathbf{t}'$. This holds for $R' = RR_0^T$ and $\mathbf{t}' = \mathbf{t}$. If we now again consider two transformations T_1, T_2 and corresponding T'_1, T'_2, obviously $\|\mathbf{t}_1 - \mathbf{t}_2\| = \|\mathbf{t}'_1 - \mathbf{t}'_2\|$. For $d_R(R_1, R_2) = d_R(R'_1, R'_2)$ it must now be that $d_R(R_1, R_2) = d_R(R_1R_0^T, R_2R_0^T)$ which holds when the rotation metric d_R is right-invariant.

Finally, when only Q_{out} is rotated by R_0, creating Q'_{out}, and Q_{in} stays the same ($T(Q_{in}) = Q_{out}$, $T'(Q_{in}) = Q'_{out}$), $T(\mathbf{x}) = R_0^T T'(\mathbf{x})$ must hold, which expands as $R\mathbf{x} + \mathbf{t} = R_0^T(R'\mathbf{x} + \mathbf{t}')$. This implies $R' = R_0R$ and $\mathbf{t}' = R_0\mathbf{t}$ because then $R_0^T(R'\mathbf{x} + \mathbf{t}') = R_0^T(R_0R\mathbf{x} + R_0\mathbf{t}) = R\mathbf{x} + \mathbf{t}$. For two transformations T_1, T_2 and corresponding T'_1, T'_2, for the translation metric we now again get $\|\mathbf{t}_1 - \mathbf{t}_2\| = \|R_0\mathbf{t}_1 - R_0\mathbf{t}_2\|$, which holds. For $d_R(R_1, R_2) = d_R(R'_1, R'_2)$ we need $d_R(R_1, R_2) = d_R(R_0R_1, R_0R_2)$, which holds when the rotation metric d_R is left-invariant.

In context of rigid surface registration, when fitting points of object Q onto points of object P and a composed metric is used to measure distances between the created transformations, the value of the composed metric is independent of the initial orientation of Q only if the rotation metric is right-invariant and independent on the initial orientation of P only if the rotation metric is left-invariant. For a composed metric to be independent on the initial orientation of both Q and P, the rotation metric must be bi-invariant, even when the mutual orientation of Q and P remains unchanged. Therefore we recommend only using rotation metrics that are bi-invariant, otherwise the composed metric might behave unpredictably, negatively impacting the registration results.

4. Proof that d_R^{DEA} is not bi-invariant

Consider general rotation matrices R_1, R_2 and R_0 and the corresponding Euler angles $(\alpha_1, \beta_1, \gamma_1)$, $(\alpha_2, \beta_2, \gamma_2)$ and $(\alpha_0, \beta_0, \gamma_0)$ respectively. If d_R^{DEA} is a bi-invariant rotation metric, then $d_R^{DEA}(R_1, R_2) = d_R^{DEA}(R_1R_0, R_2R_0) = d_R^{DEA}(R_0R_1, R_0R_2)$ must hold for any R_1, R_2, R_0. Now consider that the rotations are such that $\alpha_1 = 0$, $\beta_1 = 0$, $\gamma_1 = 90°$, $\alpha_2 = 0$, $\beta_2 = -90°$, $\gamma_2 = 0$, $\alpha_0 = 0$, $\beta_0 = 30°$, $\gamma_0 = 60°$. It can be easily shown that the Euler angles of the rotation defined by a matrix R_1R_0 are $\alpha_{10} = -30°$, $\beta_{10} = 0$, $\gamma_{10} = 150°$ and those of R_2R_0 are $\alpha_{20} = 0$, $\beta_{20} = -60°$, $\gamma_{20} = 60°$. Similarly, for R_0R_1 we get $\alpha_{01} = 0$, $\beta_{01} = 30°$, $\gamma_{01} = 150°$ and for R_0R_2 we get $\alpha_{02} = 73.896°$, $\beta_{02} = -25.658°$, $\gamma_{02} = 56.309°$. It is not hard to show now that $d_R^{DEA}(R_1, R_2) = 127.279$, $d_R^{DEA}(R_1R_0, R_2R_0) = 112.249$ and $d_R^{DEA}(R_0R_1, R_0R_2) = 119.404$. This implies that in general $d_R^{DEA}(R_1, R_2) \neq d_R^{DEA}(R_1R_0, R_2R_0)$ and $d_R^{DEA}(R_1, R_2) \neq d_R^{DEA}(R_0R_1, R_0R_2)$ thus proving that the rotation metric d_R^{DEA} is neither left nor right-invariant, and therefore not bi-invariant.

5. Results of Super4PCS

Table 1 shows the errors of the latest (2019/04/08) implementation of the Super4PCS registration method with different values of δ

δ'	**Arm**	Bir	Bub	**Bud**	**Coa**	**Dra**	**Egg**	Tee	Hea	**Hip**	Kac	**Osc**	**Tes**	Suz
0.010	1.257	1.941	1.306	1.594	1.387	1.581	1.361	1.416	1.704	1.366	1.801	1.075	1.816	1.366
0.011	0.220	1.941	1.303	1.588	1.383	1.456	1.348	1.396	1.712	1.382	1.834	0.543	1.822	1.376
0.013	1.214	1.941	1.318	1.598	1.387	1.457	1.363	1.380	1.705	1.353	1.826	1.073	1.402	1.341
0.015	1.242	1.941	1.622	1.776	1.245	1.974	1.346	1.383	1.703	1.417	1.841	1.073	1.814	1.596
0.017	**0.099**	1.855	1.296	1.590	1.385	0.418	1.380	1.411	1.710	1.357	1.828	2.016	1.824	1.550
0.020	0.334	0.890	1.464	1.592	1.389	**0.066**	0.166	1.405	1.696	1.048	1.761	1.073	0.391	2.103
0.023	**0.134**	1.319	2.262	0.373	0.225	0.587	1.363	1.398	1.718	1.340	1.852	0.151	1.327	2.099
0.026	0.232	1.577	1.464	0.800	1.380	0.197	1.151	1.397	1.926	1.357	1.844	**0.081**	1.413	1.562
0.030	**0.131**	1.026	1.432	0.273	1.380	2.068	1.295	1.399	1.915	1.096	1.783	**0.067**	0.207	2.039
0.035	**0.080**	1.742	1.454	1.589	1.380	**0.090**	1.568	1.422	1.780	0.443	1.779	**0.146**	0.162	1.752
0.040	0.285	1.277	1.495	0.280	1.378	**0.039**	1.909	1.442	1.671	**0.106**	1.825	**0.135**	**0.122**	1.825
0.046	**0.121**	1.318	1.397	0.209	0.258	**0.060**	0.769	1.443	1.915	**0.094**	1.930	0.161	1.974	1.822
0.053	**0.047**	1.729	1.485	**0.091**	0.484	**0.136**	0.557	1.431	1.964	0.350	1.848	**0.037**	1.275	1.876
0.061	**0.070**	1.411	1.307	**0.064**	**0.120**	**0.065**	1.606	1.414	1.997	0.153	1.721	**0.035**	**0.143**	1.888
0.070	**0.047**	1.421	1.482	0.178	**0.065**	**0.035**	0.207	1.410	1.755	**0.083**	1.593	**0.044**	0.237	1.877
0.080	**0.079**	1.570	1.360	**0.071**	**0.134**	**0.102**	0.383	1.420	1.554	1.685	1.858	**0.079**	**0.105**	1.771
0.092	**0.045**	1.028	1.525	**0.087**	**0.051**	**0.059**	0.519	1.471	1.638	0.168	1.452	**0.105**	1.163	2.003
0.106	**0.071**	1.481	1.548	**0.052**	**0.122**	**0.086**	0.490	1.472	1.946	**0.055**	1.474	**0.074**	**0.040**	2.030
0.121	**0.067**	0.531	1.345	**0.134**	**0.104**	**0.084**	**0.121**	0.467	1.623	**0.117**	1.758	0.165	**0.076**	1.979
0.139	**0.082**	1.558	1.467	**0.075**	**0.164**	**0.126**	0.157	1.558	1.994	0.227	1.237	**0.094**	**0.086**	1.837
0.160	**0.103**	1.473	1.472	**0.062**	**0.048**	**0.069**	0.528	1.503	1.666	0.332	1.639	**0.109**	**0.072**	1.781
0.184	**0.102**	1.410	1.521	**0.073**	**0.071**	**0.128**	0.548	1.585	1.957	**0.140**	1.857	**0.149**	**0.117**	1.723
0.211	**0.119**	1.475	1.505	0.161	**0.141**	**0.136**	0.314	1.500	1.612	0.251	1.639	0.171	**0.148**	1.779
0.243	**0.065**	1.352	1.521	**0.118**	**0.038**	0.178	0.383	0.468	1.947	0.174	1.621	0.260	**0.148**	1.769
0.279	0.162	1.358	1.536	0.152	0.178	0.160	0.401	1.797	1.750	0.201	1.764	**0.147**	**0.076**	1.909
0.320	**0.090**	1.430	1.518	**0.117**	**0.120**	1.228	0.386	1.731	1.907	0.309	1.843	**0.127**	2.002	1.835
0.368	0.256	1.250	1.533	0.183	**0.117**	0.202	0.442	1.936	1.920	0.223	1.898	0.382	0.342	1.767
0.422	1.149	1.431	1.620	1.938	**0.099**	1.412	0.502	2.003	1.895	0.130	1.509	0.603	1.349	1.836
0.485	0.354	1.222	1.622	1.964	0.371	1.610	0.462	2.028	1.785	0.427	1.382	0.244	1.387	1.826
0.557	1.676	1.464	1.628	0.372	0.185	**0.115**	0.676	2.358	2.043	0.391	1.800	0.611	1.385	1.789
0.640	2.128	1.365	1.533	0.446	**0.115**	0.313	0.746	2.422	1.837	0.328	1.405	0.652	1.360	1.621
0.735	2.017	1.497	1.334	0.557	0.259	0.333	1.901	0.329	1.690	0.283	1.463	0.491	1.529	1.748
0.844	1.462	1.278	1.552	1.187	0.426	2.011	1.941	2.626	2.069	1.747	1.428	0.643	1.365	2.560
0.970	1.161	1.284	0.834	0.869	0.303	0.819	1.929	3.507	1.779	1.680	2.208	0.720	1.800	1.514
1.114	2.713	2.683	1.575	1.917	0.554	2.168	0.927	4.015	1.934	1.435	1.944	0.602	1.675	1.688

Table 1: *Errors of Super4PCS with different values of* $\delta = \delta' r(Q)$ *for all the datasets.*

for all the datasets. The value of delta is set relative to the radius of Q as $\delta = \delta' r(Q)$. The cells with error $\leq \psi$ are marked bold, in all other cases the ragistration was considered a failure. Names of the datasets for which the registration was succesful for at least one value of δ' are also marked bold, for all the other datasets the registration failed for all the values.

The largest number of successful registrations (7) was achieved using $\delta' = 0.106$, 0.121 and 0.184. Since there is quite a large gap between 0.121 and 0.184, we select $\delta' = 0.11$ as approximately optimal because it is close to both 0.121 and 0.106. Table 2 shows comparison of Super4PCS with $\delta' = 0.11$ to the model RANSAC algorithm where d^{TSS} was used as the metric with $c = 9.52$. For each dataset, ICP alignment was used after the global registration with the same δ setting as Super4PCS ($\delta' = 0.11$). We only show binary evaluation of whether the final registration was successful instead of showing the error, since in the case of successful alignment the final error is dictated by the ICP rather than by the global registration, and in the case of incorrect result the error value is irrelevant. The running times do not include ICP and the measurements were done on a computer with CPU *Intel Core i7-4770* (clock rate 3.4 GHz, 4 cores, L1 cache 256 kB, L2 cache 1 MB, L3 cache 8 MB) and 16 GB of memory with clock rate of 1.6 GHz with *Windows 10 64-bit* operating system.

If the global registration resulted in a good alignment, then the subsequent ICP further strongly increased its precision for most datasets. The only exception was the *Bir* dataset, where the model registration algorithm actually found a very good alignment, but the ICP made it much worse, resulting in an unsuccessful registration. Super4PCS for this dataset resulted in a bad alignment even without ICP. In total, including ICP did not lead to an improvement of the

Dataset	Success		Time [ms]	
	Model	S4PCS	Model	S4PCS
Arm	yes	yes	2406	9807
Bir	no	no	820	298
Bub	yes	no	1337	596
Bud	yes	yes	2987	7203
Coa	yes	yes	1958	701
Dra	yes	yes	3071	9084
Egg	yes	no	20654	19592
Hea	yes	no	1369	9693
Hip	yes	yes	1495	1372
Kac	yes	no	1496	635
Osc	yes	yes	5018	4182
Suz	yes	no	6641	5525
Tee	yes	no	2361	1432
Tes	yes	yes	10926	10382

Table 2: *Comparison of the model RANSAC registration algorithm using d^{TSS} with $c = 9.52$ to Super4PCS with $\delta' = 0.11$, after the global registration ICP was performed also with $\delta' = 0.11$, the running times do not include ICP.*

registration using Super4PCS, which was again successful with 7 datasets.