Scale Invariant Robust Registration of 3D-Point Data and a Triangle Mesh by Global Optimization

Onay Urfalıoḡlu, Patrick Mikulastik, and Ivo Stegmann

Information Technology Laboratory (LFI), University of Hannover

Abstract. A robust registration of 3D-point data and a triangle mesh of the corresponding 3D-structure is presented, where the acquired 3D-point data may be noisy, may include outliers and may have wrong scale. Furthermore, in this approach it is not required to have a good initial match so the 3D-point cloud and the according triangle mesh may be loosely positioned in space. An additional advantage is that no correspondences have to exist between the 3D-points and the triangle mesh. The problem is solved utilizing a robust cost function in combination with an evolutionary global optimizer as shown in synthetic and real data experiments.

Keywords: scale invariant, robust registration, evolutionary optimization, 3D-transformation.

1 Introduction

Registration is applied in object recognition, 3D-geometry processing and acquisition. Often two observations of a 3D-scene are provided, where one is called the model and the other is called the data. Each of them is defined in its own coordinate system. The process of registration is to find a transformation which best fits the data to its corresponding model. The generation of 3D-point cloud data in the process of monocular camera parameter estimation cannot guarantee a correct scale with respect to a 3D-model, unless the scale is previously determined and applied.

In this paper robust registration of a 3D-point cloud with respect to a provided 3D-model, e.g. a polygon set is addressed. The most common methods for this purpose are based on the Iterated Closest Point (ICP) algorithm [1, 2, 3, 4]. Scale invariant versions of ICP-based methods were realized by [5]. However, the convergence of ICP-based methods to the global optimum requires the data set to be sufficiently prealigned with respect to the corresponding 3D-model. To overcome this problem, another approach was proposed [6] utilizing global optimization in the pose space to find the transformation which best aligns the provided range images. But this approach does not enable the estimation of the scale factor.

The approach proposed in this paper enables automatic registration where no correspondences between the 3D-point cloud and the 3D-model are required.

© Springer-Verlag Berlin Heidelberg 2006

Similar to the approach in [6], the 3D-point cloud may include outliers and may be freely initially positioned relative to the corresponding 3D-model. However, in contrast to [6], the 3D-point cloud may also have a wrong scale. A modified version of the robust cost function [7] is utilized in combination with an efficient global optimization method called *Differential Evolution* (DE) [8] in order to estimate the correct transformation of the 3D-point cloud.

The paper is organized as follows. In section 2, the modified robust cost function is presented. In section 3, the utilized evolutionay global optimization is described. In the following section 4 experimental results are presented and in the last section 5 the paper is concluded.

2 Robust Cost Function

The proposed cost function is based on the distances of the 3D-points to the triangle mesh. In order to determine the distance of a 3D-point to the triangle mesh it is required to find the closest triangle first. This is accomplished by calculating the distances to all planes defined by each triangle, respectively. Figure 1 shows a case where the 3D-point $\boldsymbol{P}$ has a distance d_0 to the selected

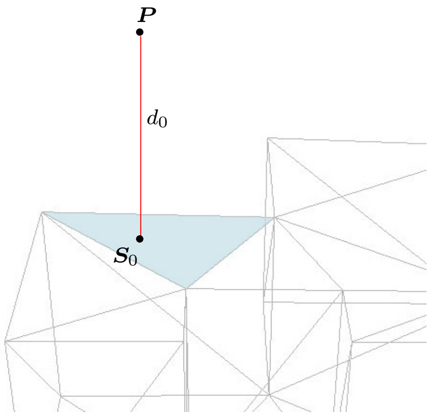

Fig. 1. Distance d_0 to plane defined by the triangle

triangle. In this case it is sufficient to calculate the distance to the plane. Only if the intersection point $\boldsymbol{S_0}$ in the plane is within the triangle the calculated distance is valid. This is checked by the calculation of the so called Barycentric coordinates. Figure 2 shows a triangle and a Point $\boldsymbol{S_0}$, where its corresponding Barycentric coordinates are $(\alpha_1, \alpha_2, \alpha_3)$ and $\boldsymbol{W}_i$ are the vectors pointing to the vertices of the triangle

$$\boldsymbol{S_0} = \alpha_1 \boldsymbol{W}_1 + \alpha_2 \boldsymbol{W}_2 + \alpha_3 \boldsymbol{W}_3. \tag{1}$$

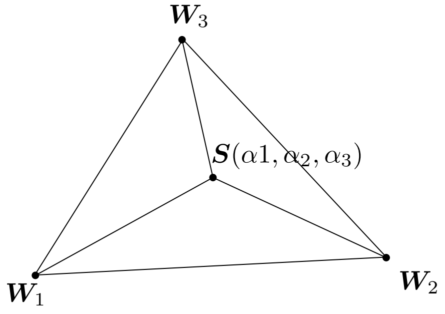

Fig. 2. Barycentric coordinates used to test whether a point S_0 is within the triangle

The point S_0 is within the triangle if and only if

$$\alpha_1, \alpha_2, \alpha_3 \in [0, 1] \ \wedge \ \alpha_1 + \alpha_2 + \alpha_3 = 1. \tag{2}$$

In contrast, figure 3 shows a case where the intersection point S_0 is not within the triangle. Therefore, in this case all 3 distances d_1, d_2, d_3 to the lines defined

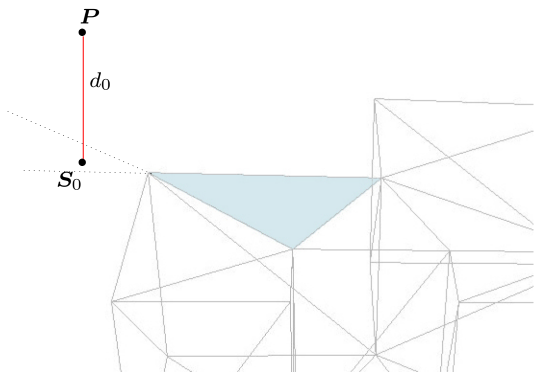

Fig. 3. Case where the intersection Point S_0 is not within the triangle

by the edges of the triangle have to be determined additionally, as shown in figure 4.

But even the determination of these distances proves to be insufficient, since the closest point on a line is not necessarily within the triangle, so three additional distances d_4, d_5, d_6 to the vertices have to be calculated, as shown in figure 5. Finally, the smallest distance d having the corresponding intersection point S_i out of all calculated distances $d_i, \ i \in [0, 6]$ is chosen.

The distances depend on the orientation and the position of the 3D-point cloud. The task is to estimate the transformation consisting of the orientation,

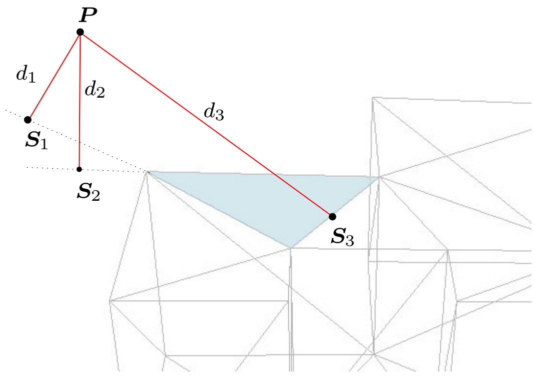

Fig. 4. Distances d_1, d_2, d_3 to lines defined by the edges of the triangle

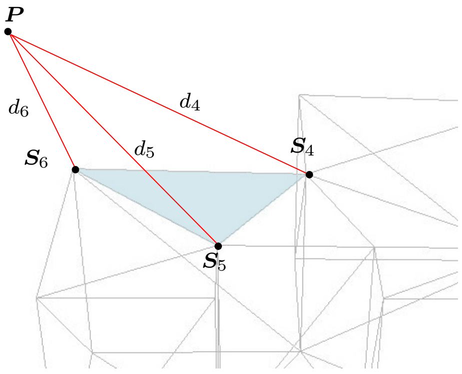

Fig. 5. Distances d_4, d_5, d_6 to vertices of the triangle

the translation and the overall scale factor of the 3D-point cloud which leads to a best match with the triangle mesh. The rotation matrix $\boldsymbol{R}$ is parameterized by the angles $\phi, \theta, \rho \in [0, \pi]$

$$\boldsymbol{R}(\phi, \theta, \rho). \tag{3}$$

The translation vector $\boldsymbol{t}$ is denoted

$$\boldsymbol{t} = (\delta x, \delta y, \delta z). \tag{4}$$

The overall scale factor is denoted s. The parameter vector $\boldsymbol{x}$ is defined by

$$\boldsymbol{x} = (\phi, \theta, \rho, \delta x, \delta y, \delta z, s)^\top \tag{5}$$

and contains all 7 parameters to be estimated. A 3D-point $\boldsymbol{P}$ is then transformed to $\boldsymbol{P}'$ by the transformation $\boldsymbol{T}$

$$\boldsymbol{P}' = \boldsymbol{T}(\boldsymbol{x})\boldsymbol{P} = s\,(\boldsymbol{R}(\phi, \theta, \rho)\boldsymbol{P} + \boldsymbol{t}). \tag{6}$$

Therefore, each distance d_j of the point $\boldsymbol{P}_j$ depends on the parameter vector and is determined by

$$d_j = d_j(\boldsymbol{x}) = \left|\boldsymbol{T}(\boldsymbol{x})\boldsymbol{P}_j - \boldsymbol{S}\right| . \tag{7}$$

Since the 3D-point cloud consisting of N points may include outlier points, a robust cost function $\Gamma(\boldsymbol{x})$ is utilized, which is a modified version of the cost function defined in [7]. Assuming a Gaussian error with zero mean and a variance of σ for the position of the point $\boldsymbol{P}$, the utilized cost function is defined by

$$\Gamma(\boldsymbol{x}) = \sum_{j=1}^{N} - \exp\left(-\frac{d_j(\boldsymbol{x})^2}{2\kappa s^3}\right) , \tag{8}$$

where the parameter κ depends mainly on σ. As a rule of thumb, good estimates are obtained for

$$\kappa \approx 10\sigma^2. \tag{9}$$

This cost function has to be minimized in order to find the best solution vector $\boldsymbol{x}$.

Generally, this robust cost function results in local minima in the search space where the *global* minimum has to be found. The number of local minima increases with the number of outliers. The scale as an additional degree of freedom leads to even more complicated search spaces. Furthermore, the initial orientation and the position of the 3D-point cloud may differ considerably with respect to the triangle mesh, so it is appropriate to use a global optimization method, which is described in the next section.

The modification of the robust const function is realized by the multiplicative term s^3 in the denominator of (8) which enforces an additional weighting of the distance depending on the scale. The reason for this modification is that there is always an attraction towards smaller scale factors since, in the average, this leads to smaller distances. This property is (over) compensated by the introduction of this term in order to enhance the global search in greater scale factor regions.

3 Differential Evolution Optimization

The utilized evolutionary optimizer is the so called *Differential Evolution* (DE) method [8, 9, 10]. It is known as an efficient global optimization method for continuous problem spaces with many applications. The optimization is based on a population of $n = 1, ..., M$ solution candidates $\boldsymbol{x}_{n,i}$ at iteration i where each candidate has a position in the 7-dimensional search space. Initially, the solution candidates are randomly generated whithin the provided intervals of the search space. The population improves by generating new positions iteratively for each candidate. New positions for the iteration step $i + 1$ are determined by

$$\boldsymbol{y}_{n,i+1} = \boldsymbol{x}_{k,i} + F \cdot (\boldsymbol{x}_{l,i} - \boldsymbol{x}_{m,i}) \tag{10}$$
$$\boldsymbol{x}_{n,i+1} = C\left(\boldsymbol{x}_{n,i},\ \boldsymbol{y}_{n,i+1}\right) , \tag{11}$$

where k, l, m are random integers from interval $[1, M]$, F is a weighting scalar, $\boldsymbol{y}_{n,i+1}$ a displaced $\boldsymbol{x}_{k,i}$ by a weighted difference vector and $C()$ is a crossover operator copying coordinates from both $\boldsymbol{x}_{n,i}$ and $\boldsymbol{y}_{n,i+1}$ in order to create $\boldsymbol{x}_{n,i+1}$. The crossover operator C is provided with a value specifying the probability to copy coordinates either from $\boldsymbol{x}_{n,i}$ or $\boldsymbol{y}_{n,i+1}$ to $\boldsymbol{x}_{n,i+1}$. Only if the new candidate $\boldsymbol{x}_{n,i+1}$ proves to have a lower cost it replaces $\boldsymbol{x}_{n,i}$, otherwise it is discarded.

DE includes an adaptive range scaling for the generation of solution candidates through the difference term in (10). This enables global search in the case where the solution candidate vectors are spread in the search space and the mean difference vector is relatively large. In the case of a converging population the mean difference vector becomes relatively small and this enables efficient fine tuning at the end phase of the optimization process.

4 Experimental Results

In order to test the proposed approach both synthetic and real data based experiments are performed.

4.1 Synthetic Data

As shown in fig. 6, 4 boxes represented by their wireframes are placed in 3D-space. The corresponding 3D-point cloud consists of the corner points of the boxes with no position error. Additionally, the 3D-point cloud includes 4 outlier points indicated by surrounding squares. For all points $\boldsymbol{P}_j$, it is

$$\boldsymbol{P}_j \in [-6, 12]u \times [0, 21]u \times [0, 41]u, \tag{12}$$

where u is any arbitrary length unit. By transforming and rescaling the coordinates of the 3D-point cloud a second 3D-point cloud is generated and the robust registration is performed utilizing the second point cloud. The 3D-point cloud is translated by $(0\ 5\ 5)^\top u$ and rotated by $\phi = 0.4$ within the x-y-plane. 3 different global scales $s = 0.5, 1, 2$ are applied. The search interval for the translation coordinates is set to $[-20, 20]$, whereas the search interval or the scale factor is set to $[0.5, 2]$. The population size of the DE-algorithm is set to 60.

Fig. 7 shows the boxes and the corresponding transformed 3D-point cloud with a scale factor of $s = 0.5$.

Fig. 8 and 9 show the same configuration with a scale factor of $s = 1$ and $s = 2$, respectively.

In all experiments with synthetic data our algorithm found the global minimum of the cost function, so that the correct transformation and scale parameters could be recovered, as shown in fig. 6.

4.2 Real Data

In order to test the proposed approach in real data based scenarios, two real world 3D-objects are modelled by hand and with a 3D scanner to create the

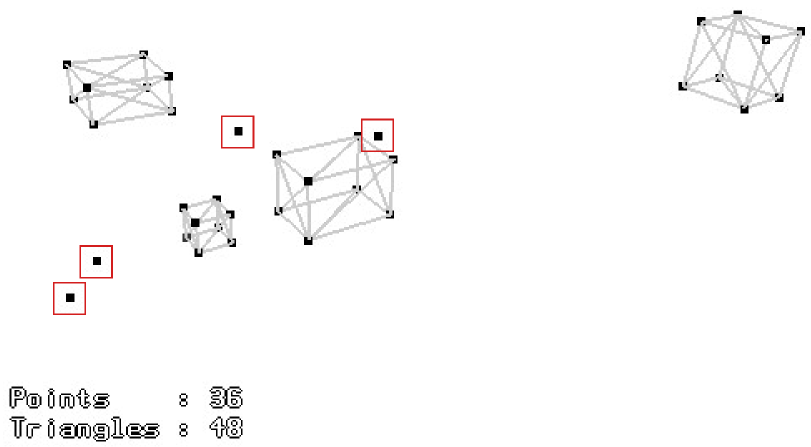

Fig. 6. Result of the robust registration

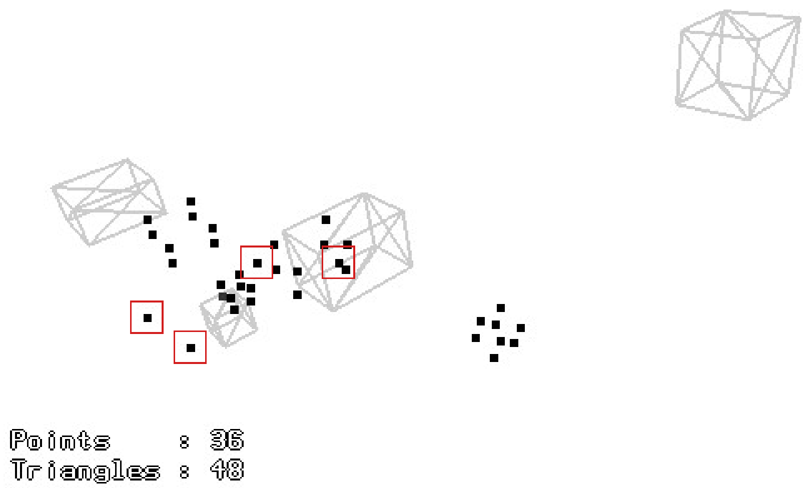

Fig. 7. Initial configuration of 3d-point cloud and corresponding triangle mesh at $s = 0.5$

corresponding triangle meshes. Fig.10 shows the 3D-objects named 'block' and 'dino', respectively. In the following experiments, the search interval for the coordinates of the translation vector is set to $[-300, 300]$. The search interval for the scale factor is set to $[0.5, 2]$. The corresponding 3D-point cloud is generated by structure-from-motion utilizing a monocular camera. In both cases, the average error variance of the 3D-points of the resulting 3D-clouds is $\sigma^2 \approx 2.25u^2$, assuming a Gaussian probability density for the position error. Figure 11 shows the initial configuration of the test object 'block'.

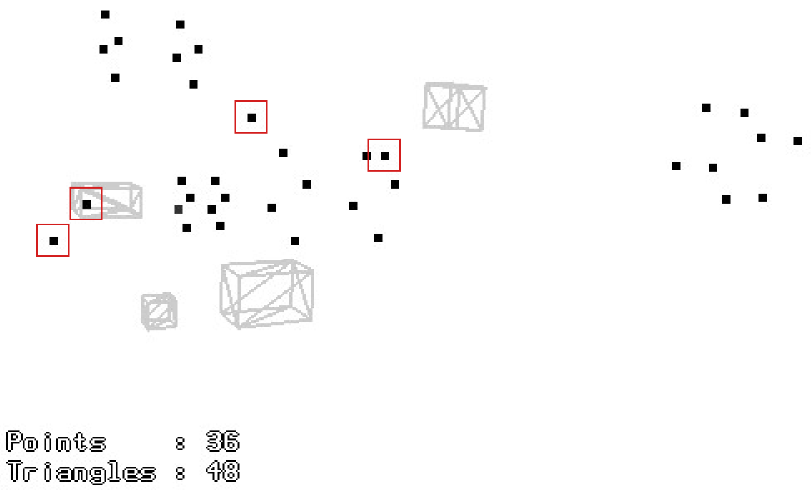

Fig. 8. Initial configuration of 3d-point cloud and corresponding triangle mesh at $s = 1$

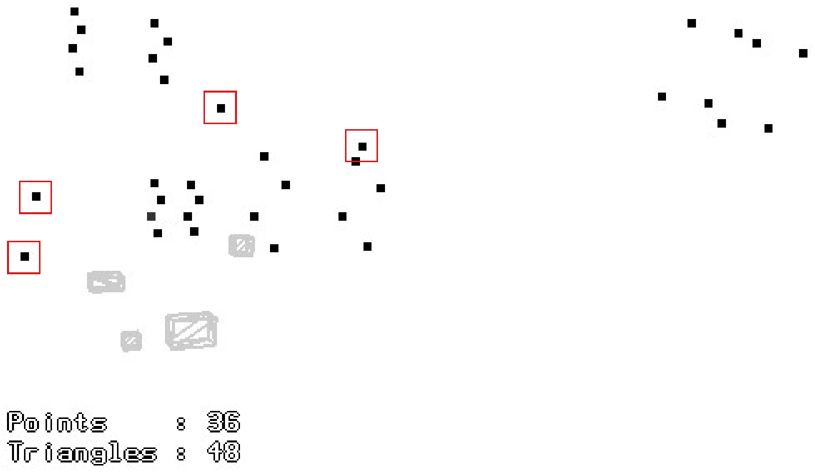

Fig. 9. Initial configuration of 3d-point cloud and corresponding triangle mesh at $s = 2$

The corresponding 3D-point cloud consists of $N = 1103$ points $\boldsymbol{P}_j$ where

$$\boldsymbol{P}_j \in [-125, 120]u \times [-132, 105]u \times [-25, 80]u. \qquad (13)$$

The corresponding triangle mesh is composed of 56 triangles. Figure 12 shows the result of the robust registration, achieved after 500 iterations at a population size of 60. The estimated required scale factor is $s = 0.514$, the number of outliers

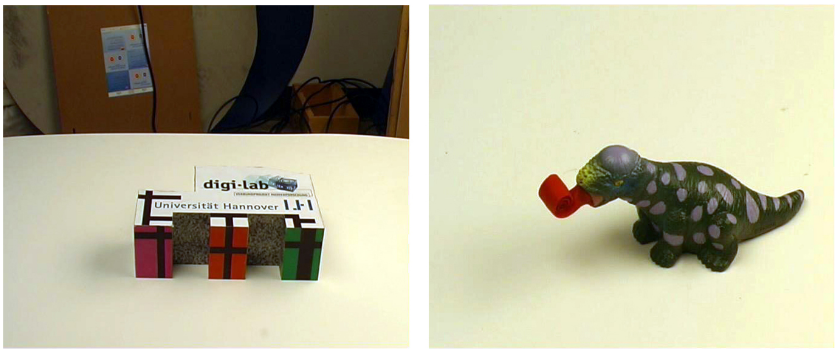

Fig. 10. 3D-objects used for real data experiments

is 214 at an inlier threshold of $\sigma_i = 3u$. The std. deviation of the inlier-error is $\sigma_e = 1.219u$.

In the second real data test for the proposed approach, the initial configuration of the 3D-point cloud with respect to the corresponding triangle mesh is shown in figure 13. The point cloud consists of $N = 904$ points $\boldsymbol{P}_j$ where

$$\boldsymbol{P}_j \in [-125, 120]u \times [-132, 105]u \times [-25, 80]u. \tag{14}$$

The corresponding triangle mesh is composed of 2000 triangles. The result of the robust registration is shown in figure 14, achieved after 900 iterations at

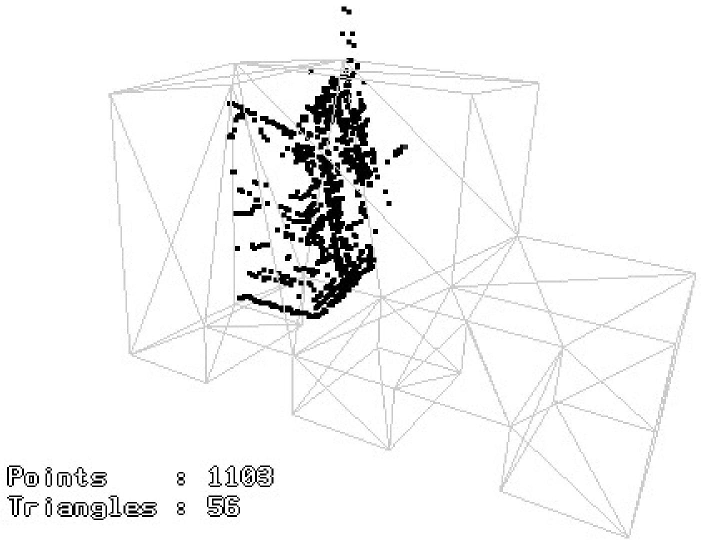

Fig. 11. 'block': Initial configuration of 3D-point cloud and the corresponding triangle mesh

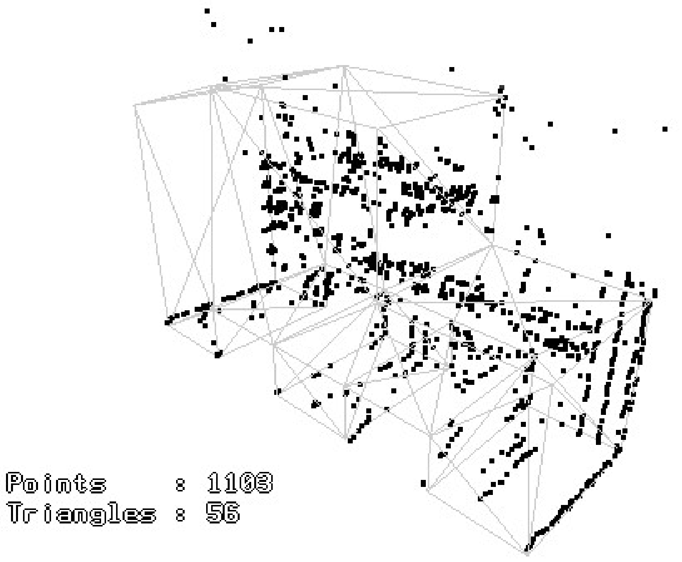

Fig. 12. 'block': Result of the registration of the 3D-point cloud

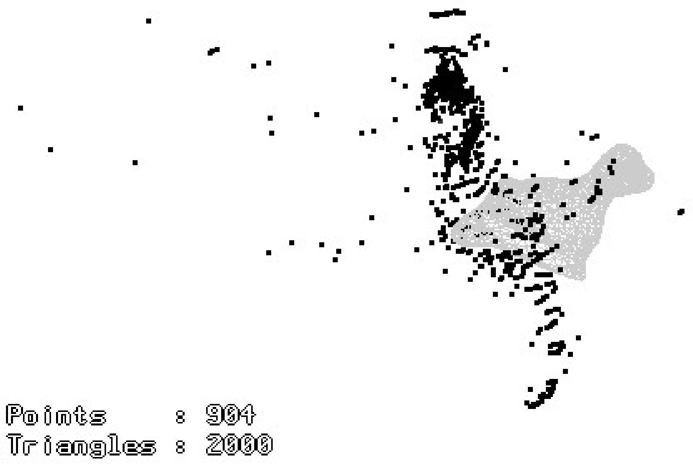

Fig. 13. 'dino': Initial configuration of 3D-point cloud and the corresponding triangle mesh

a population size of 90. The estimated required scale factor is $s = 1.874$, the number of outliers is 52 at an inlier threshold of $\sigma_i = 3u$. The std. deviation of the inlier-error is $\sigma_e = 0.687u$.

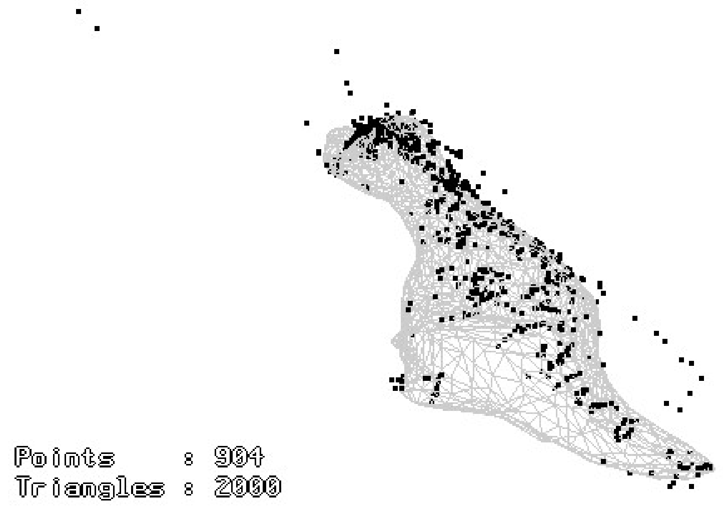

Fig. 14. 'dino': Result of the registration of the 3D-point cloud

5 Conclusions

The proposed approach enables the robust registration of two observations of a 3D-object, where one is represented as a triangle mesh and the other is a 3D-point cloud, in different coordinate systems at different scales. With this approach the automatic and accurate estimation of the transformation parameters from one system to the other is made possible, even in cases of high outlier amounts and lack of prealignment of the two coordinate systems. The additional scale parameter proved to further complicate the estimation, so the utilized cost function had to be adapted accordingly. This approach is applicable in cases where the registration has to be done automatically without manual prealignment and where the scale factors of the two coordinate systems may be different.

References

1. Besl, P., McKay, N.: A method for registering of 3-d shapes. In: PAMI. Volume 14. (1992) 239–256
2. Chen, C.S., Hung, Y.P., Cheung, J.B.: Ransac-based darces: a new approach to fast automatic registration of partially overlapping range images. In: IEEE Transactions on Pattern Analysis and Machine Intelligence. Volume 21. (1999) 1229–1234
3. Masuda, T., Yokoya, N.: A robust method for registration and segmentation of multiple range images. Computer Vision and Image Understanding **61** (1995) 295–307

4. Sharp, G.C., Lee, S.W., Wehe, D.K.: Invariant features and the registration of rigid bodies. In: IEEE International Conference on Robotics & Automation. (1999) 932–937
5. Burschka, D., Li, M., Taylor, R., Hager, G.D.: Scale-invariant registration of monocular stereo images to 3d surface models. In: ICIRS, Sendai, Japan (2004) 2581–2586
6. Silva, L., Bellon, O.R.P., Boyer, K.L.: Robust multiview range image registration. In: Proc. of the XVI Brazilian Symposium on Computer Graphics and Image Processing. (2003)
7. Urfalıoḡlu, O.: Robust estimation with non linear particle swarm optimization. In: Proceedings of Mirage 2005 (Computer Vision / Computer Graphics Collaboration Techniques and Applications), INRIA Rocquencourt, France (2005)
8. Storn, R., Price, K.: Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, ICSI (1995)
9. Storn, R., Price, K.: Minimizing the real functions of the icec'96 contest by differential evolution. In: IEEE International Conference on Evolutionary Computation, Nagoya (1996) 842–844
10. Price, K.V.: Differential evolution: a fast and simple numerical optimizer. In: Biennial Conference of the North American Fuzzy Information Processing Society, NAFIPS, IEEE Press, New York. ISBN: 0-7803-3225-3 (1996) 524–527