# Matching and Pose Estimation of Noisy, Partial and Planar B-Rep Models

Maximilian Sand
University of Bayreuth
Universitätsstraße 30
Bayreuth, Germany 95440
maximilian.sand@uni-bayreuth.de

Dominik Henrich
University of Bayreuth
Universitätsstraße 30
Bayreuth, Germany 95440
dominik.henrich@uni-bayreuth.de

## ABSTRACT

Three dimensional models are represented in various ways. One possibility are boundary representation (B-Rep) models, which contain geometric and topological information. This makes B-Reps suitable for tasks that need an explicit algebraic representation of the surface, e.g. numerical optimization or simulation. Reconstructing a B-Rep model of an object or the environment often is a tedious task needing much manual intervention. An intuitive way is to use a hand-held depth camera and perform a real-time reconstruction. In the domain of robotics, we previously presented a system [18] that is able to incrementally reconstruct a planar B-Rep model from a stream of organized point clouds acquired from a robot mounted camera. Since the acquisition poses are known, this approach can not directly be used in a setup with a hand-held camera.

The contribution of this work is a new method for matching planar B-Rep models and estimating their relative pose. In particular, the input models can be noisy and incomplete containing very few geometric features like corners, which is likely to occur when a model from only one viewpoint is processed. In combination with our previous work, we show that our approach can be used to build a simultaneous location and mapping (SLAM) system for an easy reconstruction of planar B-Rep models.

## CCS CONCEPTS

• **Computing methodologies → Volumetric models**; *Perception*; 3D imaging;

## KEYWORDS

Registration, Pose Estimation, Boundary Representation Model, Shape Matching

**ACM Reference format:**
Maximilian Sand and Dominik Henrich. 2017. Matching and Pose Estimation of Noisy, Partial and Planar B-Rep Models. In *Proceedings of CGI '17, Yokohama, Japan, June 27-30, 2017,* 6 pages.
https://doi.org/10.1145/3095140.3095170

*CGI '17, June 27-30, 2017, Yokohama, Japan*

ACM ISBN 978-1-4503-5228-4/17/06…$15.00
https://doi.org/10.1145/3095140.3095170

## 1 INTRODUCTION

Representing the environment as a 3D model plays an important role in many domains. Depending on the demands of applications, such models are represented at different levels of abstraction ranging from low-level sensor readings as point clouds to high-level models including geometry, topology, physical properties or semantics. One type of representation are boundary representation (B-Rep) models that include both geometric and topological properties. Depending on the degree and complexity of the face representations, B-Rep models can range from polyhedra up to volumes bounded by freeform surfaces. In contrast to other surface models like triangle meshes, B-Reps contain an explicit geometric representation of faces and edges and their neighborhood relations. This makes B-Reps suitable for tasks that need an explicit algebraic representation of the surface, e.g. numerical optimization or simulation. In a previous work [18], we presented a system that is able to incrementally reconstruct a planar B-Rep model from a stream of organized point clouds. In contrast to other reconstruction approaches, we instantly reconstruct a B-Rep model from each viewpoint and incrementally merge this partial B-Rep into the reconstructed B-Reps of the previous frames. This procedure can be performed online, which means that a valid B-Rep model is available at each point in time during camera movement. The more images are incorporated the more complete the model becomes. This is advantageous for example in the field of robotics, where intermediate models can be used to plan grasp poses when the robot approaches a workpiece.

The drawback of this approach is that acquisition poses have to be known, for example in a setup where the camera is mounted to a robot arm [18]. In other applications like object reconstruction or mapping, it would be useful to reconstruct B-Rep models with hand-held depth cameras. Therefore, the camera poses have to be estimated from the images. In this paper, we present an approach for estimating the relative pose of two partial, planar B-Rep models that contain noise. In combination with [18], this yields a complete SLAM (simultaneous location and mapping) system that directly reconstructs planar B-Rep models with hand-held cameras.

A similar result can be reached by using a SLAM system for point clouds (e.g. [9, 22]), merging all clouds to a global cloud and applying reverse engineering methods [1, 2, 10, 21]. Despite the advantage of higher accuracy, since all data is available, this approach has significant drawbacks: For large scenes, point cloud data might not fit into memory and processing times of the reverse engineering methods prevent an online application.

Methods based on signed distance functions [6, 11, 20, 26, 27] can produce large triangle meshes. Internally a voxel grid with distance functions is used from which a mesh can be extracted with the

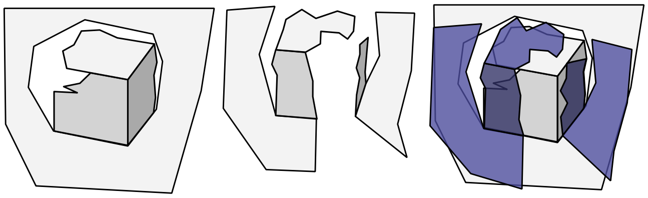


**Figure 1: Two partial B-Reps of a box on a plane with 4 faces, 5 edges and 2 corners (left) and 5 faces, 3 edges and 0 corners (middle). The incompleteness can originate e.g. due to the viewpoint or occlusions. The estimated pose aligns both models (right).**

marching cubes algorithm. In contrast to these methods, matching and merging B-Rep models has the advantage that the surface is continuously and efficiently represented without the need of voxel discretization. Additionally, intermediate results during acquisition are also valid B-Rep models that can be used for e.g. planning algorithms in robotics.

The pose estimation procedure from B-Reps presented in this paper can also be used for localization problems: If a complete model of an object is given (e.g. from CAD data), the pose of a hand-held camera observing the real object can be estimated by reconstructing a single partial B-Rep from the point cloud of the camera and matching it to the given model.

In this paper, we contribute an approach for matching two incomplete, planar B-Rep models and computing their relative pose. Our method can be performed online and can handle partially overlapping models. Faces that are available in both B-Reps need not have common edges and may overlap. In combination with our previous work [18], this can be used to build a complete SLAM system for incrementally reconstructing planar B-Rep models from organized point clouds of a hand-held camera.

## 2 RELATED WORK

Various approaches for aligning 3D models have been proposed that can be distinguished by the data representation. For aligning raw point cloud data, the well-known ICP algorithm [3] is often employed using all point data. If color is also considered, methods based on photoconsistency [12, 13, 22] can be used. Sparse methods [7, 9] reduce the amount of data to match by extracting feature points. All these methods do not consider geometric properties. Approaches that extract planes for pose estimation [14, 17, 23] head for this direction, but do not consider topological information.

Algorithms for matching of 3D surface models like triangle meshes can utilize topology information. There is a great number of methods for matching 3D surface models, mainly in the field of 3D model retrieval. An extensive survey of these methods can be found in [24]. A direct application to our problem is prevented by the fact, that all global feature-based methods do not support partial matching, since they describe the object as a whole. Local methods overcome this problem, but are designed for triangle meshes and thus can not utilize the topological and geometric information of a B-Rep.

Graph-based methods utilize the topology graph of solid models like B-Reps. Faces are represented as graph nodes and boundary edges between faces are graph edges. Using graph comparison [8, 16], a matching can be performed. An application to our use case is not possible, because partial B-Reps reconstructed from one image can contain a different topology graph due to occlusions or noise (see Fig. 1).

A group of methods that support partial matching are approaches with mixed input representations: Single point clouds can be aligned to complete CAD models [5], or CAD models are found in large laser scans [4].

In summary, we see that the main challenge is the incompleteness of the models, which leads to varying topology graphs and the necessity for partial matching. Nevertheless, both models, we want to align, are B-Rep models which contain geometric and topological information we want to employ as much as possible.

The contribution of this paper is a fast matching procedure for planar B-Rep models which suffer from a high degree of incompleteness due to occlusions or noise. Therefore, our method does not rely on topology graphs for matching, but uses the adjacency information as support if available.

## 3 POSE ESTIMATION OF PARTIAL B-REP MODELS

For representation of our B-Reps we use the half-edge data-structure [15], consisting of faces, half-edges and vertices. Since the model can be incomplete, not every half-edge must have an opposite half-edge. To describe physical features, we define the following terms (Fig. 1):

- A pair of opposite half-edges of a B-Rep is called **edge**.
- A vertex of a B-Rep with at least three adjacent faces is called **corner**.
- Edges, corners and faces are named **physical elements**.

To account for a partial matching, we employ local features for matching the models. Typically, local feature based methods contain the following three steps: First, **features** that represent special points of interest are found. **Descriptors** give a compact and comparable representation of a feature. Then, **correspondences** between features are determined utilizing their descriptors. Optionally, wrong correspondences are rejected. Last, a **pose hypothesis** is calculated using the correspondences.

The main reason for this procedure is that features are plenty in number and the pose is only uniquely estimable with a certain number of features. If features are located on points in space, at least three point-point correspondences are needed. When point clouds are registered, most elements of the cloud (the points themselves) have a corresponding point if they are located in a region where both clouds overlap. But this does not hold for elements of incomplete B-Reps: Vertices and half-edges do not necessarily match (see Fig. 1). Therefore we rely on the physical elements: faces, edges and corners. The number of these physical elements is much smaller compared to the number of points in a point cloud registration problem. But instead, they contain more information: A corner is not only a location in space, but also the common point of three faces each having a plane equation (containing a normal) and a centroid (for planar B-Reps).

These observations explain the key point of our approach: Because of the small number of physical elements containing a large amount of information, we design our features in a way that a single correspondence of features is enough to estimate the pose. Our approach can be summarized as follows:

(1) Find features and create a descriptor that contains a relative and an absolute part (Sect. 3.1)
(2) Find corresponding features using the relative part of the descriptor (Sect. 3.2)
(3) Create a pose hypothesis for each correspondence using the absolute part of the descriptor (Sect. 3.3)
(4) Find the best hypothesis using a quality measure (Sect. 3.4)

## 3.1 Features and descriptors

Traditionally, a feature is a point with special properties that can be recognized in different images. In B-Rep models the physical elements can be utilized: edges, faces and corners. In order to estimate the pose from one correspondence, the features must contain enough information to fix the six degrees of freedom of a rigid body motion. A correspondence between two corners with its at least three adjacent faces is sufficient whereas a pair of faces or edges is not. So in general, we define a **feature** as a set of three faces of a B-Rep. To find all features, we iterate over all sets of three faces of a B-Rep model and check if the three faces are linearly independent. This ensures that subsequent pose calculation is not prevented by e.g. parallelism.

**Descriptors** are used to compare features for similarity and to estimate the final pose. Suppose a feature of the linearly independent faces $f_1, f_2$ and $f_3$, then our descriptor $D$ is the $3 \times 5$-matrix

$$D(f_1, f_2, f_3) = \begin{bmatrix} \alpha_{23} & \vec{p}_1^T \\ \alpha_{13} & \vec{p}_2^T \\ \alpha_{12} & \vec{p}_3^T \end{bmatrix}$$

where $\vec{p}_i^T = (\vec{n}_i^T \quad d_i)$ are the plane coefficients of face $f_i$ with $\vec{n}_i \in \mathbb{R}^3$ being the normalized normal vector that points to the outside of a face and $d_i \in \mathbb{R}$ being the offset of the plane from the origin. The right $3 \times 4$ part of $D$ is called absolute part of the descriptor since it can be used to calculate the pose when a correspondence of features is given. The left $3 \times 1$ part is called relative part since it is translation- and rotation-invariant and can be used for similarity checks. The angles $\alpha_{ij} \in [0, 2\pi[$ are the interior angles between the faces $f_i$ and $f_j$. For example, $\alpha$ of two adjacent perpendicular faces would be $\pi/2$ if the faces build a convex edge, or $3\pi/2$ if they build a reflexive edge (Fig. 2).

Calculating $\alpha$ for two faces that share a common edge is simple since convexity can be determined using the direction of a half-edge:

$$\alpha_{ij} = \begin{cases} \pi - arccos(\vec{n}_i \cdot \vec{n}_j) & \text{if } (\vec{v}_i \times \vec{n}_i) \cdot \vec{n}_j > 0 \\ \pi + arccos(\vec{n}_i \cdot \vec{n}_j) & \text{else} \end{cases}$$

where $\vec{v}_i$ is the normalized direction of the half-edge of face $f_i$ where $f_i$ and $f_j$ are connected. See Fig. 2 for an example.

If two faces do not share a common edge, the angle is ambiguous. To decide on the convex or reflex angle, we use the following procedure: Consider the plane with normal $\vec{n}_i \times \vec{n}_j$ (see figures in Table 1). The projection of the planes $\vec{p}_i$ and $\vec{p}_j$ yields two lines $g_i$ and $g_j$ that intersect in a point $\vec{P}$. The projection of the polygon of $f_i$ and $f_j$ on $g_i$ resp. $g_j$ yields two line segments $s_i$ and $s_j$. Depending on whether the start and end point of the line segment lie on the same side of the line in relation to $\vec{P}$ , we can distinguish five cases as shown in Table 1: In Case A we use the convex angle, in Case B the reflex angle. In Cases C and D both faces cannot intersect because the normal directions are inconsistent. So there is no inner angle and we use $\alpha_{ij} = arccos(\vec{n}_i \cdot \vec{n}_j)$ in these cases. In all other cases, when at least one line segment spans both sides of $\vec{P}$, the angle is ambiguous and we discard the entire feature. An object where this ambiguity occurs is depicted in Fig. 2c.

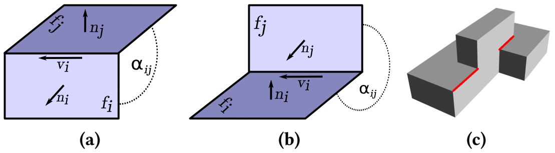


**Figure 2: (a) Convex edge. (b) Reflexive edge. (c) An object containing two faces that share both a convex and a reflexive edge (marked in red).**

This distinction is still advantageous since with incomplete B-Reps connected edges can be missing, but the faces are still uniquely convex or reflexive (for example in Fig. 1).

## 3.2 Correspondence detection

To detect whether two features $D_A$ and $D_B$ are similar, we use the relative part of the descriptor:

$$\vec{D}_{rel,A}^T = (\alpha_{23}, \alpha_{13}, \alpha_{12}), \quad \vec{D}_{rel,B}^T = (\beta_{23}, \beta_{13}, \beta_{12})$$

Since the order of the three faces is arbitrary, there are six possibilities for comparing the three angles of $D_A$ with $D_B$. We reorder the faces in each descriptor so that $\vec{n}_1, \vec{n}_2, \vec{n}_3$ build a right-handed system. This reduces the number of possibilities to three. The similarity can be expressed by

$$S(\vec{D}_{rel,A}, \vec{D}_{rel,B}) = \min \{S_0, S_1, S_2\}$$

with $S_i$ being the maximum norm of the differences during a cyclic shift by $i$:

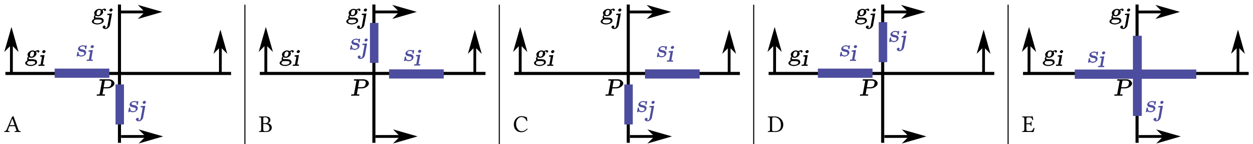
$$\begin{aligned} S_0 &= \max \{\alpha_{23} - \beta_{23},\ \alpha_{13} - \beta_{13},\ \alpha_{12} - \beta_{12}\} \\ S_1 &= \max \{\alpha_{23} - \beta_{13},\ \alpha_{13} - \beta_{12},\ \alpha_{12} - \beta_{23}\} \\ S_2 &= \max \{\alpha_{23} - \beta_{12},\ \alpha_{13} - \beta_{23},\ \alpha_{12} - \beta_{13}\} \end{aligned}$$

When $S(\vec{D}_{rel,A}, \vec{D}_{rel,B})$ is smaller than a threshold $\delta$, both features are selected to be corresponding.

## 3.3 Hypotheses

Since the absolute part of a descriptor contains enough information to estimate a pose, we build a pose hypothesis for every pair of corresponding features. The goal is to estimate the rigid body motion $T_A^B = [R \mid \vec{t}\ ]$ from the pair of corresponding descriptors $D_A$ and $D_B$. First, we reorder the rows of $D_B$ by cyclic shifting so that $S_0$ is the minimum of $(S_0, S_1, S_2)$. This ensures that the faces expressed in each row match. Next, we define a local coordinate system for a descriptor represented by the affine transform $K \in \mathbb{R}^{4\times4}$:

$$K(D) = \begin{bmatrix} \vec{n}_1 & \vec{n}_2 & \vec{n}_3 & \vec{J} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



| | | | | |
|---|---|---|---|---|
| type: convex | type: reflex | type: separated | type: separated | type: unknown |
| $\alpha_{ij} = \pi - arccos(\vec{n}_i \cdot \vec{n}_j)$ | $\alpha_{ij} = \pi + arccos(\vec{n}_i \cdot \vec{n}_j)$ | $\alpha_{ij} = arccos(\vec{n}_i \cdot \vec{n}_j)$ | $\alpha_{ij} = arccos(\vec{n}_i \cdot \vec{n}_j)$ | - |

**Table 1: Different cases for the calculation of angle $\alpha_{ij}$ for non-adjacent faces.**

with $\vec{J}$ being the point of intersection of the three planes $\vec{p}_1$,$\vec{p}_2$ and $\vec{p}_3$. The final pose can then be formulated as

$$T_A^B = K^{-1}(D_B) \cdot K(D_A).$$

If the normals of the planes in $D_A$ do not have exactly the same ratio of angles as in $D_B$, the resulting transform is affine containing anisotropic shearing and not a rigid body motion: $T_A^B = [A \mid \vec{t}\,]$. Therefore we apply a singular value decomposition (SVD) on the matrix $A = U\Sigma V^*$ and remove the anisotropic part by setting $R = UV^*$ to build the final rigid body motion $T_A^B = [R \mid \vec{t}\,]$.

### 3.4 Finding the best hypothesis

Given a set of pose hypotheses, we aim to find the one that best aligns the two B-Rep models. Therefore we propose a measure $m$ that indicates how much the faces of the models overlap. To calculate $m$ for one hypothesis, we perform the following steps:

(1) Transform B-Rep $A$ into the coordinate system of B-Rep $B$ using $T_A^B$

(2) Determine every pair of faces $(f_i, f_j)$ with $f_i \in A$ and $f_j \in B$ that satisfies these two criteria:

   (a) The plane equations of $f_i$ and $f_j$ are similar, i.e. the angle between the normals and the offset is below a threshold:

$$|\vec{n}_i \cdot \vec{n}_j| < \delta_{angular} \wedge |d_i - d_j| < \delta_{distance}$$

   (b) The polygons of $f_i$ and $f_j$ intersect. Note that this is a 2D intersection test on non-convex polygons that may contain holes.

(3) For each pair $(f_i, f_j)$, determine the intersection polygon(s) $G_{ij} = f_i \cap f_j$ and calculate the ratio

$$m_{ij} = \frac{\text{area}(f_i \cap f_j)}{\text{area}(f_i \cup f_j)} = \frac{\text{area}(G_{ij})}{\text{area}(f_i) + \text{area}(f_j) - \text{area}(G_{ij})}.$$

Sum up the values of all pairs:

$$m = \sum m_{ij}$$

The best hypothesis is the one with the greatest value of $m$.

Calculating the aforementioned procedure exactly is expensive, since it includes a non-convex polygon intersection for every pair of faces in every hypothesis. In worst case the time complexity is $O(h \cdot f^2 \cdot k \log k)$, where $h$ is the number of hypotheses, $f$ the number of faces in one B-Rep, and $k$ the number of half-edges in one face. Next, we propose an alternative approximate calculation of the area measure $m$ that is significantly faster:

As a preprocessing step, we calculate a face-aligned bounding-box for each face of the two B-Reps. A *face-aligned bounding-box* (FABB) is a 2D bounding box that lies in the face’s plane, encloses the face and is oriented to have minimum area. To calculate the FABBs, we employ the rotating-calipers algorithm [25]. The time complexity is $O(fk \log k)$ where $f$ is the number of faces and $k$ the number of edges in one face.

For speed up, we replace the intersection test of step 2(b) by a bounding box intersection test using the separating-axis algorithm [19], which can be performed in constant time. Additionally, we replace the polygon intersection (step 3) by a bounding box intersection, which has also constant time complexity, since the intersection of two rectangles contains at most eight edges. In summary, the time complexity is thus reduced to $O(fk \log k + hf^2)$.

Our procedure always finds the best hypothesis. But there can be situations where even the best pose is wrong, e.g. when the overlapping region of both B-Reps only contains linearly dependent faces. To detect theses situations, a conflict test can be performed that checks that faces of aligned models do not intersect in the interior of faces. For a fast, but approximate intersection test, the already calculated face-aligned bounding boxes can be used.

## 4 EVALUATION

To evaluate our method, we used the following experimental setup: To simulate a hand-held camera and simultaneously obtain ground-truth data, we attached an ENSENSO N10 depth camera (with a range of 20 to 90 cm) to a KUKA LWR IV robot arm (Fig. 3a). Using gravity compensation mode, the user is able to guide the robot by manually moving the robot and the mounted camera.

We used 3 scenes with different objects (Fig. 3b-d) and took 30 images per scene yielding 90 reconstructed partial B-Reps in total. For each pair of B-Reps of the same scene, we calculated the relative transformations $T_{truth}$ and $T_{experiment}$ from the robot poses and by executing our proposed procedure. The error is denoted as

$$T_{error} = T_{truth}^{-1} \cdot T_{experiment} = [R \mid \vec{t}\,]$$

We define the rotational error $\delta_{error}$ as the angle obtained by transforming the rotation matrix $R$ into an angle-axis representation and dropping the axis. Formally this can be written as

$$\delta_{error} = \arccos\left(0.5 \cdot (R_{00} + R_{11} + R_{22} - 1)\right).$$

We define the translational error $t_{error}$ as the amount by which the centroid point $\vec{p_c}$ of the B-Rep differs when applying $T_{error}$:

$$t_{error} = ||T_{error} \cdot \vec{p_c} - \vec{p_c}||_2$$

Because B-Reps may be distant from the origin, this is superior to $||\vec{t}||_2$ since it considers the translation at the point of interest, which is $\vec{p_c}$.

We define two datasets for evaluation: In the dataset **DS SEQ** all pairs of B-Reps are evaluated that are successive in the camera path or have at most one frame between. So the camera poses will not differ much. This dataset represents the common application

| | | *without conflict test* | | | | | | *with conflict test* | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\delta_{error}$ [rad] | | | $t_{error}$ [cm] | | | $\delta_{error}$ [rad] | | | $t_{error}$ [cm] | | |
| **dataset** | **area measure** | $\mu, \sigma$ | $Q_2$ | $Q_3$ | $\mu, \sigma$ | $Q_2$ | $Q_3$ | $\mu, \sigma$ | $Q_2$ | $Q_3$ | $\mu, \sigma$ | $Q_2$ | $Q_3$ |
| DS TOTAL | exact | 0.25, 0.62 | 0.018 | 0.046 | 2.5, 7.4 | 0.4 | 0.9 | 0.03, 0.14 | 0.015 | 0.029 | 0.6, 2.0 | 0.3 | 0.6 |
| | approximate | 0.23, 0.58 | 0.019 | 0.046 | 2.4, 7.4 | 0.4 | 1.0 | 0.03, 0.14 | 0.015 | 0.029 | 0.6, 2.0 | 0.3 | 0.6 |
| DS SEQ | exact | 0.03, 0.15 | 0.010 | 0.019 | 0.4, 1.1 | 0.2 | 0.3 | 0.01, 0.01 | 0.010 | 0.018 | 0.3, 0.3 | 0.2 | 0.3 |
| | approximate | 0.03, 0.15 | 0.011 | 0.019 | 0.4, 1.1 | 0.2 | 0.3 | 0.02, 0.02 | 0.011 | 0.019 | 0.3, 0.3 | 0.2 | 0.3 |

**Table 2: Average value $\mu$, standard deviation $\sigma$, median $Q_2$ and upper quartile $Q_3$ of error measures $\delta_{error}$ and $t_{error}$ for both datasets and for exact and approximate area measure of Sec. 3.4. For other quantiles see Fig. 4.**

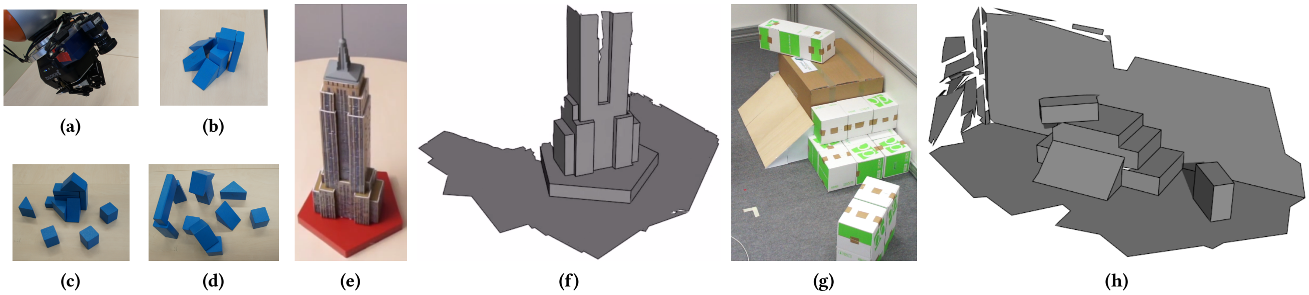


**Figure 3: For evaluation, a robot with a registered eye-in-hand depth camera (a) was used to reconstruct three different scenes (b-d). For each scene, thirty images from different poses have been acquired and compared to ground-truth. Images (e) and (g) show two of the scenes used for hand-held SLAM, (f) and (h) show the reconstructed models (see also accompanying video).**

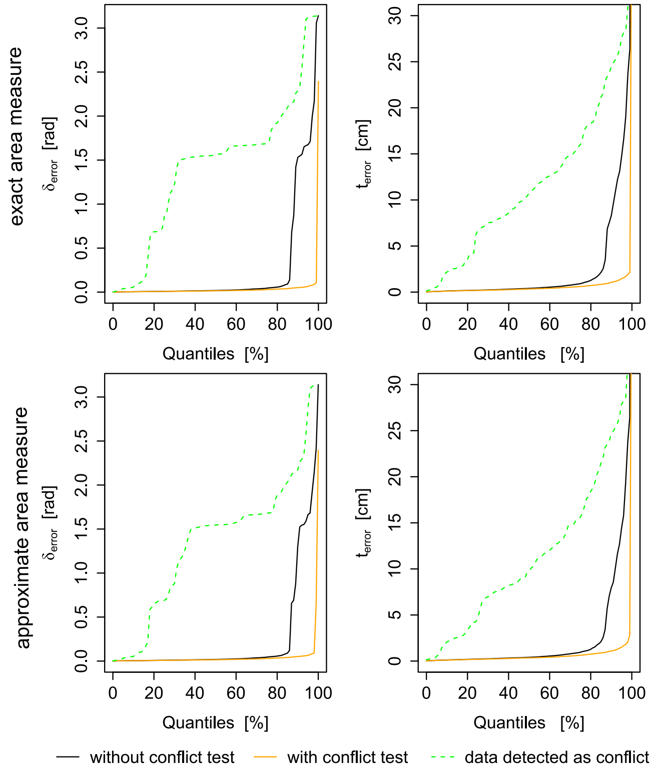


**Figure 4: Quantiles of angular error $\delta_{error}$ and translational error $t_{error}$ of the area measures of data set DS TOTAL.**

that subsequent images are incrementally registered and merged to the previous image like in SLAM applications. In the dataset **DS TOTAL** all combinations of two B-Reps from the same scene are evaluated. Thus it contains pairs whose B-Reps may not even overlap and registration is impossible.

Table 2 and Fig. 4 show the results of our evaluation. From these, the following observations can be made: The error is not normally distributed, the median differs much from the average value. The reason is that there are many samples with a low error, but also a few outliers with large error. These outliers originate from false matching of features, especially in the DS TOTAL dataset, where some pairs of B-Reps do not even overlap. As a consequence, it is better to consider quantiles instead of average and standard deviation. As Fig. 4 shows, about 75% of the samples have a translational error smaller than 1 cm for exact area measure. The quantile plots (Fig. 4) show that the conflict test discards wrong matchings and improves the results since higher quantiles have smaller values.

In the ideal case, the conflict test only detects conflicts in hypotheses with large error. Due to noise some false positives occur, too. This can be seen by the fact that the green dashed curve in Fig. 4 does not instantly rise near the 0%-quantile. About 6% of the conflicting samples have $t_{error}$ < 1 cm (approximate measure). But these false positives are not relevant in an online SLAM application: If one frame fails to match, the application can drop it and continue with the next one. Detecting true conflicts is by far more important. As expected the approximate area measure performs worse than the exact one, but is more than ten times faster (see Table 3). In real time applications, the speed improvement clearly compensates the slight increase in error.

The runtime for our matching procedure (using approximate measure) is about 50 milliseconds on average on a single core for the

| *average* | **DS TOTAL** | **DS SEQ** |
|---|---|---|
| number of hypotheses | 145 | 298 |
| runtime exact [ms] | 232 | 592 |
| runtime approx. [ms] | 18 | 47 |

**Table 3: Average number of hypotheses created during matching of datasets DS TOTAL and DS SEQ and average overall runtime in milliseconds depending on the used area measure (single core).**

DS SEQ dataset which has many corresponding features. Since the runtime depends on the number of hypotheses, it is clear that the average is lower for the DS TOTAL dataset, where some pairs of B-Reps do not even overlap.

To summarize, in the best configuration our method is able to match B-Reps of subsequent frames (data set DS SEQ) with an average error of 3 millimeters resp. 1.2 degrees.

As stated in the introduction, our approach in combination with [18] can be used as complete SLAM system for the reconstruction of B-Rep models with hand-held cameras. The accompanying video shows the reconstruction of three scenes, two of them are depicted in Fig. 3e-h.

There are also limitations of our approach: Since we do not constrain our estimated pose, matching B-Reps of symmetric objects may be wrong because there's a pose with more overlap. This is the reason for the missing backside of model in Fig. 3f. Wrong alignments can also occur when the overlapping part of two partial B-Reps only contains linearly dependent faces.

## 5 CONCLUSIONS

In this paper we presented a method to match and estimate the relative pose of two planar B-Rep models. In particular, we are able to match partial and noisy models that contain very few and differing geometric characteristics like e.g. corners (Fig. 1). This is achieved by calculating a feature set with features that - in contrast to other methods - are not plenty in number, but contain enough information to estimate a pose hypothesis from a single correspondence. For selection of the best hypothesis we proposed a measure based on the area of overlap of the models.

When we combine our approach with our previous work [18], in which partial B-Rep models are incrementally reconstructed and merged, we yield a complete simultaneous location and mapping (SLAM) system that directly generates and operates on B-Rep models. If a model is given (e.g. from CAD data), our matching procedure can also be used as a localization method for a hand-held camera. A shortcoming of our approach is that alignment may fail on symmetric objects or when the overlap only contains linearly dependent faces. To account for this, future work can be the additional use of color, since we currently only rely on geometric features.

## ACKNOWLEDGMENTS

This work has partly been supported by the Deutsche Forschungsgemeinschaft (DFG) under grant agreement He2696/16.

## REFERENCES

[1] Roseline Bénière, Gérard Subsol, Gilles Gesquière, François Le Breton, and William Puech. 2013. A comprehensive process of reverse engineering from 3D meshes to CAD models. *Computer-Aided Design* 45, 11 (2013), 1382 – 1393.

[2] Pál Benkő, Ralph R. Martin, and Tamás Várady. 2001. Algorithms for reverse engineering boundary representation models. *Computer-Aided Design* 33, 11 (Sept. 2001), 839–851.

[3] Paul J. Besl and Neil D. McKay. 1992. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* 14, 2 (Feb. 1992), 239–256.

[4] Frédéric Bosché. 2010. Automated recognition of 3D CAD model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction. *Advanced Engineering Informatics* 24, 1 (2010), 107 – 118. Informatics for cognitive robots.

[5] Dirk Buchholz. 2015. *Bin-Picking - New Approaches for a Classical Problem.* Dissertation. Braunschweig.

[6] Brian Curless and Marc Levoy. 1996. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques (SIGGRAPH '96).* ACM, New York, NY, USA, 303–312.

[7] I. Dryanovski, W. Morris, R. Kaushik, and Jizhong Xiao. 2012. Real-time pose estimation with RGB-D camera. In *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on.* 13–20.

[8] Mohamed El-Mehalawi and R. Allen Miller. 2003. A database system of mechanical components based on geometric and topological similarity. Part I: representation. *Computer-Aided Design* 35, 1 (2003), 83 – 94.

[9] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard. 2012. An evaluation of the RGB-D SLAM system. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on.* 1691–1696.

[10] Jianbing Huang and Chia-Hsiang Menq. 2003. Automatic CAD Model Reconstruction from Multiple Point Clouds for Reverse Engineering. *Journal of Computing and Information Science in Engineering* 2, 3 (Jan. 2003), 160–170.

[11] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon. 2011. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology (UIST '11).* ACM, New York, NY, USA, 559–568.

[12] C. Kerl, J. Sturm, and D. Cremers. 2013. Dense Visual SLAM for RGB-D Cameras. In *Proc. of the Int. Conf. on Intelligent Robot Systems (IROS).*

[13] C. Kerl, J. Sturm, and D. Cremers. 2013. Robust Odometry Estimation for RGB-D Cameras. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA).*

[14] Tae-kyeong Lee, Seungwook Lim, Seongsoo Lee, Shounan An, and Se-young Oh. 2012. Indoor mapping using planes extracted from noisy RGB-D sensors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012.* 1727–1733.

[15] Martti Mäntylä. 1987. *An Introduction to Solid Modeling.* Computer Science Press, Inc., New York, NY, USA.

[16] David McWerther, Mitchell Peabody, William Regli, and Ali Shokoufandeh. 2001. Transformation Invariant Shape Similarity Comparison of Solid Models. In *ASME Design Engineering Technical Confs. 6th Design for Manufacturing Conf.*

[17] K. Pathak, N. Vaskevicius, J. Poppinga, M. Pfingsthorn, S. Schwertfeger, and A. Birk. 2009. Fast 3D mapping by matching planes extracted from range sensor point-clouds. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2009.* 1150 –1155.

[18] Maximilian Sand and Dominik Henrich. 2016. Incremental reconstruction of planar B-Rep models from multiple point clouds. *The Visual Computer* 32, 6 (2016), 945–954. https://doi.org/10.1007/s00371-016-1247-7

[19] Philip Schneider and David H. Eberly. 2003. *Geometric Tools for Computer Graphics.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[20] Yifei Shi, Pinxin Long, Kai Xu, Hui Huang, and Yueshan Xiong. 2016. Data-driven Contextual Modeling for 3D Scene Understanding. *Comput. Graph.* 55, C (April 2016), 55–67.

[21] V. Stamati and I. Fudos. 2010. *Building Editable B-rep Models from Unorganized Point Clouds.* Technical Report TR-2010-04. Computer Science Department.

[22] F. Steinbruecker, J. Sturm, and D. Cremers. 2011. Real-Time Visual Odometry from Dense RGB-D Images. In *Workshop on Live Dense Reconstruction with Moving Cameras at the Intl. Conf. on Computer Vision (ICCV).*

[23] Y. Taguchi, Yong-Dian Jian, S. Ramalingam, and Chen Feng. 2013. Point-plane SLAM for hand-held 3D sensors. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on.* 5182–5189.

[24] J.W.H. Tangelder and R.C. Veltkamp. 2004. A survey of content based 3D shape retrieval methods. In *Shape Modeling Applications, 2004. Proceedings.* 145–156.

[25] Godfried Toussaint. 1983. Solving Geometric Problems with the Rotating Calipers. *In Proc. IEEE MELECON 83* (1983).

[26] T. Whelan, J. B. McDonald, M. Kaess, M. F. Fallon, H. Johannsson, and J. J. Leonard. 2012. Kintinuous: Spatially Extended KinectFusion. In *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras.* Sydney, Australia.

[27] Yizhong Zhang, Weiwei Xu, Yiying Tong, and Kun Zhou. 2015. Online Structure Analysis for Real-Time Indoor Scene Reconstruction. *ACM Trans. Graph.* 34, 5 (Nov. 2015), 159:1–159:13.