

Object detection using boundary representations of primitive shapes

Nikhil Somani¹, Alexander Perzylo¹, Caixia Cai², Markus Rickert¹, Alois Knoll²

Abstract—In this paper, an approach for matching of primitive shapes detected from point clouds, to boundary representations of primitive shapes contained in CAD models of objects/workpieces is presented. The primary target application is object detection and pose estimation from noisy RGBD sensor data. This approach can also be used to determine incomplete object poses, including those of symmetrical objects. Detection and reasoning about these under-specified object poses is useful in several practical applications such as robotic manipulation, which are also presented in this paper.

I. INTRODUCTION

Object recognition and pose estimation from 3D sensor data using CAD models is a classic problem in computer vision. This work is focused on detection of industrial objects/workpieces that are typically textureless. Hence, we focus on purely shape based approaches. Previous work in this area features primitive shape graphs [1], [2], surflet-pair based approaches [2], [3], [4], and the triple-point feature method [5]. Keypoint and descriptor based approaches include local shape keypoint [6] and global descriptors [7]. Each of these approaches have their own advantages and disadvantages. Global descriptors such as VFH [7] require a cumbersome training phase, where a large number of object views need to be generated by real experiments. Besides, their accuracy decreases significantly in case of occlusions and partial views. The advantage of these methods lies in their computational speed. On the other hand, methods such as [1], [3], [4], are designed to be robust to partial views, occlusions and noisy data but don't scale well for real-time applications or large point clouds. Approaches for efficient object detection and pose estimation using CAD models have been presented in our previous work [2].

Another important property of the typical industrial parts that we deal with, is that they are often composed of simpler geometric parts. Hence, their geometries can be approximated accurately using a set of primitive shapes such as planes, cylinders, spheres, etc. A key idea in our work is to utilize the mathematical properties of such geometric primitives for object detection, since they can be detected more accurately and efficiently than complete CAD models. We use a boundary representation (BREP) of CAD models, since it retains the information about primitive geometric shapes. Also, BREP is a standard supported by several popular CAD modeling softwares (e.g. SolidWorks) and

exchange formats (e.g., STEP, IGES). In [8], we presented an ontological representation of BREP-based CAD models, which we also use in this work.

In [2], [9], we used an energy minimization approach for decomposing object models (in point cloud format) into primitive shapes. This approach discarded the information about primitive shapes contained in the CAD models (stored in STEP, IGES files) and tried to estimate it again, possibly introducing errors in the process. In this work, we directly use the information of primitive shapes stored in CAD models, thereby achieving a perfect decomposition of the CAD models into primitive shapes.

In [2], we presented the concept of object detection using sets of primitive shapes that are minimal sets for 3D pose estimation. However, possible minimal combinations such as 3 planes or a plane and a cylinder were defined explicitly. In [9], we modelled the constraints that each primitive shape enforces on the object's pose and could detect these minimal sets automatically from a set of shapes. The presented object recognition approach could also handle over-specified constraints and estimate the pose in a least-squares sense.

For estimating object poses, we propose the use of a non-linear least squares solver that supports inequality constraints. The constraints introduced by primitive shape matching are modified to be inequality constraints, since this enables explicit modeling of sensor noise in RGBD data and errors in fitting of primitive shapes to scene point clouds, in the form of bounds for these inequality constraints. Hence, instead of matching primitive shapes exactly we explicitly model and allow tolerances, which leads to a better convergence of the pose estimation process.

In Section II-A, we briefly describe the boundary representation (BREP) format for modeling objects. The ontological representation of BREP objects is explained in Section II-B. Mathematical and ontological modeling of constraints from matching of primitive shapes, is described in Section III. Algorithms for object detection and pose estimation using primitive shape matches are presented in Section IV. Finally, applications of the presented approaches are described in Section V.

II. ONTOLOGY FOR BOUNDARY REPRESENTATION OF GEOMETRIC SHAPES

Using an ontological representation of CAD models enables us to easily link our work to important applications such as robotic manipulation and intuitive robot task programming.

Authors Affiliation: ¹ fortiss GmbH, An-Institut Technische Universität München. ² Technische Universität München, Fakultät für Informatik. Email: somani@fortiss.org

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 287787.

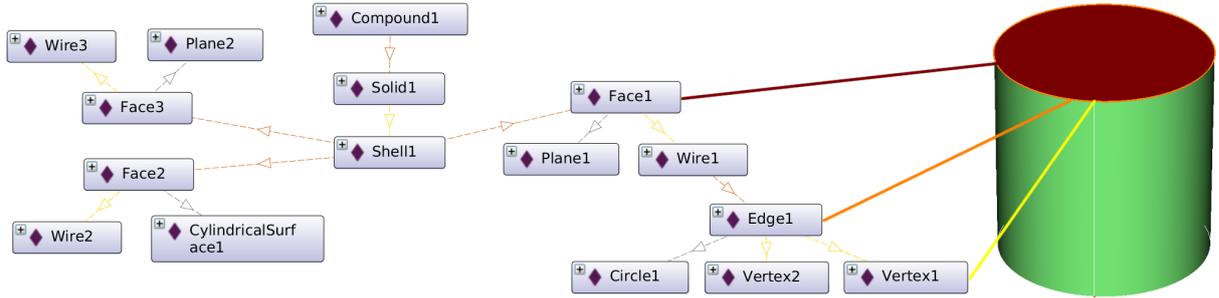


Fig. 1: The ontological representation of a simple BREP-based CAD model. The corresponding ontological instances for some of the points, lines and planes of the object have been indicated.

A. Boundary Representation of CAD models

A Boundary Representation (BREP) of CAD models uses mathematical models of primitive shape elements to describe the geometric properties of points, curves, surfaces and volumes. CAD models are constructed by defining boundaries to these (often unbounded) geometric entities. The BREP specification distinguishes between geometric and topological entities, as illustrated in Fig. 2. The topological entities describe the arrangement and connections between geometric entities, while numerical data and mathematical definitions of the base shapes are contained in the geometric entities.

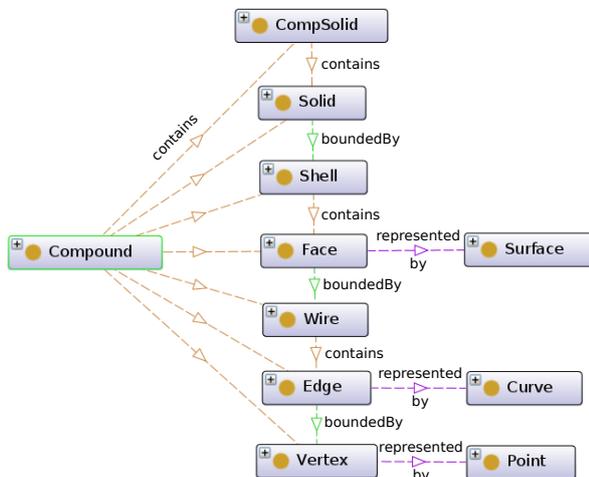


Fig. 2: Overview of the BREP structure. The data model contains the topological entities, and corresponding geometric entities (indicated by a *representedBy* property).

B. Ontological representation of shapes and constraints

Fig. 1 illustrates the representation of a simple CAD model using the BREP ontology. In our formulation, a geometric constraint is defined between two geometric entities: a fixed entity with a defined pose and a constrained entity whose pose depends on the fixed entity and the constraint itself. For the pose estimation problem, the fixed entities are the primitive shapes detected in the scene and the constrained entities are the primitive shapes in the model, whose pose needs to be estimated. The nullspace of a geometric constraint is a set

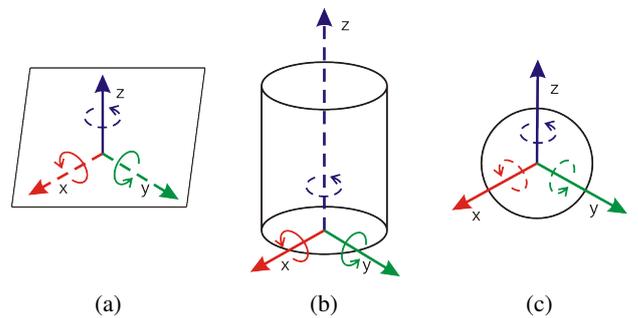


Fig. 3: Geometric inter-relational constraints enforced by each primitive shape match. Constrained/un-constrained axes are indicated by solid/dotted arrows respectively. (a) Infinite Plane: The position along z-axis (normal direction) is fixed, while positions along x and y are free. Rotation is allowed only around z-axis. (b) Infinite Cylinder: The position along x and y axes is fixed, while position along z-axis (principal axis of cylinder) is not fixed. Rotation is allowed only around the z-axis. (c) Sphere: The positions in all axes are fixed. Rotation is allowed around all axes.

of relative transformations between the involved geometries that satisfy the constraint.

III. GEOMETRIC CONSTRAINTS FROM PRIMITIVE SHAPE MATCHING

Each primitive shape P_i enforces a set of constraints (C_{p_i}, C_{n_i}) on the position and orientation of the object respectively. Each row of C_{p_i} and C_{n_i} contains a direction along which the constraint has been set. Examples of constraints set by each primitive shape are shown in Fig. 3 and explained below:

- Infinite plane: $C_{p_i} = [0, 0, 1]$ and $C_{n_i} = [1, 0, 0; 0, 1, 0]$
- Infinite cylinder: $C_{p_i} = [1, 0, 0; 0, 1, 0]$ and $C_{n_i} = [1, 0, 0; 0, 1, 0]$
- Sphere: $C_{p_i} = [1, 0, 0; 0, 1, 0; 0, 0, 1]$ and $C_{n_i} = \phi$

A *complete set* of primitive shapes is defined as a set where the constraints fully specify the 3D position and orientation of the object. A *minimal set* of primitive shapes is defined as a set which is complete but removing any primitive shape from the set would render it incomplete.

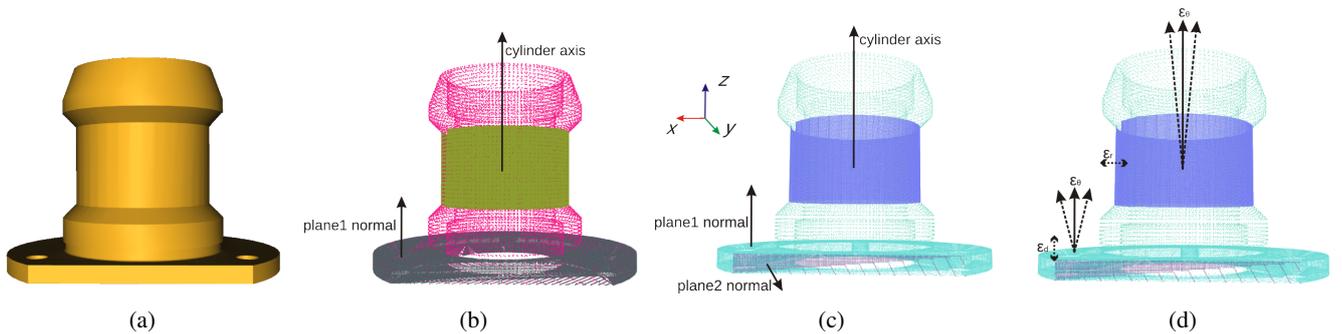


Fig. 4: Primitive shape groups in an industrial workpiece. (a) Object model. (b) The position of the object is fully defined in 3D. However, rotation of the object along the z-axis is not fixed. (c) Position and orientation of the object are both fully defined and these primitive shapes form a minimal set. (d) The expected errors or tolerances in estimation of primitive shapes from point cloud data are shown (exaggerated).

TABLE I: Summary of supported constraints between primitive shapes

Fixed	Constrained	Constraint (i)	Cost Function (g_i)	Lower Bound (lb)	Upper Bound (ub)
Plane ₁	Plane ₂	Parallel Distance (d_{\min}, d_{\max})	$[n_1^T \hat{p}_{21}; \hat{n}_2^T n_1]$	$[d_{\min}; a_{\min}]$	$[d_{\max}; a_{\max}]$
Cylinder ₁	Cylinder ₂	Parallel Distance ($d_{\min}, d_{\max}, a_{\min}, a_{\max}$)	$[\hat{p}_{21} - (n_1^T \hat{p}_{21})n_1 _2^2; \hat{n}_2^T n_1]$	$[d_{\min}^2; a_{\min}]$	$[d_{\max}^2; a_{\max}]$
Sphere ₁	Sphere ₂	Distance (d_{\min}, d_{\max})	$[\hat{p}_{21}]$	d_{\min}	d_{\max}

Table I presents the list of supported geometric constraints between primitive shapes, where

$$\begin{aligned} \hat{p}_2 &= \mathbf{R}p_2 + t \\ \hat{p}_{21} &= \hat{p}_2 - p_1 \\ \hat{n}_2 &= \mathbf{R}n_2 \end{aligned}$$

A. Feature Vectors for Sets of Primitive Shapes

For detection of objects in the scene, correspondences between the scene shape primitives and model shape primitives need to be determined. We obtain these correspondences by computing and matching feature vectors from the geometric properties of the primitive shapes. These feature vectors not only encode the geometric properties of the shapes, but also of the relations between the shapes.

Some of the feature vectors used for primitive shapes and their combinations are defined in Table II. The table only defines some simple combinations of feature vectors. However, more complicated features involving more primitive shapes can be constructed from these representations by simple extensions.

Using this approach, a feature vector can be constructed for the entire object as well. The visibility of primitive shapes depends on the viewpoint of the camera. Hence, feature vectors for all possible sets of primitive shapes should be calculated offline for the object models. Minimal sets of primitives from the scene point cloud are calculated during the pose estimation stage (see Section IV-B), and the distance between the feature vectors provides a metric for obtaining hypotheses of shape associations.

TABLE II: Feature Vectors for Primitive shape sets

Primitive shape	Feature Vector (fv)
Inf. Plane	ϕ
Sphere	$radius$
Inf. Cylinder	$radius$
Plane+Plane	$fv(plane1), fv(plane2),$ $angle(plane1_normal, plane2_normal),$ $min_distance(plane1, plane2)$
Plane+Cylinder	$fv(cylinder), fv(plane),$ $angle(plane_normal, cylinder_axis)$
Cylinder+Cylinder	$fv(cylinder1), fv(cylinder2),$ $angle(cylinder1_axis, cylinder2_axis),$ $min_distance(cylinder1, cylinder2)$
Plane+Plane+Cylinder	$fv(plane1, cylinder), fv(plane2, cylinder)$

IV. CONSTRAINT PROCESSING FOR INCOMPLETE POSE ESTIMATION

A. Detection of minimal and complete sets of primitives

The constraints (C_{p_i}, C_{n_i}) enforced by each primitive shape P_i are stacked into two matrices C_p and C_n (each having 3 columns), where C_p represents the combination of constraints for the position and C_n represents the combination of constraints on the orientation. The constraints are complete if the matrices C_p and C_n both have rank 3. Fig. 4 shows an example of a complete set of primitive shapes. An algorithm for detecting minimal sets is presented in Algorithm 1.

B. Constraint solving for pose estimation

Let us consider the object's frame to be specified at position t and orientation (rotation matrix) \mathbf{R} . This can also be represented in the form of a homogeneous transformation

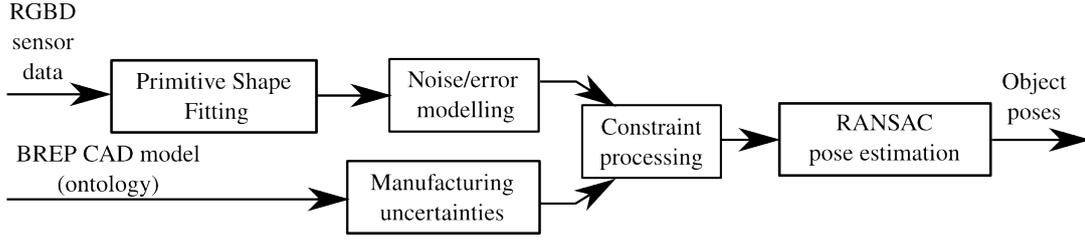


Fig. 5: Pipeline for primitive shape based object detection from noisy sensor data

Algorithm 1 Detecting minimal and complete primitive shape sets

Input : $[P_i]$ (set of primitive shapes)
Output : $[[P_i]_{\min}], [[P_i]_{\text{complete}}]$ (sets of minimal and complete primitive shapes)
forall P_i
 $\text{min_flag} \leftarrow \text{true}$
 $[P]_{\text{candidate}} \leftarrow P_i$
 forall P_j , where $j > i$
 $[P]_{\text{candidate}} \leftarrow [P]_{\text{candidate}} \cup P_j$
 If $\text{check_complete}([P]_{\text{candidate}})$
 $[[P_i]_{\text{complete}}] \leftarrow [[P_i]_{\text{complete}}] \cup [P]_{\text{candidate}}$
 If min_flag
 $\text{min_flag} \leftarrow \text{false}$
 $[[P_i]_{\min}] \leftarrow [[P_i]_{\min}] \cup [P]_{\text{candidate}}$
 EndIf
 EndIf
 EndFor
EndFor

matrix T . From the object models, the relative pose (partially defined) T_i of each primitive shape P_i w.r.t. the object's frame is available.

The optimization is performed over transformations that align the model to the objects in the scene. The transformations are represented as $\Delta \mathbf{x} = (\mathbf{t}, \mathbf{r})$ where \mathbf{t} is the translation and \mathbf{r} is the rotation in axis angle representation.

The function to be optimized is the absolute value of the transformation. In other words, this means minimizing $\|\Delta \mathbf{x}\|_2$. The constraint functions g_i along with their limits ($\text{lb}(g_i)$, $\text{ub}(g_i)$) are obtained from the primitive shape matching constraints shown in Table I. The (d_{\min}, d_{\max}) values of the constraints can be used to incorporate the noise in sensor data or primitive shape fitting errors (see Section V-A), as well as manufacturing uncertainties (see Section V-E). As an example shown in Fig. 4: If the error in estimation of a cylinder's radius (r) is ϵ_r , the bounds for the cylinder matching constraint can be set as $(d_{\min}, d_{\max}) = (r - \epsilon_r/2, r + \epsilon_r/2)$. For an error ϵ_θ in estimation of the normal direction, the bounds can be set as $(a_{\min}, a_{\max}) = (\cos(\epsilon_\theta), 1)$.

The resulting optimization problem is:

$$\begin{aligned} & \arg \min_{\Delta \mathbf{x}} \quad \|\Delta \mathbf{x}\|_2 \\ & \text{subject to} \quad \text{lb}(g_i) \leq g_i \leq \text{ub}(g_i), \quad i = 1, \dots, m. \end{aligned}$$

This set of equations is then solved using a non-linear least squares min-max solver (MA27) from [10] using the deterministic non-linear optimization utility from library Coin-OR (named IPOPT) [11].

In this way, the constraints defined by the primitive shapes can be solved to obtain a pose of the object. If the constraints are complete, the pose is uniquely defined. Otherwise, the constraint solver returns one possible solution.

C. RANSAC based constraint solving for pose estimation

A shape matching hypothesis H_i consists of a set of associations between primitive shape sets, i.e., correspondences that relate each primitive shape in the scene to a primitive shape in the model. The hypotheses can be computed by matching feature vectors, see Section III-A. An algorithm for pose estimation using RANSAC-like iterations on minimal sets of primitive shapes is described in Algorithm 2.

We use an efficient hypothesis verification approach that utilizes the geometric information from CAD models and primitive shape decomposition of scene point clouds, as described in [2].

V. EVALUATION AND APPLICATIONS

A. Calculating object poses from noisy sensor data

The primary application of this work is the detection and pose estimation of objects from noisy sensor data. Primitive shapes such as planes, cylinders and spheres are detected from RGBD data, as explained in [2]. This step also provides

Algorithm 2 Detecting object poses using RANSAC

- 1: **Input** : $[P_s, [P_m]_{\min}]$ (set of scene primitive shapes and minimal sets of model primitive shapes)
- 2: **Output** : $[T, s_{\max}]$ (best pose estimate with score for detected object instance)
- 3: **forall** $P_i \in [P_m]_{\min}$
- 4: $s_{\max} \leftarrow 0$
- 5: compute shape matching hypothesis (H_i) using fv's, see Section III-A
- 6: calculate transformation estimate T_i for H_i , see Section IV-B
- 7: compute score s_i for hypothesis H_i
- 8: **If** $s_i \geq \text{thresh} \ \& \ s_i > s_{\max}$
- 9: $T \leftarrow T_i$
- 10: $s_{\max} \leftarrow s_i$
- 11: **EndFor**

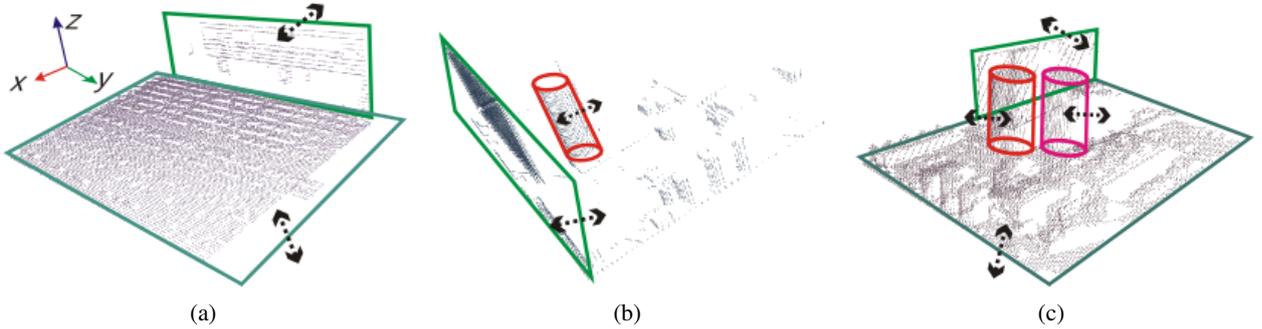


Fig. 6: Primitive shape groups for different views of an object. Using primitive shape sets, it can be calculated whether an object’s pose can be fully estimated from a viewpoint or not. The arrows indicate the expected noise (qualitative only) in estimation of the primitive shape parameters. (a) the position of the object along y axis is not determined. In (b) and (c) the complete pose of the object can be estimated. (Note: The detected primitive shapes are highlighted manually to make the image clearer.)

estimates of the shape fitting errors. These noise estimates can be expressed as inequality constraints and solved using the constraint solving algorithms (see Section IV). This pipeline is explained in Fig. 5.

B. Calculating detectability of objects from viewpoints

Depending on the viewpoint of the camera, different sets of primitive shapes might be visible in the scene point cloud. Also, the accuracy of the primitive shape detection depends on the distance of the object and the viewpoint, as shown in Fig. 6. Using Algorithm 1, it can be ascertained whether the object is detectable from a given viewpoint or not. This knowledge is important in object recognition and pose estimation problems, where primitive shape based approaches might fail due to insufficient data available from a certain viewpoint.

C. Reasoning about symmetrical objects

The ontological representation of CAD models is used to obtain geometric information about the CAD models. The information about minimal and complete sets of primitive shapes can be propagated back to the ontology and utilized by different applications. Information about the nullspace of geometric constraints, now available in the ontology, can also be used for several robotic manipulation applications, see Section V-D.

D. Manipulation of symmetric objects

Knowledge about the symmetrical properties of objects/workpieces can be used to optimize robotic manipulation tasks involving these objects. Fig. 7 shows a grasping task involving a symmetrical object. Using the object recognition approach presented in this paper, the robotic system is aware of the fact that the z -axis of the object is an axis of symmetry. Hence, as illustrated in Fig. 7, the object can be grasped using any orientation of the end effector along the z -axis. This allows optimization of the robot’s pose within this nullspace. More such applications have been described in our works [12], [13].

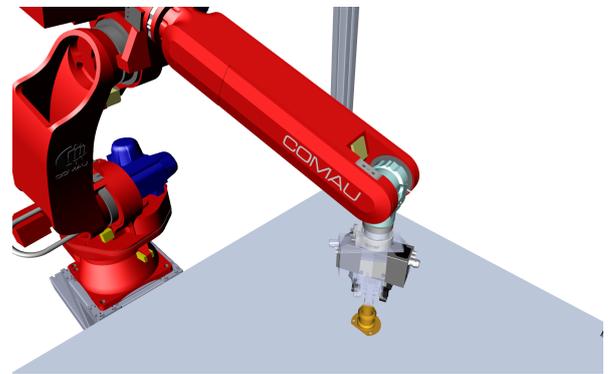


Fig. 7: Grasping of a symmetric object: The grasp pose for the object is invariant to rotations along the symmetrical axis of the cylindrical object. Two such grasping poses are shown in this figure.

E. Manufacturing uncertainties in object models

The manufactured objects might differ from the nominal CAD models due to manufacturing tolerances. In case of assembled objects, there can also be errors in the placement of individual parts. Fig. 8 shows an electronic circuit, where the capacitors (blue cylinders) are supposed to be mounted perpendicular to the circuit board. On an actual workpiece, the capacitors (red cylinders) are mounted at a slightly different angle. Also, the resistors (blue cuboids) are mounted at a slightly different location on the real workpiece (red cuboids). These are not errors but allowed tolerances in the manufacturing process.

Using our object recognition approach (see Sections III and IV-B), these uncertainties can be modeled explicitly and even such imperfect object models can be detected accurately.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we presented an approach for detection and pose estimation of industrial workpieces that can be decomposed into primitive shapes. An important contribution

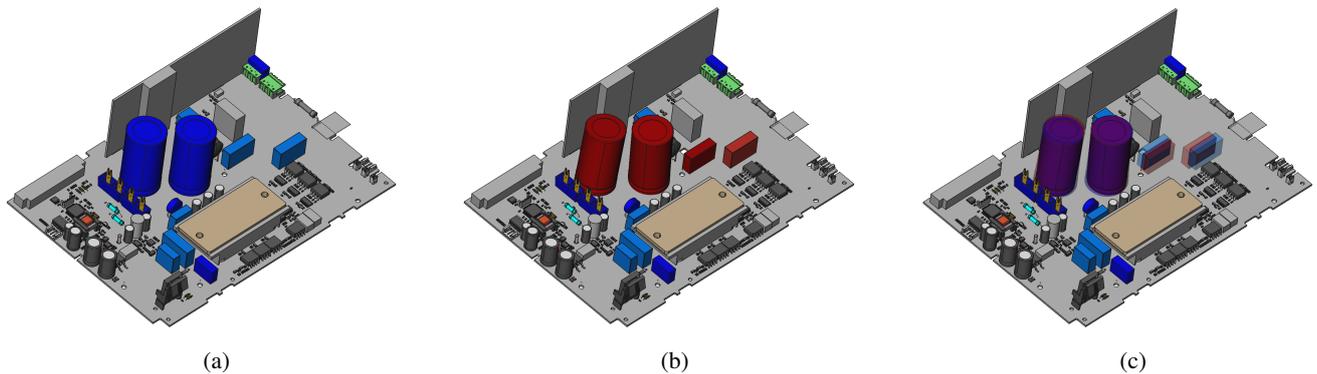


Fig. 8: Manufacturing tolerances in production or assembly of parts. (a) Nominal object CAD model (b) Actual manufactured model, with parts in Red having some placement errors (c) Superimposed image of the perfect and manufactured object, that illustrates the manufacturing tolerances.

in this work is the possibility to incorporate sensor noise and errors in primitive shape fitting directly into the non-linear optimization framework. Another contribution is the use of boundary representation for CAD models, stored in the form of an ontology. This not only enables accurate primitive shape decompositions of objects models, but also the propagation of information about geometric symmetries back into the ontology. This information is useful for several applications, which have also been briefly presented.

Future work includes implementation of more primitive shapes that are supported by the BREP standard. Also, a quantitative analysis of the benefits of this approach in robotic applications can be further investigated.

REFERENCES

- [1] R. Schnabel, R. Wessel, R. Wahl, and R. Klein, "Shape recognition in 3d point-clouds," in *The 16-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2008*, V. Skala, Ed. UNION Agency-Science Press, Feb. 2008.
- [2] N. Somani, E. Dean-Leon, C. Cai, and A. Knoll, "Scene perception and recognition in industrial environments for human-robot interaction," in *9th International Symposium on Visual Computing*, July 2013.
- [3] C. Papazov, S. Haddadin, S. Parusel, K. Krieger, and D. Burschka, "Rigid 3D geometry matching for grasping of known objects in cluttered scenes," *International Journal of Robotic Research*, vol. 31, pp. 538–553, 2012.
- [4] C. Papazov and D. Burschka, "An efficient ransac for 3d object recognition in noisy and occluded scenes," *Proceedings of the 10th Asian conference on Computer vision - Volume Part I*, pp. 135–148, 2011. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1964320.1964334>
- [5] U. Thomas, "Stable pose estimation using ransac with triple point feature hash maps and symmetry exploration," in *MVA*, 2013, pp. 109–112.
- [6] Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3d object recognition," in *Computer Vision Workshops (ICCV Workshops), 2009*, 2009, pp. 689–696.
- [7] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3d recognition and pose using the viewpoint feature histogram," in *Intelligent Robots and Systems (IROS), 2010*. IEEE, 2010, pp. 2155–2162.
- [8] A. Perzylo, N. Somani, M. Rickert, and A. Knoll, "An ontology for cad data and geometric constraints as a link between product models and semantic robot task descriptions," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015.
- [9] N. Somani, C. Cai, A. Perzylo, M. Rickert, and A. Knoll, "Object recognition using constraints from primitive shape matching," in *Proceedings of the International Symposium on Visual Computing*. Springer, Dec. 2014.
- [10] "HSL: a collection of fortran codes for large scale scientific computation," 2013. [Online]. Available: <http://www.hsl.rl.ac.uk>
- [11] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.
- [12] A. Perzylo, N. Somani, S. Profanter, M. Rickert, and A. Knoll, "Toward efficient robot teach-in and semantic process descriptions for small lot sizes," in *Proceedings of Robotics: Science and Systems (RSS), Workshop on Combining AI Reasoning and Cognitive Science with Robotics*, Rome, Italy, July 2015, <http://youtu.be/B1Qu8Mt3WtQ>.
- [13] N. Somani, A. Gaschler, M. Rickert, A. Perzylo, and A. Knoll, "Constraint-based task programming with cad semantics: from intuitive specification to real-time control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.