



## Fast visual part inspection for bin picking

**Rofalis, Nikolaos; Olesen, Anders Sig; Jakobsen, Michael Linde; Kruger, Volker**

*Published in:*

Proceedings of 2016 IEEE International Conference on Imaging Systems and Techniques

*Link to article, DOI:*

[10.1109/IST.2016.7738261](https://doi.org/10.1109/IST.2016.7738261)

*Publication date:*

2016

*Document Version*

Peer reviewed version

[Link back to DTU Orbit](#)

*Citation (APA):*

Rofalis, N., Olesen, A. S., Jakobsen, M. L., & Kruger, V. (2016). Fast visual part inspection for bin picking. In Proceedings of 2016 IEEE International Conference on Imaging Systems and Techniques (pp. 412-17). IEEE. (2016 IEEE International Conference on Imaging Systems and Techniques (ist)). DOI: 10.1109/IST.2016.7738261

## DTU Library

Technical Information Center of Denmark

---

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# Fast visual part inspection for bin picking

Nikolaos Rofalis

Robotics, Vision and Machine Intelligence Lab  
Department of Mechanical and Manufacturing Engineering  
Aalborg University, Denmark  
Email: niko@m-tech.aau.dk

Anders S. Olesen

DTU Fotonik, Department of Photonics Engineering  
Technical University of Denmark  
Email: aole@fotonik.dtu.dk

Michael L. Jakobsen

DTU Fotonik, Department of Photonics Engineering  
Technical University of Denmark  
Email: mlja@fotonik.dtu.dk

Volker Krüger

Robotics, Vision and Machine Intelligence Lab  
Department of Mechanical and Manufacturing Engineering  
Aalborg University, Denmark  
Email: vok@m-tech.aau.dk

**Abstract**—In this paper we present a novel 3D sensing approach for industrial bin-picking applications that is low-cost, fast and precise. The system uses a simple laser-line emitter. While a robot arm moves the object through the laser light, a synchronized camera captures the laser line image on the object. A full point cloud as well as an edge point cloud suitable for subsequent pose estimation is generated by the developed system. The aim of this work is to deliver an accurate point cloud based on which an object pose can be generated to support a manufacturing robot to deliver an object with high precision. The experimental evaluation of our system shows robust and accurate scanning results.

## I. INTRODUCTION

Nowadays, a robot is only able to deal with limited uncertainty in part location during bin picking. If parts are well organized in the bins, a robot can pick them with a precision which is good enough for the assembly process. However, it is a costly and time consuming process [1] to collect and organize parts for such assembly robots.

An alternative is Bin Picking [1], where the assembly robot is able to pick up parts that are randomly distributed in a bin. In the bin the parts are possibly on top of each other, they have random positions and orientations. To pick them, the robot must use advanced computer vision approaches to locate the part in terms of their exact location and orientation in the bin. Then, the robot must pick them one by one, as the robot can get access to them.

For bin-picking, we use the tool-unit from Scape Technologies A/S [2]. It contains a gripper and an embedded 3D imaging device. The 3D imaging device [3] provides the information required by the computer to identify the pose of a part in the bin.

Once picked, there can be a large uncertainty on how the picked part is exactly positioned and orientated within the coordinate system of the gripper. For feeding the object into a manufacturing machine, a large uncertainty of the object pose in the gripper is problematic. The uncertainty is due to the complicated environment around the relevant part as it is lying in the pile, and due to mechanical interaction with the pile or the bin, the part can easily shift during the picking inside

the bin. For that reason additional inspections are required to confirm the exact pose of the object in the gripper and to adjust the gripper destination and pose accordingly when delivering the part.

In this work we present a low-cost 3D imaging approach, that can do this additional identification on-the-fly while the robot moves the part from the bin to the location where it is needed. The setup is low-cost, fast enough so that the robot does not need to slow down and precise enough so that the robot can feed the part reliably into the subsequent assembly process. It requires only minor constraints on the setup of the bin-picking system.

Generally, commercial 3D cameras ([4], [5]) often suffer from limited depth resolutions which results in data with noisy surface and edge representation. This often compromises the quality of the identification of a part when comparing to the CAD model. Further, the point density in the projected light pattern usually needs to be quite high in order to determine position of e.g. edges precisely, and this could compromise the precision of the measurements of orientation and position of the entire part.

In this work, we will introduce an *in-flight 3D scanner* which acquires a 3D image of a part, while it is transported through an *inspection volume* located in front of the scanner. The 3D scanner uses the motion of the robot arm to scan the part through a single, static *light sheet* (Fig. 2). The camera acquires series of images, which provide a high resolution and high precision 3D point cloud of both the part and the gripper, simultaneously. For that reason, the camera and the robot arm do not need to be synchronized perfectly. The pattern of the intersected light sheet is a single and mostly continuous line of light. It allows to define the surface and the edges of the parts with a precision per data point, that is significantly higher than it is the case for the data points in the traditional point cloud methods of common 3D cameras. As an additional option it is possible to consider only the 3D edge data of the scanned part which allows for a minimal representation of the object, therefore offers the possibility of decreased computations and accelerated process time.

In addition, to the development of the low-cost 3D camera, we have rigorously analyzed the imaging process to get a full understanding of the camera.

The paper is organized as follows; In Section II we will present the setup and the way it performs and the calibration procedure. In Section III, we describe the entire imaging process mathematically. Finally, the quantitative and qualitative evaluation of the simulation is served in Section IV and detailed results are provided.

## II. 3D SCANNER

The entire setup is illustrated in Fig. 2. The system consists of a robot arm, with a LED marker mounted on the gripper, one camera, two mirrors, and a light-pattern projector. A part is picked with the gripper and as the robotic arm passes through the inspection volume of the 3D scanner, a 3D image is acquired.

The light-pattern projector generates a light sheet crossing the inspection volume at an angle with respect to the viewing direction of the camera, as illustrated in Fig. 2. The robotic arm moves the part through the inspection volume, where the light plane intersects the surface of the handled part. The camera is placed such that its field of view includes both the inspection volume and the mirrors. The mirrors are arranged in a way that they allow an LED marker on the gripper to be visible from two directions in all the images acquired while the robotic arm moves the part through the inspection volume. Therefore, every measurement of the part and the LED marker are simultaneously acquired by the same camera. During the post-processing, each image provides 3D measurements of the illuminated profile of the part and the LED marker by triangulation as we shall see below. By assuming that the orientation of the robotic arm is fixed throughout the scanning task, the exact 3D pose of the gripper can be estimated from the 3D position of a single LED marker.

When the position of the gripper is known from image to image, the individual profiles of the part can be stitched together, and finally, the entire point cloud of the object can be determined. Because, one camera acquires all the information available at a given time, there are no additional assumptions required with regard to e.g. maintaining a constant speed or a moving in a strictly linear motion while robotic gripper transport the part through the inspection volume.

When post-processing the images of the handled part, typically, edges can be defined as discontinuities or drop-outs in the intensity of the line. Drop-outs are evaluated by size, and discontinuities are evaluated by range in order to certify whether the discontinuity represents a step in the part surface or an edge on the part. Data points, which represent an edge of the part, are kept and their 3D positions are determined. In Fig. 1a and Fig. 1b the identification of an edge can be seen. The edge is defined as half the relative step height. In Fig. 1c measured edge positions are plotted as a function of positions of the entire part. The standard deviation from the linear fit is less than 30  $\mu\text{m}$ .

Data points, which represent an edge of the part, are kept, their 3D positions are determined by triangulation and again by using the positions of the gripper, the edge points can be stitched together to a strongly reduced point cloud, describing only the visible periphery of the part.

### A. Calibration

When the system is set up, two calibrations are required. First of all the transfer-matrix,  $\mathbf{T}$ , describing the transform of a position in the robot coordinate system,  $\vec{P}_{xyz}$ , to a position in the camera coordinate system,  $\vec{P}_{klm}$ , is determined for all three imaging processes (the imaging process of the part, and the two imaging processes of the LED marker):

$$\vec{P}_{klm} = \mathbf{T} \cdot (\vec{P}_{xyz} - \vec{O}) \quad (1)$$

where  $\vec{O}$  is the transfer vector pointing from the origin of the robot coordinate system to the origin of the camera coordinate system. By acquiring images of the LED marker, while moving the LED marker/gripper inside the inspection volume, both the imaging process of the handled part, and the two imaging processes of LED markers can be calibrated simultaneously. The position of the LED marker in all three field of views are fitted to the physical position in the inspection volume. The least square method in matrix form [6] offers a technique, which can find the best fit for  $\mathbf{T}$  and  $\vec{O}$ , using a very high number of data points in order to minimize the influence of random errors in the calibration process. In this case, the robot is a 3D motorized stage for optical applications, thus we can specify the positions with an accuracy of  $\pm 5 \mu\text{m}$ .

Secondly, the light plane needs to be located and orientated relative to the camera coordinate system. We have:

$$\vec{n} \cdot (\vec{P}_{klm} - \vec{P}_0) = 0 \quad (2)$$

where  $\vec{n}$  is the vector normal to the light plane,  $\vec{P}_{klm}$  is the position vector to be tested and  $\vec{P}_0$  is a known position in the light plane. For this purpose we use a sphere with a known radius – in our case of 28.625 mm. The sphere is mounted in the gripper and images are acquired while the robot/stage moves the sphere around in the inspection volume. By fitting the line images to ellipsoids we can estimate the positions of the center of the sphere in the camera coordinate system. Again, by using the least square method in matrix form and significantly multiplying the number of data points beyond the number of linear equations required for a solution we can find a good fit for the normal to the light plane in the robot coordinate system.

The camera has global shutter and can deliver frames at a rate of 162 frames per seconds. This means that if the robot arm can move the part with a speed of 1 m/s, the 3D imaging method will provide an data point for every 6 mm. The resolution of the camera is of 1920 x 1200 pixels in B/W and is relatively high in order to allow for the three fields of views. The precision relies mostly on the calibration carried out with the sphere ( $\gg 30 \mu\text{m}$ ), and we estimate that

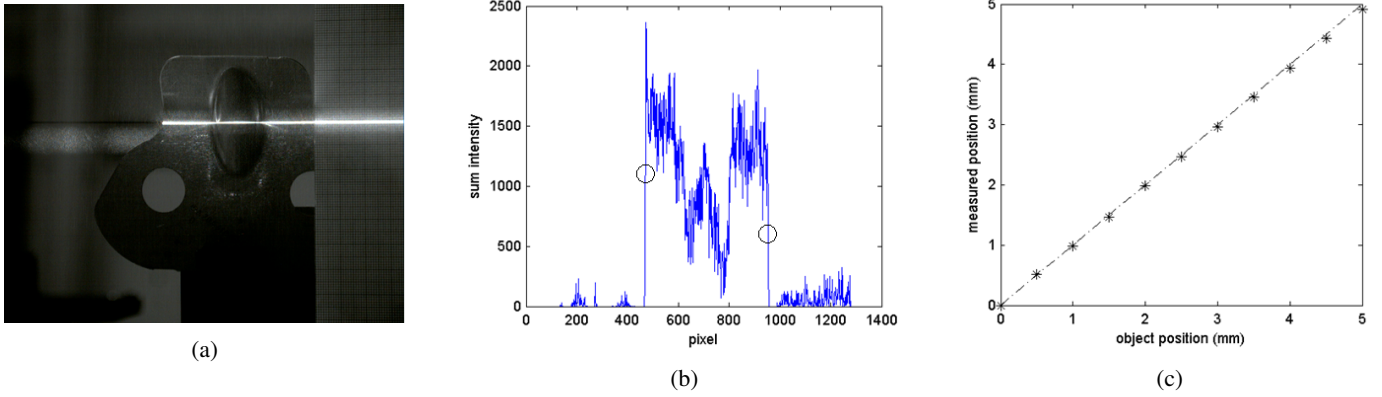


Fig. 1: The edge detection performed on the part (left) is illustrated in the middle figure and the measured position is plotted as a function of part position in the right figure.

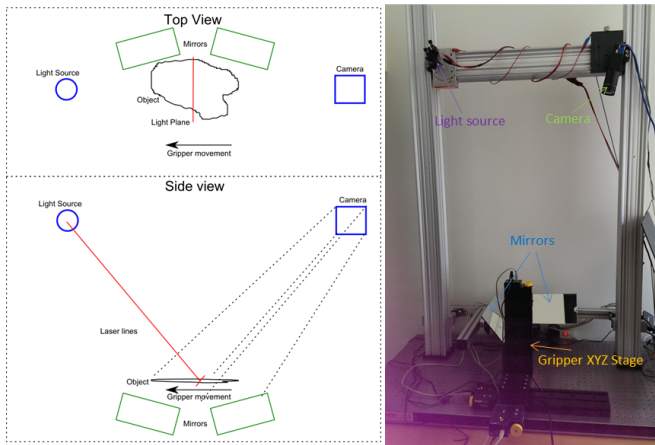


Fig. 2: Schematics (left) and a photograph (right) of the actual setup.

our procedures can determine the position of the sphere by approximately  $100 \mu\text{m}$ .

The images are processed simultaneously while obtaining data to minimize processing time. Each image can be processed individually and the full point cloud can be generated immediately after the last image is taken (after processing of the images is done). When post-processing, the data points, obtained from the images, and their positions must be transformed into the robot coordinate system, therefore, we determine the inverted transfer-matrix  $\mathbf{T}^{-1}$ :

$$\vec{P}_{xyz} = \mathbf{T}^{-1} \cdot (\vec{P}_{klm} + \mathbf{T} \cdot \vec{O}) \quad (3)$$

The final full point cloud and the edge point cloud are illustrated in Fig. 3 for an arbitrary part.

### III. 3D SCANNER SIMULATION

#### A. Virtual Setup

The virtual setup is considered a copy of the actual 3D scanner setup (Fig. 2). Important to be stated is the fact that the

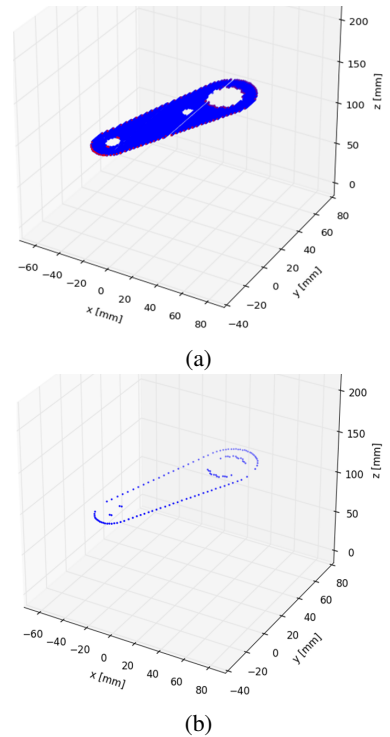


Fig. 3: The figure illustrates the surface 3D point cloud (top), of an arbitrary part, and the 3D edge-point cloud (bottom).

only input parameters to the simulation process is the position of both light source and camera. The parameters which adjust the density of the output point cloud have initially fixed values. However, the latter parameters can be adjusted accordingly to the desired output density and execution time, if necessary.

In order to simulate the light source, a virtual camera was utilized. This camera beams a line on the object that mimics the light line that is emitted by the laser pointer on the actual setup. Any point of this line that lies on the object is considered initially as a point of a temporary point cloud. In the next step, it is checked if these temporary points are

visible from an other virtual camera which acts as the real camera of the 3D scanner. All the temporary points that are clearly visible by the latter virtual camera are considered as final points of object's point cloud. Finally, this procedure is executed multiple times in order to cover the whole model.

The whole area of the object is important to be scanned, thus moving the part in front of the virtual cameras (as it is the case in the real setup) is necessary. Transforming the CAD model, in order to mimic its movement in front of the fixed camera, is computationally expensive. Moving instead the virtual setup (both cameras by keeping their geometric relation unchanged) is a more efficient solution.

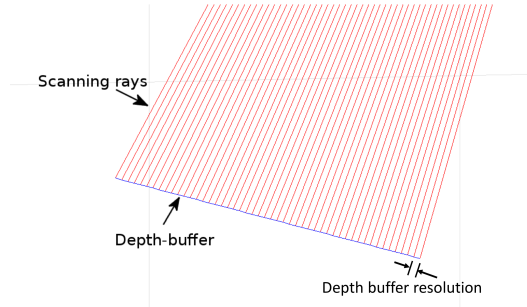


Fig. 4: Depth buffer that imitates the light plane.

### B. Algorithm analysis

As earlier stated, the line that simulates the light source beam is the first step of the simulation. For that exact reason, a depth buffer (or Z-buffer) is constructed. The Z-buffering is a commonly used technique in computer graphics as a way of managing the depth information. It is a way to distinguish whether different objects or parts of a scene are visible from a specified position. More specifically, for each element of the scene the Z-buffer provides a solution for capturing that part of it that is visible and not on the back side with respect to the viewing position. It was initially proposed by [7] and considered as a naive solution, however nowadays depth buffer is a standard and very essential method in computer graphics. In our simulation the buffer is constructed as a line (Fig. 4, Fig. 5) that is perpendicular to the line which starts from the first virtual camera –that mimics the light source– and stops to the origin of setup's coordinate system (Eq. 4). The constructed depth buffer simulates the light plane that is emitted by the light source and is seen as a line on the object (Fig. 1a).

$$\begin{aligned} x &= LP_x - LP_x \cdot t \\ y &= (LP_y + NmSt \cdot StRs) - LP_y \cdot t \\ z &= LP_z - LP_z \cdot t \end{aligned} \quad (4)$$

The transformation of the light source - camera setup is noted in Eq. 4. The  $NmSt$  notation indicates the step number and is multiplied by the  $StRs$  which signifies the chosen step size (see Fig. 7). The step size is highly dependent on the application and is easily adjusted if necessary. Smaller steps allow more details resulting in increased computations and accordingly to

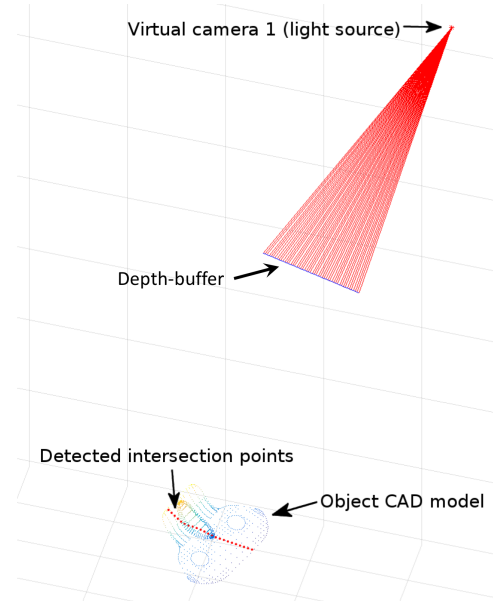


Fig. 5: Intersection points on CAD model.

highly extended execution time.  $LP_x$ ,  $LP_y$ ,  $LP_z$  denote the light source position on x, y and z axis respectively.

The continuous line of the depth buffer theoretically consists of infinite number of points, but a reasonable resolution is adopted (Fig. 4). The higher the resolution, the slower the simulation. Using the point where the first virtual camera is located and each of the selected points on the depth buffer line, shooting rays are constructed. Each one of them emits towards the CAD model of the object (Fig. 4). If the extension of a scanning ray intersects with the object's surface (Fig. 5), the intersection point is considered as a point of the temporarily generated point cloud. The aforementioned method is picked in order to mimic the straight laser line of the physical setup, which scans the object in every step.

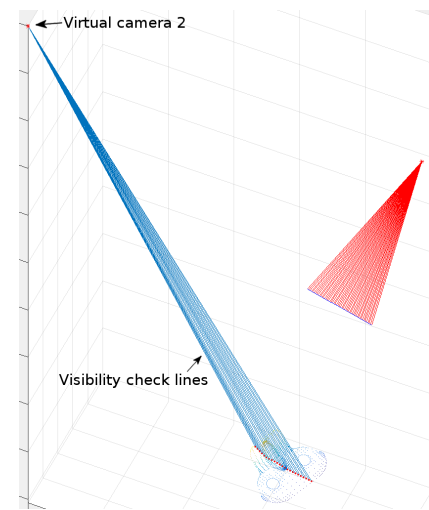


Fig. 6: Visibility check of the detected intersection points.

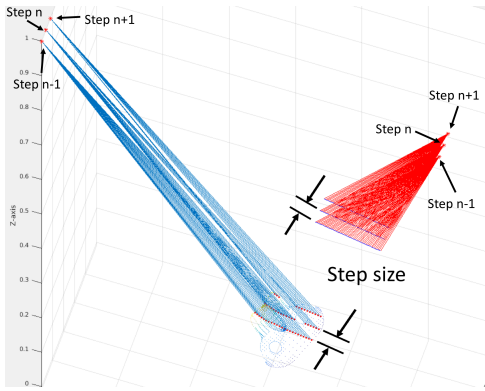


Fig. 7: Multiple scanning iterations with highly increased step size (StRs). The numbering of the scans is illustrated too ( $NmSt$ ).

The visibility check that follows, determines whether these points are indeed visible from the second virtual camera, which simulates the physical camera of the setup. Even though this step increases the computation time, it is rather important. By simulating this camera, we reproduce possible “blind spots” that are generated in the actual setup, due to objects’ cavities and occlusions. All the points from the temporal point cloud that are indeed visible, compose the final point cloud of the object (Fig. 6).

A CAD model is not a solid part as an actual object, thus a ray can possibly intersect with the latter in more than one point, due to the fact that beams pass through it. This phenomenon is taken into consideration throughout the simulation process. Only those points that are visible to the light source side of the object are on the surface towards the camera and are then selected. The back side of the model would not be visible, in the real world, thus any other point except from the closest one is neglected.

Lastly, the described procedure is executed multiple times so that the whole CAD model is fully scanned, and a dense point cloud is generated. In Fig. 7, the simulation of the illuminated object and the corresponding scanning are illustrated for three steps. The selected scanning step ( $StRs$ ) in this specific case is highly increased for the sake of better visibility.

#### IV. SIMULATION EVALUATION

In this section the speed and the accuracy of the implemented camera simulation will be evaluated. Four different objects (Fig. 8) are selected to illustrate the output of the simulation. Their CAD models have a varying number of polygons and are chosen in a way to allow us to test the speed of the algorithm in several load scenarios, i.e. fluctuating part complexity. Initially, quantitative tests regarding the execution time of the algorithm with respect to the resolution of the generated point cloud were conducted.

During testing, in order to evaluate the algorithm under various load scenarios, the parameters were adjusted to generate point clouds with a variety of densities. The aforementioned parameters refer to the consistency of the scanning lines, that

cover the whole object, and the number of the shouting beams per scanning line. For that reason, 18 different parameters’ configurations were applied to the 4 selected objects. The testing process was handled using an Intel i7-6700K processor. The simulation algorithm is able to take advantage of all cores of the CPU simultaneously.

The plots that depict the results of the testing procedure, indicate a linear relation between the density of the generated point cloud versus the runtime (Fig. 10). It is noted that for approximately the same number of points of the generated point cloud the time varies on different object. This comes as a result of the diversity on the number of polygons to describe each object. For example, for Model 2, which has 128590 polygons, the simulation requires the highest runtime even for the least dense output, comparing to all the other parts. The high number of polygon and vertices lead to highly increased computations during the simulation. On the other hand, for Model 3, which consists of 4230 polygons, even for the most dense output, less than a second is required.

Additionally, qualitative tests concerning the accuracy of the output indicate that the simulation depicts accurately the parts. Due to the fact that conducting quantitative tests in order to evaluate the quality, would lead to evaluation of the selected method that measures the accuracy of the generated point cloud, it was decided to visually assess the result. In Fig. 9, the generated point cloud (green colored points) alongside the pure vertices of the CAD model (red colored points) are depicted for 2 of the 4 selected parts. It is clearly visible that the generated point cloud fits exactly on the CAD model of the parts. Also, no points from the back side of the objects are generated. Only points that are visible, to the camera position, are projected.

#### V. DISCUSSION

At present, the camera is triggered for a measurement by inspecting the images continuously for the presence of the LED marker. This arrangement limits the frame rate to 16 frames per second, and the speed of the robot arm to 50 mm/s in order to get edge data points for every 3 mm. In the future work, the trigger will be activated by optoelectronic hardware, and frames will be transferred to the computer at full speed (162 fr/s).

By combining the edge point cloud with information of the initial inspection (the full point cloud) we would like to demonstrate that the method is still a flexible solution in terms of addressing many different types of parts.

#### VI. CONCLUSION

A precise method for 3D imaging was presented. The proposed system can generate the surface and the edge 3D point cloud of an object with almost no noise (comparing to well-known 3D cameras) and high accuracy.

The simulation of the process applied on the CAD model of the part, outputs also a very detailed and accurate 3D point cloud, by imitating the real setup. Only the position of the

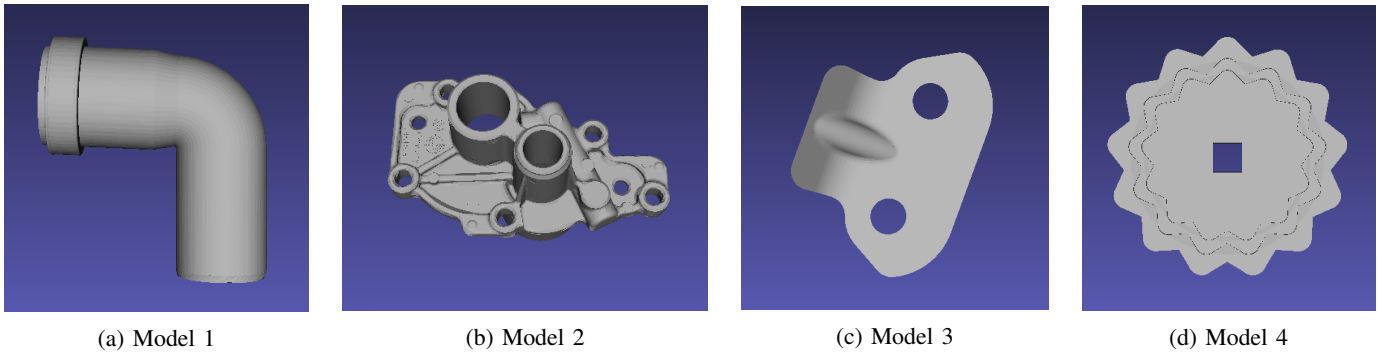


Fig. 8: Objects' CAD models

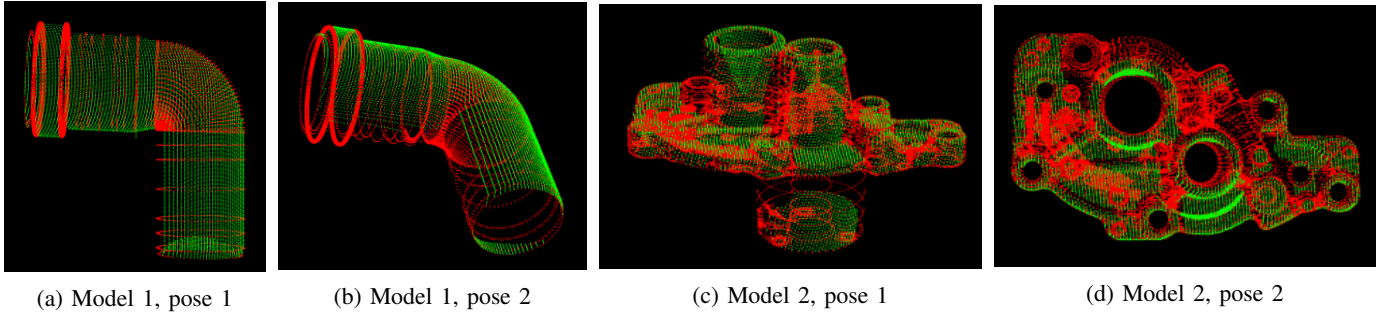


Fig. 9: Qualitative test results

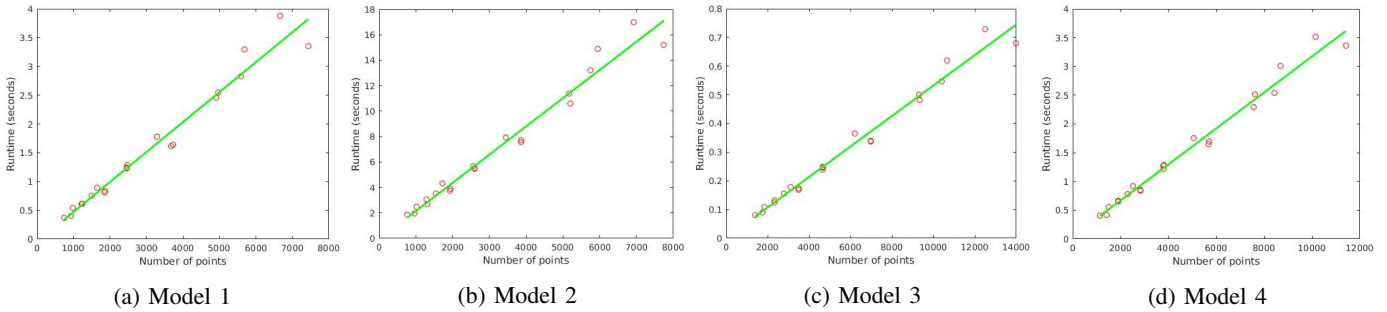


Fig. 10: Quantitative tests results

camera and the light source are necessary input parameters in order to have a precise imitation of the system.

The actual 3D point cloud and the one generated by the simulation, will let calculation of misplacement of the part inside the gripper. The resulted pose refinement will grand highly improved accuracy during the placing task, hence boosted efficiency.

#### ACKNOWLEDGMENT

This work has been supported by the Danish National Advanced Technology Foundation, (InnovationsFonden), under the project Mini-Picker, Grant #115-2013-3. The work is carried out in collaboration with Scape Technologies A/S, Universal Robots, Blue Ocean Robotics.

#### REFERENCES

- [1] K. Rahardja and A. Kosaka, "Vision-based bin-picking: Recognition and localization of multiple complex objects using simple visual cues," in *Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, vol. 3. IEEE, 1996, pp. 1448–1457.
- [2] "Scape Technologies A/S," <http://www.scapetechnologies.com/en/>, accessed: 2016-05-30.
- [3] S. Baby, K. Simonsen, I. Balselv, V. Krger, and R. Dencker, *3D Scanning of Object Surfaces Using Structured Light and a Single Camera Image*. United States: IEEE, 2011, pp. 151–156.
- [4] K. Khoshelham, "Accuracy analysis of kinect depth data," in *ISPRS workshop laser scanning*, vol. 38, no. 5, 2011, p. W12.
- [5] H. Haggag, M. Hossny, D. Filippidis, D. Creighton, S. Nahavandi, and V. Puri, "Measuring depth accuracy in rgbd cameras," in *Signal Processing and Communication Systems (ICSPCS), 2013 7th International Conference on*. IEEE, 2013, pp. 1–7.
- [6] A. Björck, *Numerical methods for least squares problems*. Siam, 1996.
- [7] I. E. Sutherland, R. F. Sproull, and R. A. Schumacker, "A characterization of ten hidden-surface algorithms," *ACM Computing Surveys (CSUR)*, vol. 6, no. 1, pp. 1–55, 1974.