RESEARCH PAPERS
FACULTY OF MATERIALS SCIENCE AND TECHNOLOGY IN TRNAVA
SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA

2015 Volume 23, Special Number

IMPROVING COGNITIVE SKILLS OF THE INDUSTRIAL ROBOT

Pavol BEZÁK

Ing. Pavol Bezák, PhD., Slovak University of Technology in Bratislava, Faculty of Materials Science and Technology in Trnava, Advanced Technologies Research Institute, Bottova 25, 917 24 Trnava, Slovakia
e-mail: pavol.bezak@stuba.sk

Abstract

At present, there are plenty of industrial robots that are programmed to do the same repetitive task all the time. Industrial robots doing such kind of job are not able to understand whether the action is correct, effective or good. Object detection, manipulation and grasping is challenging due to the hand and object modeling uncertainties, unknown contact type and object stiffness properties. In this paper, the proposal of an intelligent humanoid hand object detection and grasping model is presented assuming that the object properties are known. The control is simulated in the Matlab Simulink/ SimMechanics, Neural Network Toolbox and Computer Vision System Toolbox.

Key words

industrial robot, cognitive skills, object detection, pose estimation

INTRODUCTION

Three-dimensional pose estimation is one of the core technologies of object grasping and manipulation. Three-dimensional pose estimation has been extensively searched in the robotics random bin picking problems. The most common pose estimation adopts an algorithm framework shown in Fig. 1. Three-dimensional objects are perceived by one or a multiple vision sensor and two-dimensional images and three-dimensional information are extracted thereof. After the sensor captures step, the captured data first undergoes a pre-processing step that includes noise filtering, data structure reorganizing, and object data segmentation. Then the pre-processed data is compared with the object's template pose in a database that is represented by both features and point clouds. The comparison will yield a pose estimation result that represents both translation and rotation of the object with respect to a world coordinate frame.

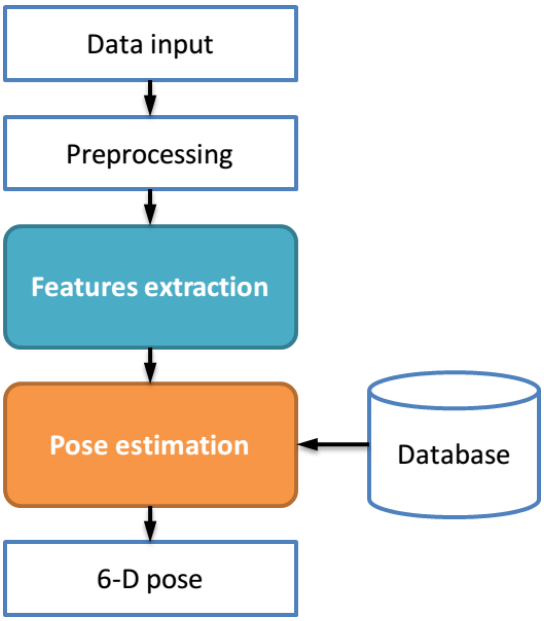

Fig. 1 *An architecture framework of a pose estimation algorithm*

The pose of a three-dimensional object can be represented by its translation (position) and rotation (orientation) with respect to a reference coordinate system frame, such as the common world frame. The position information of an object can be represented by its centroid using (x, y, z) with respect to a reference coordinate system frame as shown in Fig. 2(a). The orientation information of an object can be represented by the three-rotation angle with respect to a reference coordinate system frame with each axis orthogonal to one another. Euler angle is the most commonly used orientation representation where the order of rotation axis affects the final orientation. Fig. 2(b) shows the Euler angle of the objects orientation in the X-Y-Z order. That means the object was first rotated from its predefined position along the x-axis; the rotation angle is denoted as θ (pitch). Then the object was rotated along the y-axis with an angle of ϕ (yaw). Finally, the object was rotated along the z-axis with an angle of ψ(roll). As a result, the orientation of the object in terms of Euler angle is (θ, ϕ, ψ).

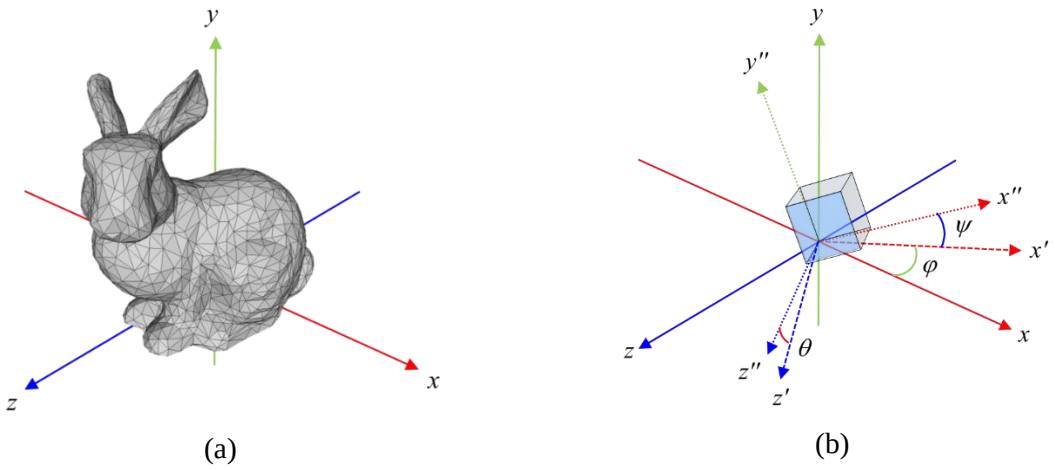

Fig. 2 *Definition and representation of a coordinate system for a 3D object (a) Right-hand Euclidean coordinate system for position, (b) Euler angle representation for orientation*

Pose estimation is a common and necessary technique in robotics random bin picking (3-7), object localization (8), object cognition and learning (9), and 3-dimensional environment reconstruction. Traditional 2-dimensional vision approach (8, 11, 12) often suffers from difficult object definition, view point occlusion, insufficient information. These limitations make estimating a 3-dimensional pose of an object from 2-dimension image unreliable. However, as 3-dimensional optical metrology and devices improve their performance along time, depth information that provides additional reliable z-direction information can be easily obtained. Therefore, the methods estimating the pose of an objects based on 3-dimensional information have been extensively searched.

1. OBJECT DETECTION AND MANIPULATION

One of the most commonly adopted approach in 3-dimensional pose estimation is the correspondence framework (13). Assume there are two identical objects with different poses and it is possible to find a pair of corresponding points on the surface of the two objects. By using such corresponding point pairs, one can find the spatial transformation, which consists of translation and rotation between two identical objects. With a given object coordinate system, the transformation is the pose of the object with respect to this given coordinate system. Finding corresponding points relies on using 3-dimensional feature points that reside on the surface of objects. These feature points have unique descriptor and characteristics that enable them to discriminate different feature points. Ideal implementation would be to match the extracted feature points directly and find the transformation using minimal pairs of corresponding points. However, in real case, corresponding point pairs often include a portion of point pairs being ambiguous. That means that some point pairs may not really have a corresponding point pair due to mismatch. This often results in incorrect transformation In addition, the correctness of a feature point also influences the accuracy of pose estimation result. This is a consequence of the feature extraction method itself. To avoid these issues, one practical implementation often uses more corresponding point pairs that would result in an over determined system in finding the correct transform. Then techniques such as sample and consensus-based techniques or least-square matching can be applied to reduce the impact of noises, outliers and mismatches. Finally, most pose estimation adopted iterative closest point to further improve and fine-tune the pose estimation result.

Two major factors affecting the performance of corresponding framework are feature point representation and the corresponding point pair search accuracy. Feature points that are used to represent the surface of a 3-dimensional object can be categorized into global structural feature and local feature. Kahdan et al. proposed a global spherical harmonic feature (15) that considers the overall shape of an object. However, this approach is not suitable for industrial random bin picking application because often only partial 3-dimensonal surface information is available.

Another approach is to consider only the geometrical relation, such as position or normal vector information, among reference points in a spatial neighbourhood on the surface of an object. Early methods exploited mesh-based representation of 3-dimensional surface. These methods include Stein's splash feature (16) and Johnson et al.'s spin-image (17). Lately, a new representation of 3-dimensional surface that originated from robotics vision turned to using point cloud data representation. Point cloud data can be extracted from various 3-dimensional vision sensors such as time-of-flight, active stereo, and structured light sensors. One of the most commonly reference feature that is based on point cloud representation is the Point Feature Histogram (PFH) (18) and Fast Point Feature Histogram (FPFH) (19) proposed by Rusu et al.

Other than methods of extracting 3-dimensions feature point, extracting 2-D feature based on multiple view still plays an important role in checking correspondence. In this type of feature extraction, it is possible to establish a database that consists of all the views of an object and then find the correspondence using 2-D image feature matching. Liu et al. (7) used the edge feature in different view images of an object to establish a template database, and applied it to find the pose of an object from a pile of randomly stacked objects. However, this approach is often of limited effectiveness on objects of slightly flat shape.

The reliability of pose estimation result directly affects the picking success rate in industrial picking/grasping operations. Objects in industrial application may often reside in a bin randomly stacked together or in a tray not overlapped. When an object is in a bin or a tray, it is often partially occluded by either other objects or the bin/tray. Therefore, it is crucial for pose estimation to be able to work with partial fragmented input point cloud or mesh data. Böhnke and A. Gottscheber proposed a progressive mesh (6) to represent the surface of an object. They also proposed a modified ICP method to perform progressive mesh matching. Their method resulted in the good pose estimation accuracy and could achieve a processing throughput of 1 Hz. Skotheim et al. later proposed a 3DMaMa (3) method that used point cloud data as input. Their method is based on searching flat regions on an object as features for correspondence matching. Then they used a RANSAC-based method called DARCES (20) to perform pose estimation. Their experimental results showed good pose estimation accuracy and achieved a processing throughput of 4 Hz.

2. CONCEPT OF PROPOSED SOLUTION

The aim is to perform a reliable 3-D pose estimation on objects available in potential industrial applications. Existing approach often relied on using 3-D point features as cues for pose registration. However, such method may suffer in situation when a target object failed to have sufficient number of feature points. To improve such limitation, a pose estimation algorithm based on cocktail-feature extraction and weighted-voting scheme will be developed. The main concept is to use different feature, including point, edge, shape, plane, or discriminative geometrical structure feature, to boost up the amount of useful feature information. On the one hand, these various features may strengthen pose inference reliability due to additional information, thus reducing incorrect pose estimation. On the other hand, these features may compensate one another when an object failed to generate sufficient feature of a certain type. The voting scheme transform feature matching into a hash table look-up operation that could avoid exponential growth of computation speed.

After kinematic analysis, the 3D model of simplified (three-fingered) humanoid hand can be created. For this task, the Autodesk Inventor software was used. In the CAD model, we did not assume actuators. The CAD model (Figure 3) served us only for getting a better idea and for observation of possible motions and operations with hand.

It is possible to import the model from Autodesk Inventor to Matlab Simmechanics to further analysis. We used the tool smlink linkinv. After successful import, we gained also Simulink model of the model of the hand.

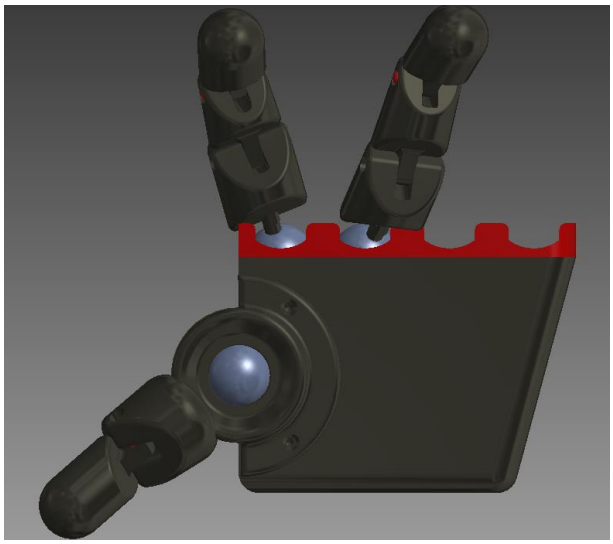

Fig. 3 *CAD model of humanoid robot hand*

In the Matlab Simulink model, we can see parts of the simulated robotic hand after the import from Autodesk Inventor (Figure 3). The model had to be edited to add the required functionality.

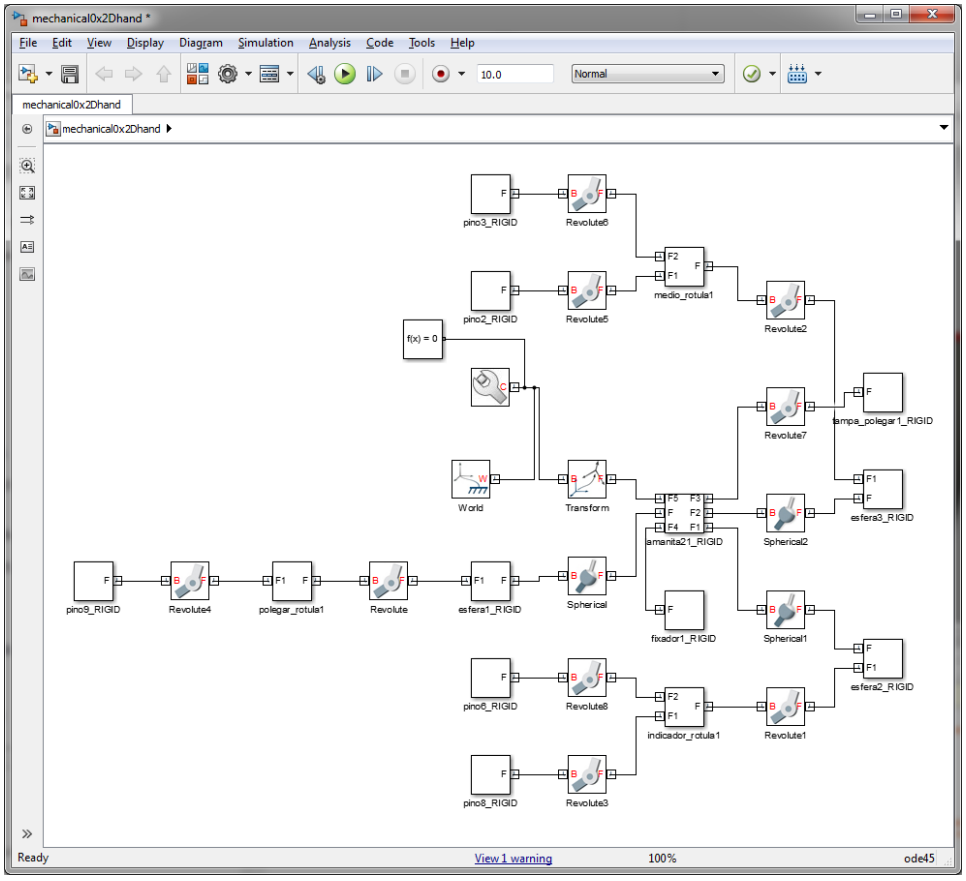

Fig. 4 *Imported Matlab Simulink model of humanoid robot hand*

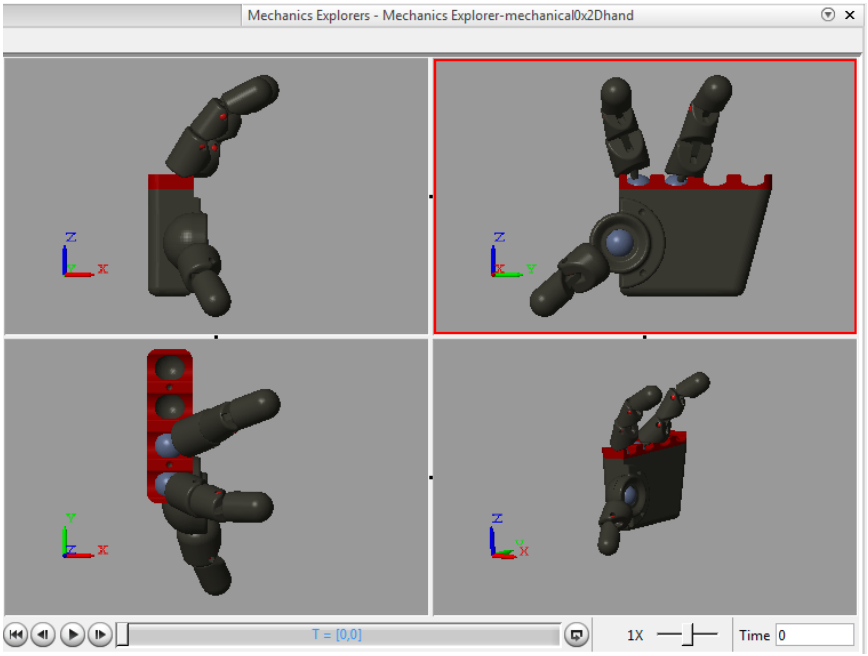

Fig. 5 *3D model of humanoid robot hand in Matlab Mechanics Explorer*

During the execution of a manipulation task, unexpected or pre-planned interactions of the object with the environment may occur. The main goal of the control is to ensure that the robotic system does not lose the object and that the exchanged forces remain limited. A crucial point is the control of the contact forces between the object and the fingers. Keeping the contact forces within a certain range is important for several reasons. On the one hand, these forces must be sufficiently high to guarantee the satisfaction of the friction cone constraints; on the other hand, contact forces cannot be too high to avoid saturation of the motors and waste of energy, as well as to preserve the materials. In this context, the presence of the force sensors at the fingertips plays an important role for the control of both the arm and the hand.

Visual object detection is the most important step in robotic grasping. Many methods have been reported so far. As almost all the object recognition methods rely heavily on the accuracy of foreground object detection, efficient and reliable methods are necessary.

In machine learning and related fields, artificial neural networks (ANNs) are computational models inspired by an animal central nervous systems (in particular the brain) which is capable of machine learning as well as pattern recognition. Artificial neural networks (Fig. 6) are generally presented as systems of interconnected "neurons" which can compute values from inputs.

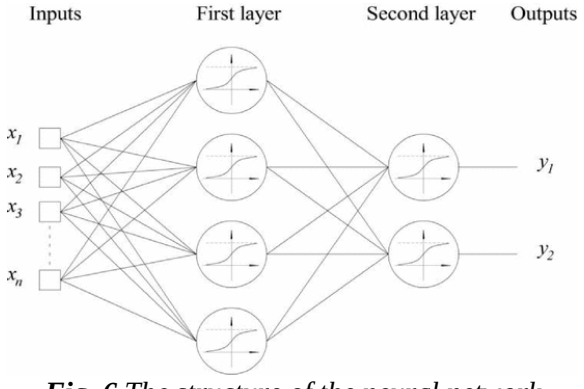

Fig. 6 *The structure of the neural network*

In machine learning, a deep belief network (DBN) is a generative graphical model, or alternatively a type of deep neural network, composed of multiple layers of latent variables ("hidden units"), with connections between the layers but not between units within each layer. Deep learning is a set of algorithms in machine learning that attempts to model high-level abstractions in data by using model architectures composed of multiple non-linear transformations. Various deep learning architectures such as deep neural networks, convolutional deep neural networks, and deep belief networks have been applied to fields like computer vision, automatic speech recognition, natural language processing, and music/audio signal recognition where they have been shown to produce state-of-the-art results on various tasks.

A Convolutional neural networks (CNN) is composed of one or more convolutional layers with fully connected layers (matching those in typical artificial neural networks) on top. It also uses tied weights and pooling layers. This architecture allows CNNs to take advantage of the 2D structure of input data. In comparison with other deep architectures, convolutional neural networks are starting to show superior results in both image and speech applications. They can also be trained with standard back propagation. CNNs are easier to train than other regular, deep, feed-forward neural networks and have many fewer parameters to estimate, making them a highly attractive architecture to use.

Comparing with the traditional object recognition based on the deep learning model, we focused on the object pose estimation including object recognition. Deep learning methods have the capability of recognizing or predicting a large set of patterns by learning sparse features of small set of patterns. With this advantage, we can use a small set of poses to train the deep learning model, and then predict a large set of poses with the model (25).

For general objection recognition and image classification tasks, variants of Convolutional Neural Networks (CNNs) have emerged as robust supervised feature learning and classification tools, especially when combined with max-pooling (MPCNN) (Fig. 7) (26).

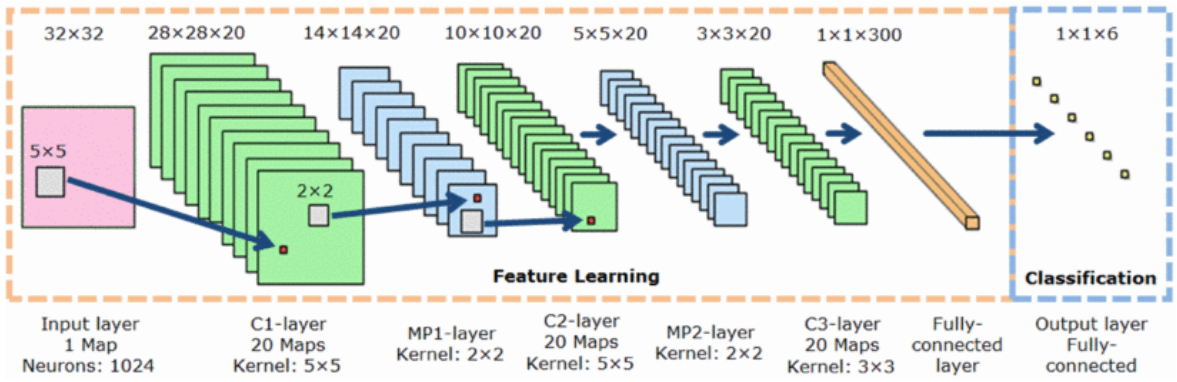

Fig. 7 *MPCNN architecture using alternating convolutional and max-pooling layers (26)*

MPCNNs include convolutional layers and subsampling layers. MPCNNs are different according to the variety of training and realization of convolutional and subsampling layers.

Convolutional layer

The parameters of the convolutional layer are: the number of maps, the size of the maps and kernel sizes. Each layer (L) includes maps (M). A kernel (K) of size is shifted over the valid region of the input image. Each map in Layer Ln is connected to all maps in layer Ln-1. Neurons of a given map share their weights but have different input fields (26).

Max-pooling layer
The output of the max-pooling layer is determined by the maximum activation over non-overlapping rectangular regions. Max-pooling improves generalization performance (26).

Classification layer
To complete the MPCNN, a shallow Multi-layer Perceptron (MLP) is used. The output layer has one neuron per class in the classification task (26).

3. EXPERIMENT

Object Detection
The system consists of the Matlab SimMechanics model of robotic hand scene with objects and simulated camera. The virtual objects are in the reach of the vision system and are recognized through the Matlab Computer Vision Toolbox with implemented model of MPCNN.
The input images come from RGBD camera data that, opposed to simple 2D image data, has been shown to significantly improve the grasp detection results.
Visual object detection is the first key action for robotic grasping. It is needed to use or develop reliable methods that effectively recognize foreground objects that are present at the input to the vision system and detect the objects from the background.
One of the possible approaches is to use sparse coding or K-means clustering. The first step is to build dictionary of objects. Clustering enables us to separate groups of color components of the images with background and objects. Each component has a location in feature space and it is important to find partitions such that components within each cluster are as close to each other as possible and as far from components in other clusters as possible (25).

Object Recognition and Pose Estimation
Using the simulation, we implemented object pose estimation and pose estimation using the methods of deep learning, which have the capability of recognizing or predicting large set of patterns by learning sparse features of small set of patterns. With this advantage, we can use a small set of poses to train the deep learning model, and then predict a large set of poses with the model (25).

Robotic Grasping
This stage presents the system that changes the position and orientation of the gripper in order to grasp the objects.

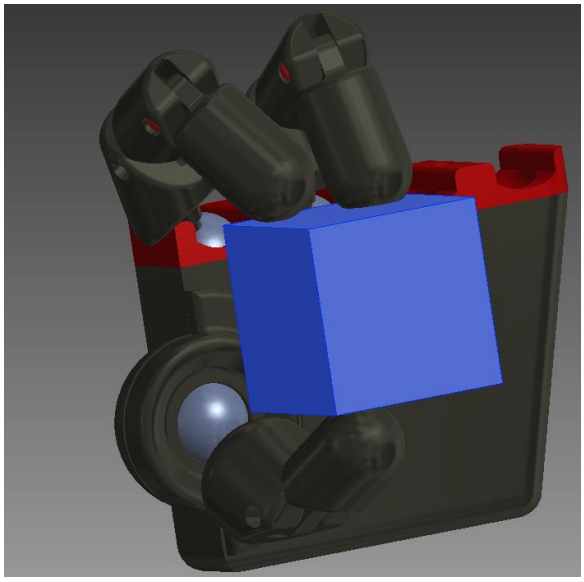

Fig. 8 *3D model of humanoid robot hand grasping the object*

CONCLUSION

The developed models described in the sections above were implemented in selected Matlab Toolboxes (Computer Vision System Toolbox, Deep Learning Toolbox and SimMechanics).

This paper represents a simulated model of a multi-fingered robotic hand for grasping tasks. The model includes kinematics, dynamics, object representation and contact modelling. The contact model determinates the forces that are applied to the object by the robotic hand. The object detection, object recognition and robotic hand pose estimation are based on Max-pooling Convolutional Neural Networks – one of the most popular deep learning models. Using deep learning enables to avoid hand-engineering features, learning them instead.

References:

1. OpenCV: http://opencv.org/, (Accessed: Apr. 24, 2013).
2. Point Cloud Library (PCL), http://www.pointclouds.org/, (Accessed: Apr. 24, 2013).
3. O. SKOTHEIM, J. T. THIELEMANN, A. BERGE, AND A. SOMMERFELT. 2010. Robust 3D object localization and pose estimation for random bin picking with the 3DMaMa algorithm. *Instrumentation*.
4. R. BLOSS. 2006. Smart robot that picks parts from bins. *Assembly Automation*, Vol. 26, No. 4, pp. 279–282.
5. K. BOEHNKE. 2007. Object localization in range data for robotic bin picking. In: *Automation Science and Engineering*, 2007. CASE 2007. *IEEE International Conference on*, 2007, pp. 572–577.
6. K. BÖHNKE and A. GOTTSCHEBER. 2010. Fast Object Registration and Robotic Bin Picking. In*: Research and Education in Robotics-EUROBOT* 2009, pp. 23–37.
7. M.-Y. LIU, O. TUZEL, A. VEERARAGHAVAN, Y. TAGUCHI, T. K. MARKS and R. CHELLAPPA. 2012. Fast object localization and pose estimation in heavy clutter for robotic bin picking. *The International Journal of Robotics Research*, Vol. 31, No. 8, pp. 951–973.
8. S. SAVARESE and L. FEI-FEI. 2007. 3D generic object categorization, localization and pose estimation. In: *IEEE 11th International Conference on Computer Vision*.
9. J. STURM, K. KONOLIGE, C. STACHNISS, and W. BURGARD. 2010. 3D Pose Estimation, Tracking and Model Learning of Articulated Objects from Dense Depth Video using Projected Texture Stereo. In: *Proc. of the Workshop RGB-D: Advanced Reasoning with Depth Cameras at Robotics: Science and Systems (RSS)*.
10. S. MAY, D. DROESCHEL, D. HOLZ, C. WIESEN and S. FUCHS. 2008. 3D pose estimation and mapping with time-of-flight cameras. In: *International Conference on Intelligent Robots and Systems (IROS)*, 3D Mapping workshop. Nice, France.
11. V. LEPETIT, J. PILET, and P. FUA. 2004. Point matching as a classification problem for fast and robust object pose estimation. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2004*, Vol. 2, pp. 244–250.
12. M. VILLAMIZAR et al. 2011. Efficient 3D Object Detection using Multiple Pose-Specific Classifiers. In: *Proceedings British Machine Vision Conference (BMVC).*
13. B. M. PLANITZ, A. J. MAEDER, and J. A. WILLIAMS. 2005. The correspondence framework for 3D surface matching algorithms. *Computer Vision and Image Understanding*, Vol. 97, No. 3, pp. 347–383.
14. P. J. BESL and H. D. McKAY. 1992. A method for registration of 3-D shapes. *IEEE Trans.*

on Pattern Analysis and Machine Intelligence, Vol. 14, pp. 239–256.
15. M. KAZHDAN and T. FUNKHOUSER. 2002. Harmonic 3D shape matching. *Int'l* Conf. on Computer Graphics and Interactive Techniques, p. 191.
16. C. DORAI and A. K. JAIN. 1997. COSMOS-A representation scheme for 3d free-form object. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 10, pp. 1115–1130.
17. A. E. JOHNSON and M. HEBERT. 1999. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 5, pp. 433–449.
18. R. B. RUSU, Z. C. MARTON, N. BLODOW, and M. BEETZ. 2008. Learning informative point classes for the acquisition of object model maps. In *2008 10th International Conference on Control, Automation, Robotics and Vision*, pp. 643–650.
19. R. B. RUSU, A. HOLZBACH, N. BLODOW, and M. BEETZ. 2009. Fast geometric point labeling using conditional random fields. In: 2009 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 7–12.
20. C.-S. CHEN, Y.-P. HUNG, and J.-B. CHENG. 1999. RANSAC-based DARCES: A New Approach to Fast Automatic Registration of Partially Overlapping Range Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 11, pp. 1229–1234.
21. J. PAPON, A. ABRAMOV, M. SCHOELER, and F. WORGOTTER. 2013. Voxel Cloud Connectivity Segmentation - Supervoxels for Point Clouds. *IEEE Conference on CVPR 2013*. Portland, Oregon.
22. R. O. TSAI and R. K. LENZ, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," IEEE Transactions on Robotics and Automation, Vol. 5, No. 3, 1989.
23. J. K. LEE, K. KIM, Y. LEE, and T. JEONG. 2011. Simultaneous Intrinsic and Extrinsic Parameter Identification of a Hand-Mounted Laser-Vision Sensor. *Sensor*, Vol. 11, pp. 8751–8768.
24. Q. ZHANG and R. PLESS. 2004. Extrinsic Calibration of a Camera and Laser Range Finder (improves camera calibration). *International conference on Intelligent Robots and Systems*, pp. 2302–2306. Sendia, Japan. G.-Q. WEI and G. HIRZINGER. 1998. Active Self-Calibration of Hand-Mounted Laser Range Finders. *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 3, pp. 493–497.
25. Y. JINCHENG, K. WENG, G. LIANG, G. XIE. 2013. A vision-based robotic grasping system using deep learning for 3D object recognition and pose estimation. *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1175-1180.
26. J. NAGI, F. DUCATELLE, G.A. DI CARO, D. CIRESAN, U. MEIER, A. GIUSTI, F. NAGI, J. SCHMIDHUBER, L.M. GAMBARDELLA. 2011. Max-pooling convolutional neural networks for vision-based hand gesture recognition. IEEE *International Conference on Signal and Image Processing Applications (ICSIPA)*, pp. 342-347.

Reviewers:

doc. Ing. Michal Kopček, PhD.
Ing. Robert Halenár, PhD.