Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 81 (2019) 316–321

www.elsevier.com/locate/procedia

52nd CIRP Conference on Manufacturing Systems

Determining stable equilibria of spatial objects and validating the results with drop simulation

Márk Fekula[ab]*, Gergely Horváth[ab]

[a]*EPIC Center of Excellence, Institute for Computer Science and Control, Hungarian Academy of Sciences, Kende st. 13-17., Budapest 1111, Hungary*
[b] *Department of Manufacturing Science and Engineering, Budapest University of Technology and Economics, Műegyetem rkp. 3., Budapest 1111, Hungary*

* Corresponding author. Tel.: +36-1-279-6262; E-mail address: mark.fekula@sztaki.mta.hu

Abstract

Robotic pick and place processes require the workpieces to be prepositioned and aligned to a fixed pose. This ensures the accurate grasping, but lacks flexibility and efficiency. Camera based solutions are available for simple workpieces, but no simple solution exists for complex ones. We offer a universal software, that determines the stable equilibrium poses of the workpiece from 3D model, and renders the top view image of it. This image can be compared to one captured by a single camera setup to determine the workpiece position and orientation in realtime for robotic application. The results are validated applying drop test simulation.

Keywords: Flexible manufacturing system (FMS); Simulation; Equilibrium Poses

1. Introduction

Robotic workcells are used widely in all major industries. One common use of these robots is pick and place process on assembly lines, which relies on highly constrained workpiece poses. A frequent problem is locating objects on the line. There are applications where the pose of the target object is well-defined—e.g. pre-assembled objects—however there are cases where the objects to be grasped can be found in varying positions as stated in [1]. In such situations it would prove highly beneficial if the position and orientation of the workpiece could be precisely calculated even with a relatively inexpensive and simple camera setup. In this article we propose a pose calculating method, with a multitude of benefits, such as simpler part feeder and orienting systems, faster lead times, and lower sensor and processing costs.

Some image processing solutions rely on constant object pose. If position noise can be modelled, for example the one caused by wind outdoors as Juntao et. al. [2], or the pose can be calculated by known features in Jiménez et. al. [3], the grasping accuracy may improve. The position also can be calculated utilizing a depth and an RGB camera after a manual training process seen in [4,5,6], but these methods lack flexibility needed for adapting to a new workobject, and noises that cannot be modeled.

A progressive approach could be the use of Deep Neural Networks (DNNs) for object classification by utilizing an RGB-D or even 3D camera, because well trained networks can take advantage of such fine and complex data. The usability of this approach was tested and evaluated by Giulia et. al. [7], which was promising but not robust enough. Experiments were conducted using depth cameras by Xi et. al. [8] with around 68% accuracy. Also, the training time of these networks can be really high, while they lack flexibility when the dataset is expanded, requiring retraining.

In the following section, we specify the problem. Section 3 covers the theoretical background while Section 4 covers the actual method of our solution. In Section 5 we detail our efforts and experience with the measurement of the restitution of coefficient for our validation. Section 6 describes the validation process, as well as the results of it. Section 7 concludes the paper, and briefly looks on some possible improvements.

10.1016/j.procir.2019.03.055

2. Problem statement

The pose recognition or aligner systems currently in use lack flexibility needed for fast workpiece switching, as well as accuracy regarding the pose of a more complex concave object. Recognition speed is also a main factor, as high response times obstruct the grabbing of a moving object. These solutions may also need an operator dedicated solely to them, but their high price and complexity could also prove to be sub-optimal.

The aim of this paper is to create a method to calculate top-down view images of a workpiece with corresponding transformation matrices in equilibrium poses. We strive to provide a competitive alternative to the currently used methods and surpass their productivity in most aspects. It is assumed, that the 3D triangle mesh model of the workpiece is given.

As most of the workpieces involved in the pick and place processes are uncomplicated parts, consisting of only one material, they can be treated as homogeneous. When we are considering stability in an industrial context, unstable equilibria do not provide essential information, because in a mechanically dynamic environment they never rest in an unstable pose. Therefore, the focus of our research is finding only the stable equilibria.

3. Approach

Currently no complete solution exists for industrial pose identification utilizing the calculation of stable equilibria, therefore our work revolved around finding state of the art methods for the sub tasks and implementing them. This whole compact package is one of the main advantages of the solution.

Before calculating stable equilibrium poses, the necessary and sufficient conditions of stability is reviewed based in particular on [9]. Accordingly, the algorithms for finding stable equilibria can be divided into three classes. The first option is calculating the gradients of the object in a 3D polar coordinate system, as seen in [10]. The second widely used approach is based mostly on geometrical transformations, as it consists of projecting the center of mass of the object to the plane of each facet (triangle face of a convex polyhedron), and checking if it is located inside the triangle, used also by Várkonyi et. al. [11]. Finally, the stable poses can be determined by projecting the surface of the object from its center of mass, to the surface of a sphere, which has a radius of 1. Then by analyzing the gradient fields of two functions which represent geometrical properties, the stable equilibria can be calculated, as well as their capture regions introduced in [12,13].

As stated in Section 1, there are various methods for calculating stable equilibria. For our work we used the second one, because it utilizes graphical transformations, which provide clear, easy to understand process and representation for the stable poses of the object.

The disadvantage of using this method is that it applies a finite graphical model, instead of mathematical functions to describe a 3D object, which may produce false positive stable equilibrium poses (detailed description see in Section 4.).

When working with a highly complex concave body, it is advised to calculate them beforehand using their convex hull, because for a body it is only possible to rest on the ground, with points and edges that intersect the convex hull. As a result, only an object's convex hull is representative when searching for stable poses. Without reference to convexity it is important to note that every real world object has an infinite number of equilibrium points [14]. Also, coarse surfaces can provide a huge number of stable and unstable equilibrium points. This produces micro equilibria, which must be merged, resulting in a single stable or unstable pose [15].

After determining the stable poses of an object, we validate the results with drop test simulations. This requires a physical collision model see [16,17], utilizing either an inelastic or a perfectly inelastic material model.

On the basis of the validation data, the entire solution can be reviewed and optimized. For example, applying pose statistics as in [18,19] could yield a faster process, as it highlights the poses which have the highest probability of resting on the horizontal surface.

4. Method

The algorithm, generating the transformation matrices and the desired images from the 3D triangle mesh model of the workpiece, consists of 9 steps:

1. loading the model to program,
2. calculating the model's convex hull,
3. calculating the center of mass of the convex hull,
4. projecting the center of mass to the plane of each facet,
5. making intersection tests between the projections and the facets of the convex hull,
6. expanding the found equilibrium facet to every other facet that belongs to the same stable pose,
7. calculating transformation matrix for the stable poses,
8. rendering top-down view images of the model in equilibrium poses,
9. exporting images and transformation matrices.

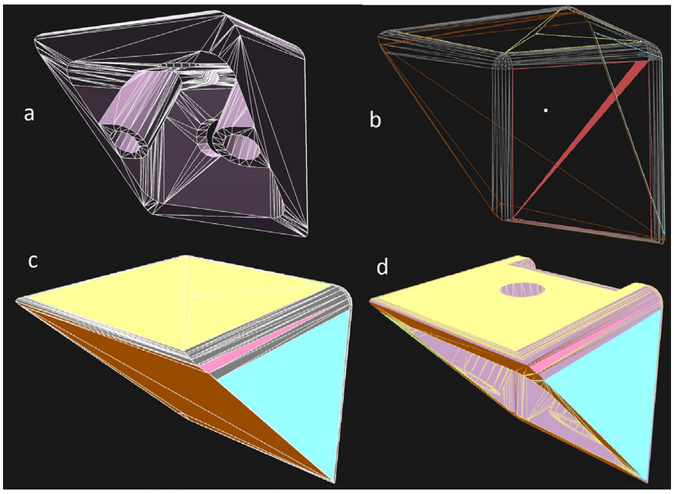

Fig. 1. (a) the loaded model in 3D space; (b) wireframe view of the convex hull, center of mass is represented by a white dot, the edges belonging to different stable poses are color coded, and one equilibrium facet is shown before expanding it; (c) color coded equilibrium faces on the convex hull; (d) color coded poses on the original model

The model loading is done with the ASSIMP [20] library, which supports many 3D formats, and excels the alternatives in ease of use. A model is handled as a triangle mesh. The rendered image of a model can be seen on Fig. 1a. as an example.

After the model has been imported, it must be converted to its convex hull for the calculations. This is done by the CGAL [21] library, which has fast implementations for each computer graphics transformations. Also, the ASSIMP input ensures a triangulated concave surface, which can easily be converted to a triangulated convex one.

The model's center of mass (white dot on Fig. 1b) is determined by discrete volumetric integration. The solution presented in [22] is a fast and accurate implementation in C, for which we created a C++ wrapper and customized it for the actual task.

The projection of the center of mass is done with simple 3D vector equations. The intersection test utilizes a barycentric coordinate system, which greatly simplifies the calculation process. The projected point is connected to the triangulated facet's vertices, this cuts the triangle to 3 pieces, these piece's area ratio produces then the point's 3 barycentric coordinates. Hence a projection is inside the triangle if its barycentric coordinates are not less than 0 and not more than 1. The coordinates can be calculated by solving an equation system (1), where the matrix contains the Descartes coordinates of triangle vertices, $\boldsymbol{\lambda}$ vector contains the barycentric coordinates, and the $\boldsymbol{p}$ vector points to the projected point.

$$\begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{pmatrix} \cdot \boldsymbol{\lambda} = \begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix} \tag{1}$$

If a projection is located inside a facet triangle, then it is a stable equilibrium facet. For example, the one shown on Fig. 1b. The next task is finding all the other facets belonging to this pose. There are two requirements, which a facet must satisfy in order to classify as an additional stable facet of a pose:

- It must have one or more shared vertex with a stable facet.
- Its facet normal must be equal to the equilibrium facet's normal.

The expansion is done by a recursive algorithm, which checks first the neighboring facets, if they have an equal face normal. In the event of a match the algorithm checks the neighbors as well and does so, until no new stable facet is found in a step. The expansion results can be seen on Fig. 1c.

After the stable poses have been located, the next step is to calculate their 4x4 dimensional transformation matrices, which contain the necessary rotation and translation for them to rest on the horizontal XZ plane (as we used OpenGL, Y was the vertical axis). Each triangle is transformed onto the plane vertex by vertex, and the combination of these partial transformations results in the final matrix.

These matrices are then used to produce the resting top-down views for each stable pose, which is rendered using our custom Shader and OpenGL programs, and exported to the desired image format. Also, the matrices are saved in text format.

Table 1. Stable pose calculation results.

Workpiece name	Original vert	Convex hull vert	Time (s)
Metal batten	2610	98	5.26
Concave workpiece	27000	504	17.29
L bracket	5124	175	5.96
90 degree bracket	3828	150	6.16

Tests were conducted on the 3D models of four workpieces, (see Fig. 3). The number of original and convex hull vertices, as well as the running times can be seen on Table 1. Because of the sufficient simplification of the model, the calculations run fast.

5. Experimental measurement of restitution coefficient

Drop tests are based on mechanical collision simulations and can produce the stable poses as a resting face of a drop. Our goal with these tests is not to validate the algorithm introduced in the section before—as stable poses are not correlated—but to test multiple collision models for pose statistics. That is why instead of only using perfectly inelastic models, we simulate inelastic collisions as well.

Inelastic collision can be described with a restitution coefficient, which shows the percentage of kinetic energy stored in the object after collision. Note that this coefficient applies to object pairs that take part in the process, not only to the test object. It is not only a material property, as shape can also affect the results hugely.

There are models to approximate the coefficient of restitution, but the margin of error is high due to the diversity of shapes. The most common approach is to measure the coefficient for every object, that one wants to work with. It can be done using real world drop tests, and evaluating the results based on (2), where h is the maximum bounce height after a collision, and H is the drop height.

$$C_r = \sqrt{\frac{h}{H}} \tag{2}$$

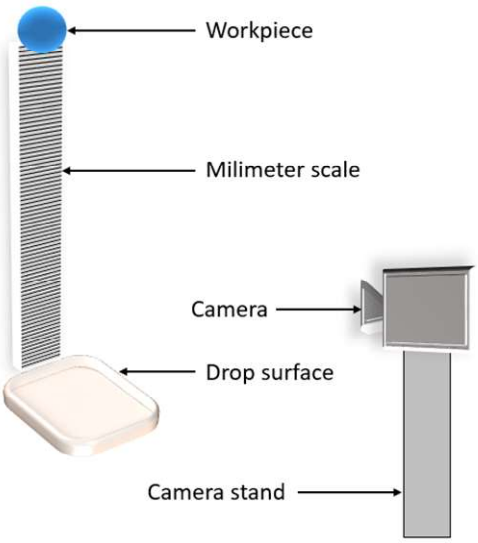

Fig. 2. The measurement setup

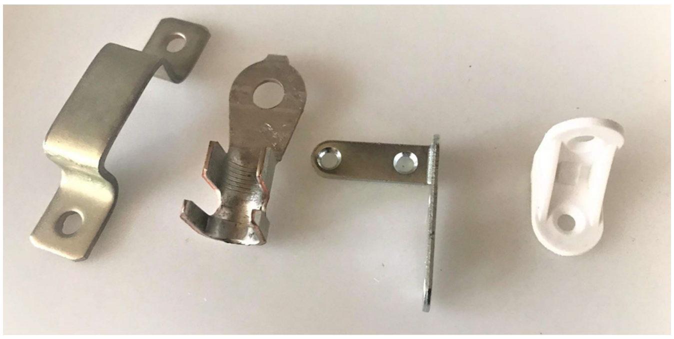

Fig. 3. The workpieces used in drop tests

To record and measure the drop tests, we used an HD camera, with the sampling rate of 30 fps. The drop surface was a 3 mm thick poly-methyl methacrylate (PMMA) sheet. The surface was covered with a 1 mm thick silicone layer. This layout proved to be robust, while adequate dampening is also provided. Behind the drop surface there was a height scale, with millimeter resolution. The camera was positioned to be perpendicular to the scale to minimize distortion. The structure of the measurement setup can be seen on Fig. 2. We conducted tests for 4 objects, see Fig. 3. Each one was dropped 25 times, Table 2. contains the main results.

Table 2. Drop test results.

Workpiece name	Material	Drop height (cm)	C_r (-)
Metal batten	Cu alloy	50	0.288
Concave workpiece	Cu alloy	100	0.207
L bracket	Al alloy	100	0.241
90 degree bracket	ABS polymer	100	0.275

The measured coefficients of restitution showed high deviation values. This was caused by the concave nature of these objects, as landing on a flat surface leads to significantly lower bounce height, than on an acute-angled vertex.

6. Validation of the solution

The software was run on a PC equipped with an Intel Core i7 920 @2.67 GHz CPU, 8 GB of DDR3 @1333 MHz memory, and a NVidia GTS 450 GPU.

The first step is to calculate the stable poses of the models of these workpieces with the algorithm introduced in Section 2. Then drop test simulations are conducted, using $C_r = 0$ (perfectly inelastic) and the measured values. Each simulation consists of 5000 drop tests with the original model. The results are shown on the 3D model (see Fig. 4.), with warmer colors representing more landings on the specific area. These areas can be defined by checking intersection with the horizontal plane point by point, or per each facet. The colors on Fig. 4. are almost identical, because the frequency values are close. The results are also exported to a csv table, containing each equilibrium point and how many times the object landed on them. The two orange rectangles on top of the batten, and the two red ones seen on the bottom, are only edges in real life. This way in these poses the body has a 4-point support, hence these are stable equilibrium points. That is caused by the curved surfaces of the object, so instead of planes only equilibrium edges exist in those poses.

To validate the algorithm, the number and location of the calculated stable poses must be compared to the results of the simulation (see Table 3.). The coefficient of restitution only affects the occurrence frequency of poses, so for the comparison only perfectly inelastic collisions were used.

Table 3. Drop test results.

Workpiece name	Nr calc	Nr sim	Type I	Type II
Metal batten	4	4	0	0
Concave workpiece	12	5	7	0
L bracket	6	4	2	0
90 degree bracket	7	5	2	0

The observed errors can be classified as false positive (Type I error), when the program finds a stable equilibrium in a point, which does not belong to any stable pose, and as false negative (Type II error), when it deems a point unstable although it is stable. There must not be Type II errors, because that means the program cannot fulfil its function. Type I errors are only causing slight inconveniences and their numbers can be reduced.

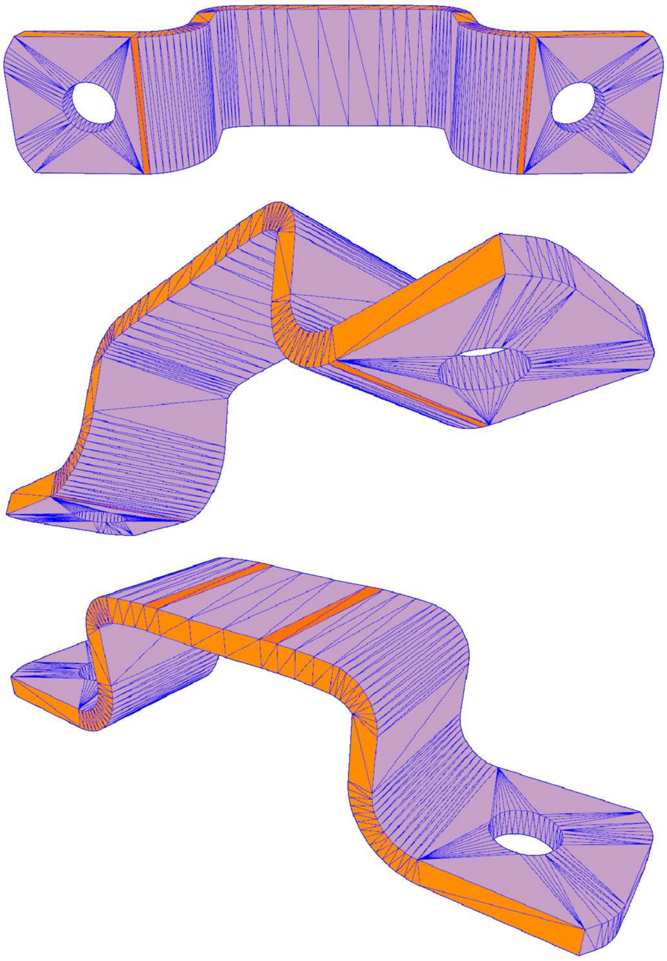

Fig. 4. The stable poses and their occurrence frequencies calculated by the drop test simulation system

The number of Type II errors were 0 for every workpiece, so the algorithm found all the stable poses. On the contrary, Type I errors show up almost every time. This can be traced back to various causes.

The first one is the finite approximation of the curves, which creates seemingly stable 3-point supports (triangle), where in reality only a 2-point support (a line) exists. These false positives tend to be located near the unstable equilibria of the object. Also, as seen in Table 3., more complex and concave models produce more Type I errors. This correlates with containing more curved surfaces.

The second cause is model inaccuracy. For example, the model of the second workpiece was created by 3D scanning, and due to a minimal signal noise, some planes were not flat enough, producing curved surfaces, which—in the end—leads back to the first problem. The convexity of that noise does not matter, as its convex hull will be convex and curved, though a higher noise may cause Type II errors.

Using these error feedbacks various improvements were applied to the original method. The problems caused by 3D scanning could be mitigated with modifying the expansion algorithm. Instead of pure equality, the normal vector of a neighboring facet must be inside a small range. This has to be carefully chosen, because a wide range would only increase the occurrence of Type I errors, but with careful calibration, the small differences on a flat surface can be bridged.

The Type I errors caused by curve approximation can be lowered by using more detailed models, but that also increases the complexity of the calculations and does not provide a full solution. A better approach is to set a minimum limit to the size of the stable surfaces. As these errors tend to be near unstable equilibria and produce smaller equilibrium surfaces, this solution can completely eliminate the rest of Type I errors. If the surface is still comparable to stable ones, their size can be lowered by generating a more detailed model.

Drop test simulations can also find unstable equilibria, provided a reasonable coefficient of restitution has been set, and the number of drops is significantly higher than the 5000 we used in the experiments. This is due to the rarity of landing and staying on an unstable equilibrium vertex, because every move from that pose points to a place with lower kinetic energy. The main reason for also conducting simulations with measured coefficients of restitution is pose statistics.

6.1. Pose statistics

Using pose statistics, the probability of landing on a specific face can be calculated. There are different models for calculating pose statistics, but only an approximation can be made. The main factors of resting probability are the potential energy while resting, the area of the face, and the coefficient of restitution.

The calculator used is only inspired by pose statistics, as implementing a full model would produce even higher processing times. Instead only the potential energy of a pose and the coefficient of restitution is taken into consideration. The program can be adjusted by a variable—based on empirical knowledge—which controls the number of poses to process, ordered by potential energy. This provides a granted performance boost and can also result in ignoring Type I error poses altogether, but it must be noted that if not correctly handled, they can also take the spots reserved for the low energy stable equilibria.

6.2. Discussion

After evaluating the equilibrium calculation algorithm, we concluded that in case of highly complex models lacking curves, it finds all stable equilibria correctly. However due to the approximation errors, a more curved surface may cause false positive, and also false negative results. Further enhancements were made to decrease the number of Type I errors occurring. The validation process with drop tests shed light on the fact that, further analyzing the models from a pose statistical perspective would prove highly beneficial. The probability of resting on a certain face is correlated to its area and the potential energy of the model in such pose. In our case we only used the potential energy data in our equilibrium calculating solution. The program worked with all the three poses, which had the least potential energy since, according to our experience, their resting facets have the highest probability of being on the ground surface after a drop.

The solution was also tested in a real life environment, the MTA SZTAKI smart factory introduced in [23]. The experiments conducted there also produced positive results, which supports the statement that our solution is a viable approach for the problems described in Section 2.

7. Conclusion and future work

Our aim was to offer a viable alternative to part feeding systems in robotic pick and place scenarios. We deduced that the most fitting solution would be to use image processing technologies powered by an enhanced dataset. We strived to produce such dataset, by calculating the equilibrium poses and faces of the workpiece. With the knowledge of the workpiece equilibria, the system could render top view images—in the equilibrium poses—of the object, paired with transformation data. After preprocessing these images, the object position could be calculated by comparing the image of a simple top view camera with this dataset [24].

One of the greatest benefits of our method is being an offline preprocessing step, creating a database of possible poses for every object. Although this calculation may take extra time the actual usage of the data will cut down on the number of possible sample artifacts for workpiece or object position determination, which greatly reduces the calculation time when applied in real time.

We also had the opportunity to assess the limitations derived from the finite approximations used in computer graphics. To further analyze these limitations and their effects, an objective evaluation tool was developed.

In the future we would like to decrease the rate of Type I errors even further and make it easier to handle them. This could be achieved by an adaptive algorithm, which sets the facet normal acceptance range based on the surface properties, and ties to identify unstable equilibria marked as stable.

Another solution could be implementing a pose statistics model directly to the program to help reducing Type I errors, as well as improving the speed of processing.

The solution was tested in our smart factory containing few work cells, but a larger scale comprehensive case study would also prove beneficial.

8. Acknowledgements

This work has been supported by the OTKA K-113038 and the GINOP 2.3.2-15-2016-00002 grants.

9. References

[1] Kinnell P, Rymer T, Hodgson J, Justham L, Jackson M. Autonomous metrology for robot mounted 3D vision systems. CIRP Annals. 2017. 66(1), 483-486.

[2] Juntao X, Zhiliang H, Rui L, Zhen L, Rongbin B, Zhengang Y, Hongxing P, Xiangjun Z. Visual positioning technology of picking robots for dynamic litchi clusters with disturbance. Computers and Electronics in Agriculture. 2018. 151, 226-237.

[3] Jiménez P. Visual grasp point localization, classification and state recognition in robotic manipulation of cloth: An overview. Robotics and Autonomous Systems. 2017. 92, 107-125.

[4] Weiwei W, Feng L, Zepei W, Kensuke H. Teaching robots to do object assembly using multi-modal 3D vision. Neurocomputing. 2017. 259, 85-93.

[5] Martínez D, Alenyà G, Torras C. Relational reinforcement learning with guided demonstrations. Artificial Intelligence. 2017. 247, 295-312.

[6] Duque D A, Prieto F A, Hoyos J G. Trajectory generation for robotic assembly operations using learning by demonstration. Robotics and Computer-Integrated Manufacturing. 2019. 57, 292-302.

[7] Pasquale G, Ciliberto C, Odone F, Rosasco L, Natale L. Are we done with object recognition? The iCub robot's perspective. Robotics and Autonomous Systems. 2019. 112, 260-281.

[8] Chena X, Guhl J. Industrial robot control with object recognition based on deep learning. Procedia CIRP. 2018. 76, 149-154.

[9] Kriegman D J. Computing stable poses of piecewise smooth objects. CVGIP: Image Understanding. 1992. 55(2), 109-118.

[10] Domokos G, Várkonyi P L. Geometry and self-righting of turtles. Proceedings of the Royal Society of London B: Biological Sciences. 2008. 275(1630), 11-17.

[11] Várkonyi P L. Estimating part pose statistics with application to industrial parts feeding and shape design: new metrics, algorithms, simulation experiments and datasets. IEEE Transactions on Automation Science and Engineering. 2013. 11(3), 658-667.

[12] Várkonyi P L. The secret of gambling with irregular dice: estimating the face statistics of polyhedra. Periodica Polytechnica Architecture. 2014. 46(1), 1-5.

[13] Kriegman D J. Let them fall where they may: Capture regions of curved objects and polyhedral. International Journal of Robotics Research. 1997. 16(4), 448-472.

[14] Zamfirescu T. How do convex bodies sit?. Mathematika. 1995. 42(1), 178-181.

[15] Domokos G, Sipos A Á, Szabó T. The mechanics of rocking stones: equilibria on separated scales. Mathematical Geosciences. 2012. 44(1), 71-89.

[16] Rodriguez A, Bowling A. Solution to indeterminate multipoint impact with frictional contact using constraints. Multibody Sytem Dynamics. 2012. 28(4), 313-330.

[17] Ngoi B K A, Lim L E N, Ee J T. Analysis of natural resting aspects of parts in a vibratory bowl feeder - validation of "Drop Test". The International Journal of Advanced Manufacturing Technology. 1997. 13(4), 300-310.

[18] Suresh M, Nisaanthakumar N. Determining the most probable orientation of a part using centroid solid angle method. Journal of Advances in Mechanical Engineering and Science. 2015. 1, 21-27.

[19] Wiegleyy J, Raoz A, Goldberg K. Computing a statistical distribution of stable poses for a polyhedron. Proc. 30th Annu. Allerton Conf. Commun., Contr., Comput. Urbana-Champaign: Univ. Illinois; 1992.

[20] ASSIMP, Open Source Asset Importer, http://www.assimp.org/

[21] CGAL, Computational Geometry Algorithms Library, http://www.cgal.org/

[22] Mirtich B. Fast and accurate computation of polyhedral mass properties. Journal of Graphic Tools. 1996. 1(2), 31-50.

[23] Kemény Zs, Beregi R, Erdős G, Nacsa J. The MTA SZTAKI smart sactory: Platform for research and project-oriented skill development in higher education. Procedia CIRP. 2016. 54, 53-58.

[24] Tripary B, Erdős G. Planning and optimization of 2D vision-based robotic pick-and-place operations. MTA SZTAKI Technical Report 2019