Stability Assessment Through Mechanical Analysis and Machine Learning

Nie Yiding 12111826, Fan Yuheng 12111034, Wang Junling 12112821,
Huang Shuo 12111608, Liu Guanhua 12010711, Kong Zhiwei 12111205

***Abstract*—Grasp points optimization enables significant feedforward control in robot manipulation, providing optimized decisions for robot grasping scenarios. Our work combines both mathematical analysis and neural network learning to obtain simultaneous solutions for grasp points of any given object. We generate training data using a mathematical approach and apply the acquired model to real-life applications. The proposed algorithm demonstrates significant improvements in grasp stability and reduced solution times for both simple and complex objects in a simulated environment. Our focus is on 3D-2Point grasp of rigid bodies, with a notable increase in overall stability. The developed stability evaluation method could also optimize multi-finger grasp points in 3-D space in future work.**

I. Introduction

The focus of this research is on how to grasp objects most stably. Compared with the traditional search for the easiest grasping point based on computer vision, this study focuses more on mathematical modeling through physical methods, to find the most resistant to interference and the most stable grasping point and reduce the computational cost through neural network. For most objects, using vision-based grasp points is a more affordable option; The main purpose of this research is to provide solutions for the grasping of some special objects, such as fragile, deformable, explosive and other high-value objects. In this paper, we will first present our mathematical model, discuss its working principle, and explain why it can evaluate the stability of grasping well. We will then introduce the neural network we use in the simplified computation process, and through its training and use, the implementation cost of the method proposed in this study becomes acceptable from unacceptable. Later, we will introduce our method of obtaining data. To train the neural network, we generate point clouds of 6 kinds of objects with certain differences, and in order to adapt to the high time complexity of the mathematical model, only the grasping features are retained for each object, thus greatly reducing the amount of computation. Finally, in the experimental part, we design an experimental method. By comparing the grasping points corresponding to our trained neural network with the grasping points chosen intuitively, we finally prove the reliability of this method. At the same time, we also mention the shortcomings of this network when working in special cases.

II. Related work

A. Grasp Stability Prediction with Time Series Data Based on STFT and LSTM

This study by Tao Wang and Frank Kirchner explores the use of Short-Time Fourier Transform (STFT) and Long Short-Term Memory (LSTM) networks for predicting grasp stability using time series data from force and pressure sensors. The paper highlights the effectiveness of combining these techniques to predict unstable grasps, potentially enhancing the application of AI in traditional industries. Their model was tested across different grippers, showing promising results in generalizing across various sensor types, thus providing a robust method for grasp stability analysis [2].

B. Grasp Stability Prediction for a Dexterous Robotic Hand Combining Depth Vision and Haptic Bayesian Exploration

This research delves into grasp stability prediction by integrating depth vision and haptic feedback through Bayesian methods. The study employs a multimodal approach where both visual and tactile data are used to predict the stability of a grasp on unknown or un-modeled objects. By using depth sensors and Bayesian exploration techniques, the researchers could make real-time adjustments to the grasping action, enhancing the reliability and safety of the robotic hand during complex manipulations [3].

C. Task-Oriented Grasp Planning Based on Disturbance Distribution

Focusing on optimizing grasp configurations based on anticipated disturbances, this paper presents a novel approach to grasp planning. The methodology involves analyzing potential disturbances that might affect the grasp and planning the gripper's actions accordingly to mitigate these effects. This strategic planning is crucial for applications requiring high precision and reliability in dynamic environments, as it enables the robotic system to maintain stability and control even under unpredictable conditions [1].

Each of these papers contributes uniquely to the field of robotic grasping by addressing the critical aspects of grasp stability from different technological and methodological perspectives.

METHODS

A. Method of Grasp Points Optimization

Our approach towards grasp points optimization consists of training data generation and neural network learning. We first generate optimized grasp points of given point clouds using a mathematical model and then feed the obtained training data to the neural network.

The mathematical model determines the stability of each provided grasp points set (pair of grasp points in our case) through the stability evaluation algorithm we developed, and finds the largest combination of grasp points in the provided point cloud (input object).

2-point grasping of an object is considered statically undetermined, which means the applied force on each side of the gripper has infinite amount of possibilities, while only the critical scenario where the gripper is about to drop the object can be determined with given physical parameters.

During stability evaluation, our mathematical model tests every possible gripper force that fulfills both friction restrictions and normal force restrictions. A point in the resultant external action domain is then calculated and recorded accordingly [4]. The weighted volume of space that points of plausible resultant external action occupied, which represents the stability of the grasp points, is calculated through Eq.1:

$$S = \oiiint (F + M/l) \cdot e^{-(F+M/l)} \cdot dV \cdots\cdots Eq.1$$

where S is the stability, $\vec{F}_{\text{res}}$ is the resultant external force, $\vec{M}_{\text{res}}$ is the resultant external moment of force, and d is the distance between two grasp points.

The resultant external action domain is a 6D flat space, in which a point is determined by 3 components of external force and 3 components of external moment of force.

In our application, the number of points in the external action domain is designed to be around 2000, which presents balanced performance between solution time and accuracy.

The physical theorems we choose to describe grasping include friction law in both force and moment of force. The action force in a contact point is modeled by Eq.2:

$${f_x}^2 + {f_y}^2 + k \cdot M^{\alpha} = \mu^2 N^2 \cdots\cdots Eq.2$$

where f_x, f_y are two components of in-plane friction, k is a coefficient of moment related to material properties, M_z is the normal moment of force caused by the uneven distribution of in-plane shear stress (friction), μ is the coefficient of friction, and N is the normal force applied by the gripper.

The resultant external action, in regard to every component of action from both gripping points, is described by Eq.3:

$$\begin{bmatrix} \frac{\hat{n}_1 \times \hat{z}}{|\hat{n}_1 \times \hat{z}|}\cdot\hat{x} & \frac{\hat{n}_1 \times (\hat{n}_1 \times \hat{z})}{|\hat{n}_1 \times \hat{z}|}\cdot\hat{x} & \hat{n}_1\cdot\hat{x} & 0 & \frac{\hat{n}_2 \times \hat{z}}{|\hat{n}_2 \times \hat{z}|}\cdot\hat{x} & \frac{\hat{n}_1 \times (\hat{n}_2 \times \hat{z})}{|\hat{n}_2 \times \hat{z}|}\cdot\hat{x} & \hat{n}_2\cdot\hat{x} & 0 \\ \frac{\hat{n}_1 \times \hat{z}}{|\hat{n}_1 \times \hat{z}|}\cdot\hat{y} & \frac{\hat{n}_1 \times (\hat{n}_1 \times \hat{z})}{|\hat{n}_1 \times \hat{z}|}\cdot\hat{y} & \hat{n}_1\cdot\hat{y} & 0 & \frac{\hat{n}_2 \times \hat{z}}{|\hat{n}_2 \times \hat{z}|}\cdot\hat{y} & \frac{\hat{n}_1 \times (\hat{n}_2 \times \hat{z})}{|\hat{n}_2 \times \hat{z}|}\cdot\hat{y} & \hat{n}_2\cdot\hat{y} & 0 \\ \frac{\hat{n}_1 \times \hat{z}}{|\hat{n}_1 \times \hat{z}|}\cdot\hat{z} & \frac{\hat{n}_1 \times (\hat{n}_1 \times \hat{z})}{|\hat{n}_1 \times \hat{z}|}\cdot\hat{z} & \hat{n}_1\cdot\hat{z} & 0 & \frac{\hat{n}_2 \times \hat{z}}{|\hat{n}_2 \times \hat{z}|}\cdot\hat{z} & \frac{\hat{n}_1 \times (\hat{n}_2 \times \hat{z})}{|\hat{n}_2 \times \hat{z}|}\cdot\hat{z} & \hat{n}_2\cdot\hat{z} & 0 \\ \multicolumn{3}{c}{\vec{r_1} \times \begin{bmatrix} \frac{\hat{n}_1 \times \hat{z}}{|\hat{n}_1 \times \hat{z}|}\cdot\hat{x} & \frac{\hat{n}_1 \times (\hat{n}_1 \times \hat{z})}{|\hat{n}_1 \times \hat{z}|}\cdot\hat{x} & \hat{n}_1\cdot\hat{x} \\ \frac{\hat{n}_1 \times \hat{z}}{|\hat{n}_1 \times \hat{z}|}\cdot\hat{y} & \frac{\hat{n}_1 \times (\hat{n}_1 \times \hat{z})}{|\hat{n}_1 \times \hat{z}|}\cdot\hat{y} & \hat{n}_1\cdot\hat{y} \\ \frac{\hat{n}_1 \times \hat{z}}{|\hat{n}_1 \times \hat{z}|}\cdot\hat{z} & \frac{\hat{n}_1 \times (\hat{n}_1 \times \hat{z})}{|\hat{n}_1 \times \hat{z}|}\cdot\hat{z} & \hat{n}_1\cdot\hat{z} \end{bmatrix}} & \hat{n}_1 & \multicolumn{3}{c}{\vec{r_2} \times \begin{bmatrix} \frac{\hat{n}_2 \times \hat{z}}{|\hat{n}_2 \times \hat{z}|}\cdot\hat{x} & \frac{\hat{n}_1 \times (\hat{n}_2 \times \hat{z})}{|\hat{n}_2 \times \hat{z}|}\cdot\hat{x} & \hat{n}_2\cdot\hat{x} \\ \frac{\hat{n}_2 \times \hat{z}}{|\hat{n}_2 \times \hat{z}|}\cdot\hat{y} & \frac{\hat{n}_1 \times (\hat{n}_2 \times \hat{z})}{|\hat{n}_2 \times \hat{z}|}\cdot\hat{y} & \hat{n}_2\cdot\hat{y} \\ \frac{\hat{n}_2 \times \hat{z}}{|\hat{n}_2 \times \hat{z}|}\cdot\hat{z} & \frac{\hat{n}_1 \times (\hat{n}_2 \times \hat{z})}{|\hat{n}_2 \times \hat{z}|}\cdot\hat{z} & \hat{n}_2\cdot\hat{z} \end{bmatrix}} & \hat{n}_1 \end{bmatrix} \begin{bmatrix} f_{1x} \\ f_{1y} \\ N_1 \\ M_1 \\ f_{2x} \\ f_{2y} \\ N_2 \\ M_2 \end{bmatrix} = \begin{bmatrix} \vec{F} \\ \vec{M} \end{bmatrix}$$

With the stability evaluation algorithm we proposed, the model can search for the most stable grasp points through various methods (traversal in our case).

However, the mathematical model requires a large amount of time to process an unknown point cloud, which makes it impractical to implement this model in real-life applications where instant solutions of grasp points are always desired. Thus, we use the mathematical model to train a neural network model, which promises a rapid response.

B. Grasp Representation

We parameterize a grasp as a combination of two contact points and a rotation angle. The contact points are two points touched by the robot gripper on the object surface. In general, one is visible while the other is invisible if the object is partially observed. The rotation angle is formed by the plane of the gripper (dashed lines) with the horizontal plane. Its range is limited within $[0, \pi]$ in order to avoid collision with the ground surface.

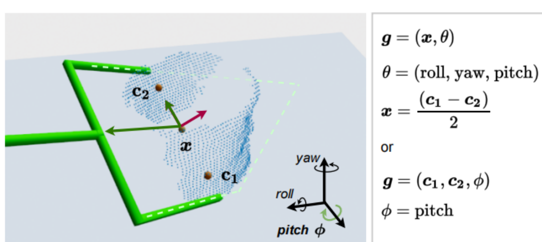

Fig. 1. Grasp representation. The 7-DoF grasp is made up of two contact points and a plane-to-plane rotation angle.

C. Learning to Grasp (L2G)

We adopt the network in [1] as our baseline. As is displayed in Fig. 1, a feature extractor is first applied to extract point-wise features from the input point cloud. Subsequently, the per-point features are utilized by a point sampler to sample a handful of points that constitute the visible contact points of the grasp. Thereafter, point features around the sampled points are aggregated and fed into a grasp regressor, which directly estimates the rest of the grasp components, i.e., the invisible contact points and the plane-to-plane rotation angles. Lastly, the aggregated point features and the raw grasp candidates are passed into a grasp classifier that assigns a grasp quality score in [0, 1] to each grasp candidate, denoting whether it is graspable (1) or not (0).

D. Implementation Details

In our work, we employ DeCo [2] as the feature extractor backbone. We sample N=2000 points from the object's point cloud and use the default configurations of DeCo in the original paper. We set the point feature dimension F=128, and the feature aggregation neighbor size to 20. The model is queried to predict M=50 grasp candidates and the corresponding grasp quality scores. At training time, we use a batch size of 4, a learning rate of 0.0001, a weight decay of 0.0001, and the Adam optimizer to train the model for 500 epochs. At inference time, we extract the grasp with the highest grasp quality score and then execute it on the robot in simulation.

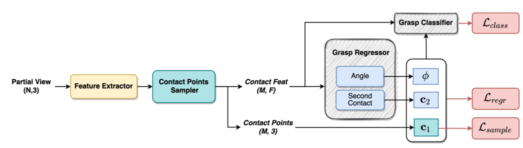

Fig. 2. Model overview. The model takes as input a point cloud and outputs a 7-DoF grasp together with a 0-1 grasp quality score.

DATA

In this project, we utilized point cloud data stored in `.npy` files. The data consists of matrices representing geometric shapes. Specifically, we worked with two types of matrices:

1. **6×n Matrix**: Each row in this matrix represents a point in 3D space. The columns include

the xyz coordinates of the point and its corresponding normal vector, providing comprehensive geometric information about each point. 2. **3×n Matrix**: This matrix includes only the xyz coordinates of the points, representing the point cloud data of a geometric shape without the normal vectors.

The variable n represents the number of points that make up the point cloud for each geometric shape. These matrices allow for detailed representation and analysis of the geometric properties of the shapes in our study.

Sources of Point Cloud Data

The point cloud data has two primary sources:

1. **Depth Camera Data**: Point cloud data was captured using a depth camera from four different angles. By utilizing the camera parameters and information, depth images from these four angles were stitched together to form a comprehensive point cloud representation of the actual objects. 2. **Mathematical Modeling**: For basic geometric shapes, point cloud data was generated artificially using mathematical modeling and methods. This approach enabled us to create accurate point clouds for simple geometries.

Dataset Division

The dataset is divided into two parts:

1. **6×n Matrices**: Each matrix includes the xyz coordinates and corresponding normal vectors of the points. Here, n is approximately 50, indicating that each matrix consists of 50 points and their normal vectors. 2. **3×n Matrices**: Each matrix includes only the xyz coordinates of the points. In this case, n is 2000, meaning that each matrix consists of 2000 points.

Each part includes the following geometric shapes:

- Sphere

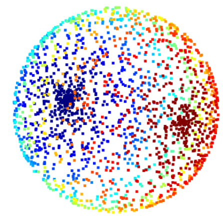

Fig. 3. Sphere

- Cylinder

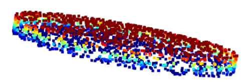

Fig. 4. Cylinder

- Triangular prism

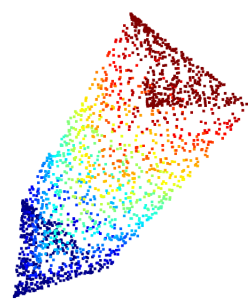

Fig. 5. Triangular prism

- Rectangular prism

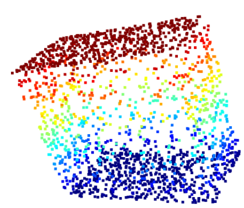

Fig. 6. Rectangular prism

- Octagonal prism

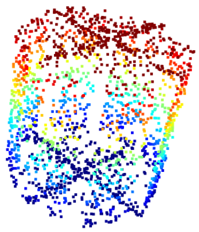

Fig. 7. Octagonal prism

- A complex object formed by two hemispheres

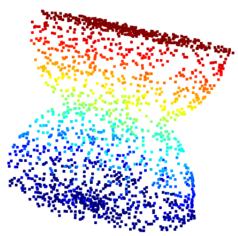

Fig. 8. A complex object formed by two hemispheres

For each geometric shape, we have 100 sets of data, providing a robust dataset for our analysis and experiments.

Data Preprocessing

For the 6×n matrices, as these are used for model training, we performed preprocessing to optimize for computational efficiency. Given the hardware constraints, we retained only the key grasping feature points of the geometric objects and removed redundant points that did not contribute additional information for grasping.

Experiments

To verify and evaluate the effectiveness of the prediction model obtained through machine learning, we utilized CoppeliaSim to construct a series of simulated environments as depicted in the images below. Our basic validation approach involves placing the objects in their designated positions, using all point cloud information of the target objects in the environment as the input for our prediction model. After the model has completed its calculations, we send the corresponding poses to the robotic arm, ultimately enabling it to successfully grasp the target object and return to a specific position, P.

Here, to verify the stability of our model, we have constructed a track in the simulated environment. We wait until the robotic arm holding the target object returns to point P, then place a stationary ball at the same position on the track. Under the influence of its own gravity, the ball will also move to point P. We simulate external environmental disturbances to the grasped object by having the ball collide with the target object. Since both the release position and initial velocity of the ball are zero, we will continuously change the mass of the ball until we find a critical mass that allows the robotic arm to just stabilize the target object. Once this critical mass is exceeded, the robotic arm will lose control of the grasped object. We will use the critical mass of this ball to measure the stability of our model in grasping objects.

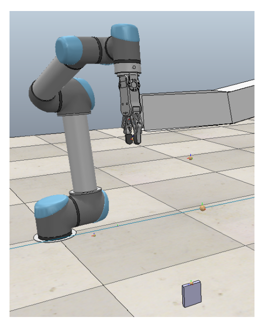

Fig. 9. Simulation scenario

After grasping the target object at different points and returning to point P, the stability varies. As shown in our previous stability formula, for a regular quadrilateral pyramid (rectangular prism), under two-point grasping, the longer the edge length between the two grasping points, the higher the grasping stability, and vice versa, the lower the stability. The table below records the stability of grasping target objects with different centroid positions and shapes during the testing phase.

Index	Centroid [x,y,z]	Length [lx,ly,lz]	Grasping-point	Critical mass (kg)
1	[-0.15,0.75,0.03]	[0.03,0.02,0.06]	[-0.15,0.75,0.62]	2.31
			[-0.15,0.75,0.02]	2.61
2	[-0.15,0.75,0.03]	[0.06,0.04,0.06]	[-0.15,0.75,0.55]	2.62
			[-0.15,0.75,0.02]	5.28
3	[-0.15,0.75,0.03]	[0.05,0.01,0.06]	[-0.15,0.75,0.60]	2.72
			[-0.15,0.75,0.02]	2.97
4	[-0.15,0.75,0.03]	[0.05,0.07,0.06]	[-0.15,0.75,0.55]	1.57
			[-0.15,0.75,0.02]	3.55
5	[-0.15,0.75,0.05]	[0.03,0.02,0.10]	[-0.15,0.75,0.90]	2.43
			[-0.15,0.75,0.50]	2.60

Fig. 10. Test Data Performance Table

In the table above, ”Index” corresponds to the test number, ”centroid [x,y,z]” represents the current centroid position of the object, and ”length [lx,ly,lz]” indicates the current length, width, and height of the object. ”Grasping-point” refers to the central position of the robotic gripper during grasping (all units are in meters). ”Critical mass” is the maximum mass of the ball that the robotic arm can balance and stabilize at point P under environmental disturbances.

In each major row, the lower sub-row represents the robotic gripper grasping the target object with a longer length, while the upper sub-row represents the robotic gripper grasping the target object with a shorter length. According to our previous analysis, when the robotic gripper grasps the target object with a longer length, the grasp stability should be higher. Here, the stability corresponding to the lower sub-rows in each major row is relatively greater, and these sub-rows correspond to the robotic arm grasping the target object with a longer length, thus verifying our previous stability hypothesis. This further confirms the correctness of our proposed formula for measuring grasp stability and the accuracy of the object coordinates obtained through machine learning.

Moreover, for obtaining grasp points in these scenarios, if the optimal grasp point positions were calculated entirely using classical mechanics and other physical methods, the computation time for a single scenario would be approximately 10 minutes. However, with our model, the computation time for a single scenario is only 0.5 seconds. The time difference between achieving the same highly stable grasp points is nearly 1000 times! Our machine learning model for obtaining the target object’s point cloud and its corresponding grasp points maintains high accuracy and object stability while significantly reducing computational costs and improving efficiency.

Conclusion

In this research, we modeled the grasping task mathematically through physical methods, so as to propose a new method to examine the grasping stability and carried out code implementation. This method has some shortcomings in terms of time complexity, but it can effectively evaluate the stability of the grasping point. By training the neural network, the efficiency of the method proposed in this paper is greatly improved, so that it becomes acceptable in the actual work requirements. However, due to the amount of training data or the degree of difference and other factors, our trained network works well in convex polyhedral with grabbable symmetric plane features such as quadrilateral prism, but there are shortcomings when working in objects with only circular or spherical grabbable surfaces.

Future research can focus on the following points: First, to further improve the authenticity of this mathematical model, such as more accurate modeling of elastic deformation of objects and mechanics of contact points; The second is to further improve the number and difference of neural network training data in order to achieve better results. The third is to test in a real environment to further explore the reliability of the method proposed in this study.

Acknowledgments

The authors would like to thank their peers and professors at the Southern University of Science and Technology for their support and guidance.

References

[1] Computation of multi-fingered grasping force with linear combination. https://max.book118.

com/html/2019/0115/6101222011002002.shtm, 2019. Accessed: 2024-04-14.

[2] S. Kumra and C. Kanan. Robotic grasp detection using deep convolutional neural networks. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 769–776, Vancouver, BC, Canada, 2017. doi: 10.1109/IROS.2017.8202237.

[3] A. Zeng, K.T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, and J. Xiao. Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1386–1383. IEEE, May 2017.

[4] Tony Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *ArXiv*, abs/2304.13705, 2023. URL https://api.semanticscholar.org/CorpusID:258331658.