Proceedings of the ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference
IDETC/CIE 2016
August 21-August 24, 2016, Charlotte, North Carolina, USA

IDETC2016-60095

A DATA-DRIVEN APPROACH TO PREDICT HAND POSITIONS FOR TWO-HAND GRASPS OF INDUSTRIAL OBJECTS

Erhan Batuhan Arisoy
Siemens Corporate Technology
Corporate Research
755 College Road East
Princeton, New Jersey 08540
erhan.arisoy@siemens.com

Guannan Ren
Siemens Corporate Technology
Corporate Research
755 College Road East
Princeton, New Jersey 08540
guannan.ren@siemens.com

Erva Ulu
Carnegie Mellon University
Mechanical Engineering
5000 Forbes Avenue
Pittsburgh, Pennsylvania 15213
eulu@cmu.edu

Nurcan Ulu
Carnegie Mellon University
Mechanical Engineering
5000 Forbes Avenue
Pittsburgh, Pennsylvania 15213
ngu@cmu.edu

Suraj Musuvathy
Siemens Corporate Technology
Corporate Research
755 College Road East
Princeton, New Jersey 08540
suraj.musuvathy@siemens.com

ABSTRACT

The wide spread use of 3D acquisition devices with high-performance processing tools has facilitated rapid generation of digital twin models for large production plants and factories for optimizing work cell layouts and improving human operator effectiveness, safety and ergonomics. Although recent advances in digital simulation tools have enabled users to analyze the workspace using virtual human and environment models, these tools are still highly dependent on user input to configure the simulation environment such as how humans are picking and moving different objects during manufacturing. As a step towards, alleviating user involvement in such analysis, we introduce a data-driven approach for estimating natural grasp point locations on objects that human interact with in industrial applications. Proposed system takes a CAD model as input and outputs a list of candidate natural grasping point locations. We start with generation of a crowdsourced grasping database that consists of CAD models and corresponding grasping point locations that are labeled as natural or not. Next, we employ a Bayesian network classifier to learn a mapping between object geometry and natural grasping locations using a set of geometrical features. Then, for a novel object, we create a list of candidate grasping positions and select a subset of these possible locations as natural grasping contacts using our machine learning model. We evaluate the advantages and limitations of our method by investigating the ergonomics of resulting grasp postures.

INTRODUCTION

The ever rising demand for innovative products, more sustainable production, and increasingly competitive global markets require constant adaptation and improvement of manufacturing strategies. Launching faster, obtaining higher return on investment, and delivering quality products, especially in demanding economic times and considering regulatory factors necessitates optimal planning and usage of manufacturing production capacity. Digital simulation of production plants and factories are invaluable tools for this purpose. Commercial software systems such as Siemens PLM Software Tecnomatix provide powerful

Copyright © 2016 by ASME

simulation functionality, and tools for visualizing and analyzing results of the simulations.

Key aspects of optimizing manufacturing facilities that involve human operators include optimizing work cell layouts and activities for improving human operator effectiveness, safety and ergonomics. Examples of operations that are typically configured and analyzed in a simulation include humans picking and moving objects from one place to another, assembling a product consisting of multiple components in a factory, and using hand tools to perform maintenance tasks. One of the challenges in configuring such a simulation is in specifying the locations of the grasp points on objects that human interact with. The current approach relies on a manual process through which a user must specify the places where the human model should grasp each object. This is a tedious and time consuming process, and therefore a bottleneck in configuring large scale simulations. Therefore automated techniques for estimating natural grasp points are desirable. This paper presents a data driven approach for estimating natural looking grasp point locations on objects that human operators typically interact with in production facilities. These objects include mechanical tools, parts and components specific to products being manufactured or maintained such as automotive parts, etc. The proposed system takes a CAD model of an object as input, and outputs a list of candidate natural grasping point pairs. Each point pair consists of one point to place the palm of the left hand, and the other to place the palm of the right hand. We start with a crowdsourced database that consists of CAD models and corresponding grasping point locations that are labeled as natural or not. A Bayesian network classifier is used to learn a mapping between object geometry and natural grasping locations using a set of geometrical features. Then, for a new object not present in the database, we create a list of candidate grasping positions and select a subset of these possible locations as natural grasping contacts using the machine learning model. We evaluate the advantages and limitations of our method by investigating the ergonomics of resulting grasp postures. Our main contributions are as follows:

1. A novel grasp point estimation algorithm for two-hand grasps of objects that is tailored towards the digital simulation for production planning.
2. A set of geometric features to capture the natural appearance of two-hand grasps.
3. Ergonomics based evaluation of the performance.

Section 2 presents related work. Section 3 presents an overview with technical details in Section 4. Results and discussion of the proposed approach are presented in Section 5, and conclusions are summarized in Section 6.

RELATED WORK

Analysis of holding objects has been explored in robotic grasping, computer graphics and fixture design. In robotics, researchers try to estimate contact points on an object based on sensor data that they have in the current setup. In general, problem is shrunk into hand grasping for certain types of robotic hands. Several examples of approaches adopted in the robotics fields can be found in [1–4]. In general, disadvantages of these methods are that they are all hardware dependent and main focus is daily used objects (such as bottles, cans and pans). On the other hand, researchers in computer graphics are interested in human grasping for simulation and animation purposes where natural looking grasping motions are desired. Some examples of grasping applications developed in computer graphics include [5, 6]. In fixture design, the grasping is studied to solve the problem of maintaining a specified position and orientation of an object in the presence of external disturbances (such as cutting forces in manufacturing). Since precision of the manufacturing process depends on workpiece stability, constraining the workpiece is critical [7]. Due to strict performance requirements, most approaches use physics based methods. Examples include [8] where Wang et al. uses force closure solutions for precision fixture design and a fixture layout design method based on largest simplex calculation [9].

While robotic grasping and computer graphics applications mainly focus on handling of everyday objects in our daily lives, fixture design examines holding mechanical objects firmly in place during manufacturing processes. In this paper, we focus on mechanical objects that will be handled in a factory environment as in fixture design. However, our main motivation is similar to computer graphics applications where achieving natural looking grasps is the main priority rather than firm and strict grasps.

In these three areas, many techniques have been developed for grasping. The most common approaches to grasping problem include physics based analytical methods and data-driven methods. In literature, data-driven methods are presented to capture human decision making in grasping process. The examples include primitive based approaches [10, 11] where objects in database are represented as combinations of simple primitive shapes (such as cylinder, elipsoid, and cuboid) and shape matching methods [12] where suitable grasping pose is matched with the object to be manipulated. Other data driven methods are based on collecting human grasping data through labeling [2, 10, 12–14] or motion capture [15]. In physics based methods, the main idea is to find a set of feasible contact points that is optimum in terms of a pre-defined quality measure. Examples include [16] where authors present a method to select grasp points that minimize the contact forces. Similarly, Chinellato et al. [17] use geometrical constraints based on grasp polygon to evaluate the feasible point sets. A review of grasp quality measures can be found in [18].

While having an optimum solution for a certain physical ob-

Copyright © 2016 by ASME

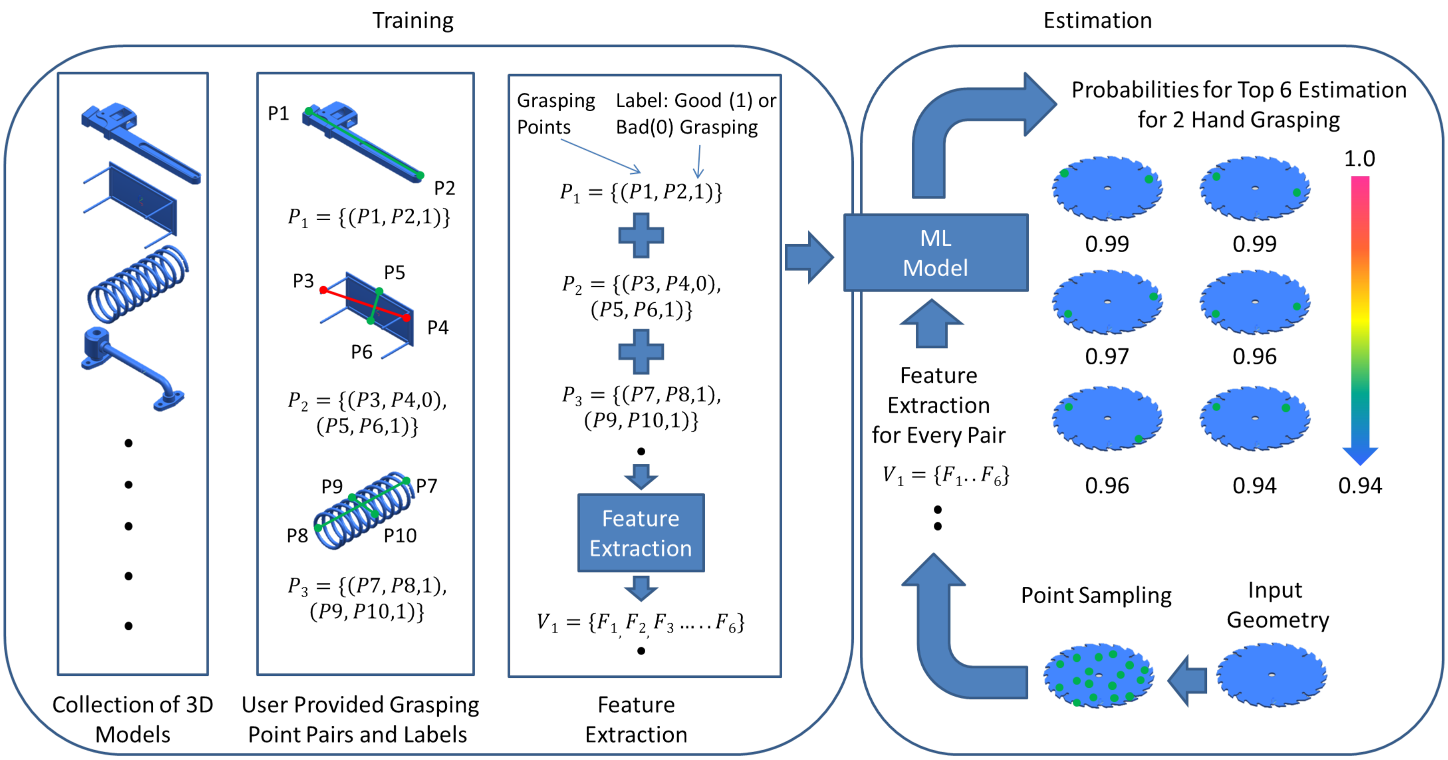


FIGURE 1. Overview of the proposed methodology for data-driven grasping point estimation.

jective could be suitable for robotic grasping applications and fixture design, we are, here, mainly interested in finding grasping configurations that are closest to natural human behavior. For this reason, instead of choosing a physical approach we use a data-driven method with crowdsourced labeling of human grasp.

OVERVIEW

The objective of the proposed methodology is to design and prototype a decision support framework for estimating natural grip positions for a new 3D object. To achieve this goal, we take inspirations from the fact [ref] that humans are able to identify good grasping locations for novel objects, in a fraction of a second, based on their previous experiences with grasping different objects. To mimic this extraordinary capability, we are developing a learning based algorithm that utilizes a database of 3D models with corresponding crowdsourced natural grasp locations and identifies a set of candidate hand positions for two hand natural grasps of new objects. Figure 1 illustrates overview of the proposed methodology. Our natural grasping point estimation algorithm consist of 4 main phases: 1) Grasping database generation using crowdsourcing, 2) Geometrical feature selection and extraction for learning the relationship between objects' geometry and natural grasping point locations, 3) Learning phase for understanding how people grasp objects in a high dimensional feature space and 4) Data-driven grasp point estimation. For an efficient algorithm, we apply the following simplifications:

1. Objects will be lifted with both hands.
2. Objects are assumed to be solid and have uniform material distribution, hence center of mass matches with the centroid of the input geometry.
3. Objects are light enough to be carried by human.
4. We assume that the object does not contain handles or thin edges where humans can grasp these objects using these handles.
5. Hand/finger joint positions/orientations are ignored. Only hand positions will be estimated. A great analogy for this assumption is modeling the human workers as if they are wearing boxing gloves while lifting target objects.

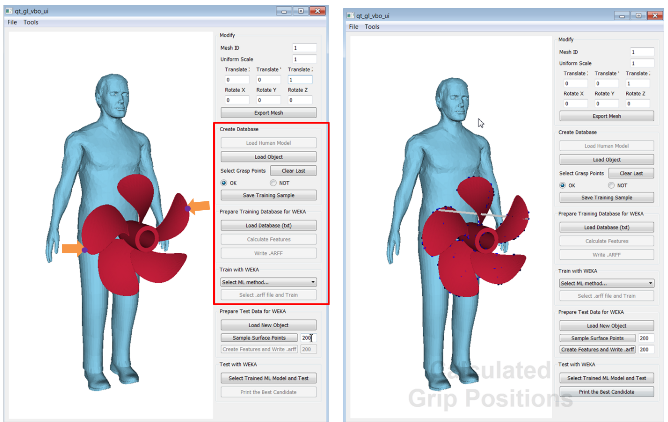
These assumptions will be detailed as we progress further.

TECHNICAL DETAILS

Grasping Database Generation

In order to estimate natural grasping positions given a new object, we took inspirations from the fact that human conceptual knowledge can identify grasping regions for a new target

Copyright © 2016 by ASME



a) Database generation and training phase

b) Estimated grasp locations connected by the gray line.

FIGURE 2. Developed user interface for data collection and grasping location estimation.

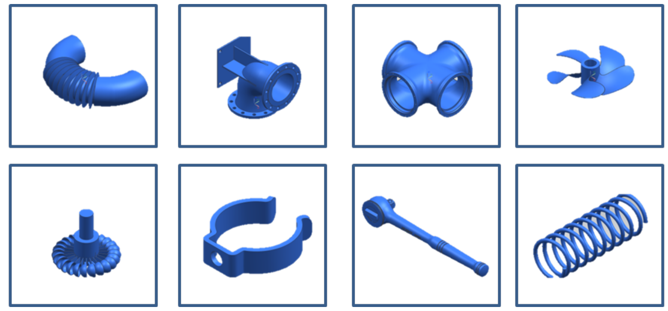


FIGURE 3. Example geometries in training and testing database.

object in a fraction of seconds based on his previous interactions with different objects. For instance, people may only need to see one example of a novel screw driver in order to estimate grasping boundaries of the new concept. Although recent studies for grasp location estimation focus on pure geometrical approaches, our goal is to mimic human conceptual knowledge to learn the way people create a rich and flexible representation for the grasping problem based on their past interactions with different objects and geometries. To achieve this goal, we have implemented a C++ based user interface where users can import 3D models and pick two candidate grasping locations on the imported 3D surface. After picking candidate grasping locations, the user has to label selected point pairs as good or bad grasping positions. We have utilized developed software interface in order to collect a database of 3D models and pair of grasping point pairs that are manually labeled as good or bad using crowdsourcing. In Figure 2, the graphical user interface for the software is shown. The object model is illustrated in red and dark blue points (pointed to by the orange arrows) correspond are manually selected to as the graspable contact points. The database generation menu is highlighted with the red boundaries. We have retrieved a set of 50 different CAD geometries from GrabCAD community including but not limited to screwdrivers, drills, razors, saws, etc. and preprocessed these geometries to scale to fit them in a 50 cm by 50 cm by 50 cm bounding box and adjusted their orientation such that the gravity is in the negative y direction and the front face of the object is aligned in the positive z direction. After preprocessing, we have applied different scaling transformations in order to populate our database with additional synthetic models. After the addition of the synthetic models, the total number of CAD models in our database is increased to 70. Figure 3 illustrates some of the CAD models in our database. Collecting and preprocessing a database of 3D CAD geometries is the first stage of our data generation phase. In the second phase, we asked 50 people to provide pairs of grasping point locations on the 3D geometry that is randomly selected among the models in our database and displayed to the user. The users are asked to provide examples of both good and bad grasping point locations and these point locations and corresponding geometries are recorded. The random draw from the database is determined by the current status of the distribution of the recorded both good and bad grasping point locations for every 3D model. For example, if the database has many both positive and negative grasping locations for a geometry A compared to geometry B, the random draw algorithm will lean toward selecting geometry B for grasp location data collection. We created our database with generalization and portability in mind. Current database includes over 1000 manually selected grasp positions. In order to save our grasping configurations, the following list shows the stored database information:

1. The name of the object file
2. The transformation matrix for the original object to its final location, orientation, and scale
3. Manually selected gripping locations (right hand, left hand)
4. Surface normal at gripping locations (right hand, left hand)
5. Classification of the instance (1 for graspable, 0 for not graspable)

Feature Selection

In this section, we describe the geometrical features that are used in our algorithm to capture the conceptual human knowledge that is encoded in the collected database of grasps. Our goal here is to find a mathematical representation that will allow us to determine whether a given grasp can be evaluated as viable or not. In particular, we would like our feature set to capture the natural way of grasping an object, therefore we base our formulations mainly on observations.

We want our feature set to contain the information about the stability and relative configurations of contact positions with re-

Copyright © 2016 by ASME

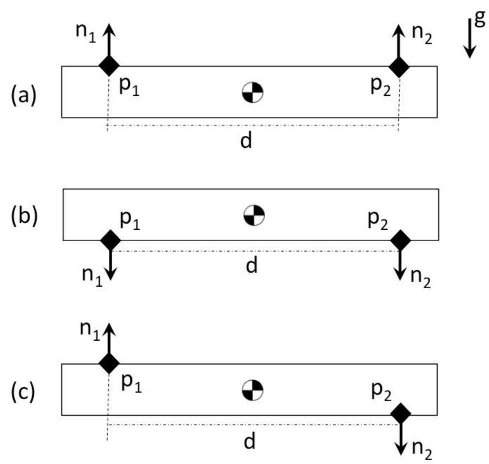


FIGURE 4. Although all three examples look the same in terms of distance based features (f^1 and f^2), only (b) is a stable grasp point configuration to carry the rectangular object. Features f^3 and f^4 allow us to distinguish between these three situations.

spect to each other and the center of the object's mass. To calculate the center of mass of an object in our database, we make the assumption that the center of mass can be approximated by the geometrical centroid of the object. The centroid is calculated by computing the surface integral over a closed mesh surface. For each grasping configuration, the contact positions are denoted as p_1 and p_2. The surface normals at p_1 and p_2 are marked as n_1 and n_2 and the location of the center of mass is denoted as p_{CoM}. The vector connecting p_1 to p_2 is labeled as n_c. Additionally, the signed distance between every grasping point and the vertical plane passing through the center of the mass of the input geometry is labeled as d_1 and d_2. Following equations present the calculation of n_c, d_1 and d_2 values.

$$\begin{aligned} n_c &= (p_1 - p_2)/\|p_1 - p_2\| \\ d_1 &= n_c \cdot (p_1 - p_{CoM}) \\ d_2 &= n_c \cdot (p_2 - p_{CoM}) \end{aligned} \tag{1}$$

Although we studied a wide variety of features to represent the solution space for the two-hand grasping problem, we identified following subset of geometrical features as the most relevant ones for our problem.

Feature 1: Humans tend to lift objects using symmetrical grasping locations with respect to the vertical plane passing through the center of mass in order to minimize the difference between lifting forces applied by both hands. In an effort to measure humans' tolerance to mismatch in this, we formulated our first feature as in Eq. 2.

$$f^1 = d_1 + d_2 \tag{2}$$

This feature also allows our algorithm learn and avoid generating unstable cases such as grasping an object from two points at one side of the center of mass.

Feature 2: Human anatomy allow them to extend their arms only to some extend while carrying an object comfortably. Similarly, keeping two hands very close while lifting a large object is uncomfortable for humans. In order to capture the comfortable range of distance between two grasp locations, we formulated the second feature as in Eq. 3.

$$f^2 = |d_1| + |d_2| \tag{3}$$

Feature 3 and 4: In addition to the distance based features, f^1 and f^2, inspired by [12] and [19], we store the angles formed between the surface normals and the line passing through the contact points as our third and fourth features, f^3 and f^4 (Eq. 4). Note that this formulation is based on the assumption that p_1 and p_2 correspond to contact points for certain sides hands (e.g. p_1 is right and p_2 is left hand) and this should be consistent throughout the entire database.

$$\begin{aligned} f^3 &= atan2(\|n_c \times n_1\|, n_c \cdot n_1) \\ f^4 &= atan2(\|n_c \times n_2\|, n_c \cdot n_2) \end{aligned} \tag{4}$$

These features allow us to distinguish contact pairs that look the same in terms of distance based features, f^1 and f^2, but very different in the sense of grasp stability. This discriminatory power is important to capture human tendency to balance forces and torques applied on the object while manipulating it. An example case is illustrated in Figure . Although all three cases would be evaluated the same in terms of f^1 and f^2, features f^3 and f^4 let us distinguish these cases from each other.

Feature 5: As our fifth feature, f^5, we use the angle formed between the gravitational field vector and the line passing through the contact points (Eq. 5). This feature captures the orientation of the grasping pairs mutually with respect to a global static reference.

$$f^5 = g \cdot n_c \tag{5}$$

Copyright © 2016 by ASME

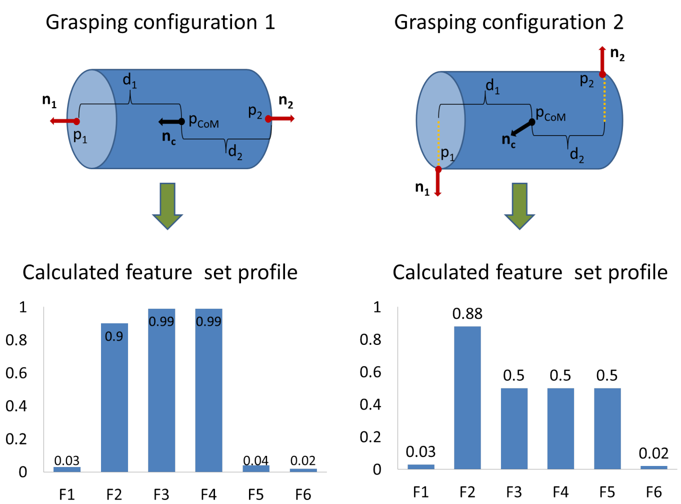


FIGURE 5. Feature set profiles calculated for 2 different configurations.

where g represents the gravitational field vector. In our applications, we take $g = [0, -1, 0]^T$.

Feature 6: The last geometrical feature that needs to be extracted for our learning problem is given in Eq. 6.

$$f^6 = z \cdot n_c \tag{6}$$

where z represents frontal direction at which human is facing. In our applications, we use $z = [0, 0, 1]^T$ by fixing the global coordinate frame on human body. Together with f^5, this feature allows our algorithm learn allowable orientation of human grasps with respect to a global static reference frame.

For every grasping point pairs i and j, a six dimensional feature vector (Eq. 7) is generated where every component corresponds to one of the calculated features.

$$F_{ij} = [\ f_{ij}^1,\ f_{ij}^2,\ f_{ij}^3,\ f_{ij}^4,\ f_{ij}^5,\ f_{ij}^6\ ]^T \tag{7}$$

Detailed illustration of features for different grasping conditions on a basic cylindrical geometry is shown in Figure 5. According to this figure, even if the target geometry to be lifted is the same for all four grasping cases, corresponding feature sets are unique for every case. The feature set profile demonstrates the capability of differentiating varying p_1 and p_2 configurations in the six dimensional feature space.

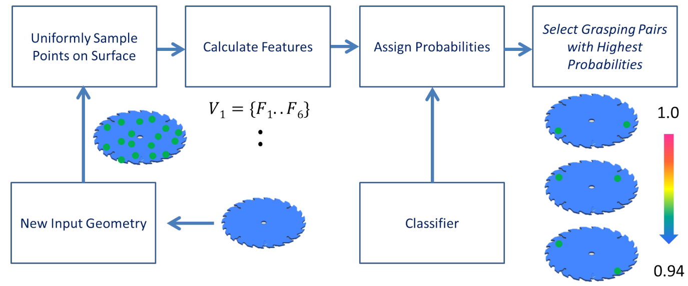


FIGURE 6. Pipeline for grasping point estimation.

Learning Phase

After the identification of the geometrical features to mathematically encode the configuration of different grasping locations on 3D geometries, our next task is selecting a machine learning model that can be trained on the collected grasping database using these features. The key learning problem that our approach focuses on is extracting a mapping between the geometry of 3D objects and the corresponding natural grasping locations for these 3D objects by mathematically encoding how people lift 3D objects in their daily lives using the database generated in section Feature Selection. To achieve this goal, we have utilized the WEKA machine learning toolkit [20] to experiment and study the performance of different machine learning models. First of all, we have partitioned our database into 2 sections: 1) training set (eighty percent of the entire database) and 2) testing set (twenty percent of the entire database). A collection of the geometries used for training and testing are displayed in Figure 3. After splitting the database into training and testing components, we performed multiple experiments with several types of different classifiers that are Naive Bayes Decision Trees, Random Forests and Multilayer Perceptron learning approaches.

To begin, we experimented with the simple decision tree classifier [21], namely J48, in the WEKA library. The key advantage of using decision tree classifier is the fact that it is fast to train and interpret the resulting tree. However, oftentimes, decision trees have a tendency to over-fit the training dataset, i.e. allowing high performance on the training dataset but poor performance on the testing data set. To alleviate overfitting issues, we apply tree pruning for the trained J48 classifier in WEKA. Another classifier we examined is WEKAs Random Forest classifier. The Random Forest classifier belongs to the ensemble-based learning methods. It can be viewed as an aggregation of decision trees and the majority vote among all of the trees determines the label for a classification problem. Each decision tree is constructed using a bootstrapped sample of the training dataset, and at each branch point, the best feature to split on is selected from a random subset of all features [22]. Furthermore, multilayer perceptron (MLP) is a feedforward artificial neural network algorithm that is trained using the backpropagation algorithm.

Copyright © 2016 by ASME

Classifier	Precision	Recall	F1
Random Forest	1.00	1.00	1.00
k-NN	1.00	1.00	1.00
J48	0.98	0.99	0.99
NBTree	0.97	0.98	0.98
Bayes Net	0.95	0.96	0.95
SMO	0.84	0.91	0.87
MLP	0.83	0.90	0.86

FIGURE 7. Training performance evaluation.

Classifier	Precision	Recall	F1
Bayes Net	0.98	0.60	0.74
Random Forest	0.96	0.73	0.83
J48	0.90	0.68	0.77
MLP	0.88	0.72	0.79
NBTree	0.87	0.69	0.77
k-NN	0.80	0.64	0.71
SMO	0.79	0.85	0.82

FIGURE 8. Testing performance evaluation.

Each hidden node in the network transforms the input data using a sigmoid function. This transformation allows the multilayer perceptron to model non-linear problems. Each layer within the MLP is fully connected to the next layer. We ran 500 epochs under a learning rate of 0.3 and a momentum rate of 0.2 when training the multi-layer perceptron classifier.

Additionally, we have experimented with various other machine learning classifiers. Similar to the J48 and the Random Forest algorithms, the Naive Bayes decision tree (NBTree) adds another tree-based classifier to the study. The NBTree is a combination of the Naive Bayes classifier and the decision tree classifier. It splits the dataset at each branch point and uses Naive Bayes classifier at the leaves [23]. We also experimented with the Bayesian network classifier. The default WEKA settings were used during the training process for the Bayesian Network.

Grasp Point Estimation

For grasping point estimation, we utilized the interface that we developed for database collection. The approach we adopted for the estimation process is shown in Figure 6. First of all, the user imports the 3D geometry of the target object as a triangular representation into our interface for grasping point estimation. Secondly, our estimation algorithm uniformly samples a fixed number of points on the 3D surface of the input geometry. The number of sampled points is controlled by a parameter adjusted by the user and these sample points serve as an initial candidate set for the estimation problem. Then, we randomly select pairs of points (since we focus on two-hand grasping, our pairs are groups of 2 points) among these uniformly sampled points and calculate feature vectors for every pair as described in previous section. Finally, we classify each candidate pair using their feature vector and assign probabilities them based on the classification results. The system automatically ranks candidate grasping pairs based on their probability values and displays 5 top grasping pairs. For visualization purposes, the system automatically creates gray lines that connect grasping points for every downselected pair.

RESULTS AND DISCUSSIONS

Our approach to learning a grasping location estimation function using a machine learning model that resembles human conceptual knowledge requires a database for both training and testing purposes. Based on crowdsourcing technique, we collected a database of 150 CAD geometries with over 1000 manually labeled good and bad grasping point configurations. However, crowdsourcing user studies require special care when user ratings are subjective [24]. We use a diligence based reliability check method to filter out unreliable responses. We expect that most of the respondents will complete labeling of a grasping instance within a similar period of time. We have discarded the grasping pairs that are labeled more than ten times faster than the mean labeling time by users. After the filtering operation, collected database is divided into training and testing datasets where 80 percent of the entire data is included for training. For the learning phase, we have employed several classification techniques and we have compared their individual performances on the testing dataset. We present the performance of these classifiers on the training set using precision, recall and F1 scores as shown in Figure 7. For training phase, we have used 10 fold cross validation technique and for the evaluation of different classifier techniques we use the precision as the performance metric. The highest training precision is observed for the Random Forest classification technique with a precision value of 0.96.

However, for the testing database, the highest precision value is observed with Bayesian Network Classification. Figure 8 illustrates the precision and recall values for the testing set. We discovered noticeable improvement on the precision when we used tree-based classifiers than function-based classifiers. We attribute this to the fact that there is noise in our training dataset, and the tree-based classifiers have high tendency to overfit the training dataset. Based on the precision values, we have selected Bayesian Network Classifiers for the grasping point estimation, although it has lowest recall value, which means that there is a high false negative rate. We believe that this behavior is because of the limited size of the collected database. Figure 9 illustrates some example results for the proposed data-driven grasping point

Copyright © 2016 by ASME

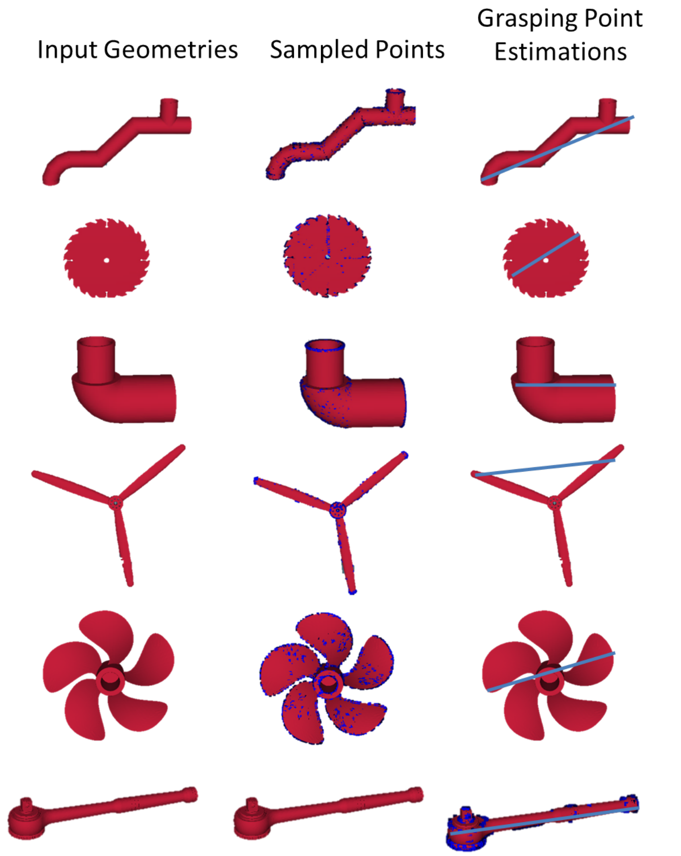


FIGURE 9. Example results for grasping point pair estimation (gray line illustrates identified best grasping point pairs with highest probability).

estimation algorithm using Bayesian Network Classifier. Additionally, we have performed an individual performance analysis for every feature in our proposed feature list in section Feature Selection using the testing database. According to this analysis, we conclude that feature 5 tends to be the most influential feature that separates the grasp-able and the non-grasp-able samples in the training dataset. 75 percent of the grasp-able data samples fall within 0 to 0.15 value for feature 5; while 75 percent of the non-grasp-able data samples fall between 0.4 to 1.0 value for feature 5. Figure 10 illustrates this relationship.

For training and testing purposes, we employed a desktop workstation with Intel Xeon E5620 2.40GHz processors and 16GB memory. Among the different classifiers we experimented with, we found that the training for the Multilayer Perceptron (MLP) classifier is the most complex and time-consuming learning technique. The MLP has many different training parameters, and we evaluated our classifiers based on the number of hidden

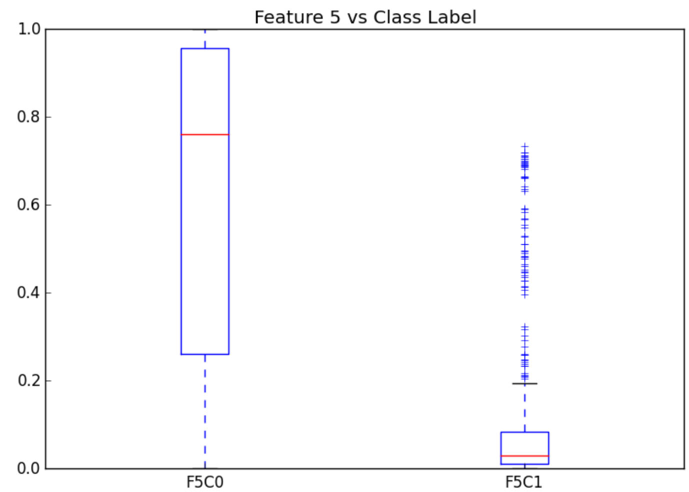


FIGURE 10. Effect of feature 5 on the classification result

layers, as well as the learning rate. We notice that the accuracy of the MLP classifier over the training set improves when the number of hidden layers are increased. However, the increase in accuracy levels off after adding 3 hidden layers as shown in Figure 11.

In addition to precision analysis on the testing database, we utilized Siemens Jack Software to evaluate if the grasping points identified by the data-driven algorithm result in comfortable grasping. To do that, we employed the comfort assessment tool under Packaging Toolkit. This tool enables comfort analysis of two different body postures with respect to each other for grasping or lifting tasks. The Jack comfort assessment tool is based on the comfort study [25] that targeted the ergonomics aspect of drivers. The comfort assessment interface shows the name of the joint on the left panel, a bargraph displaying the current postures deviation from the mode, the numeric comfort values at the current position, and the range and mode based on research data.The mode value of a given joint can be interpreted as the most adopted posture of the joint during driving. We expect that the grasping points with high classification probability will result in body postures with higher comfort metrics compared to the points that have lower classification probabilities. Figure 12 illustrates the Jack Interface and the calculated comfort metric for 2 different grasping configurations for a ladder object. We use Jacks comfort assessment tool to evaluate strictly the upper body (i.e. arm and wrist) postures of the final grasp results.

The first set of points are labeled as good grasping locations by the proposed algorithm with ≈ 0.98 classification probability , whereas the second configuration has a very low classification probability ≈ 0.01. The resulting comfort metrics are consistent with our expectations that the Jack Software returns higher comfort values for the first grasping configuration compared to the second one (Compare the values for the joint angles with re-

Copyright © 2016 by ASME

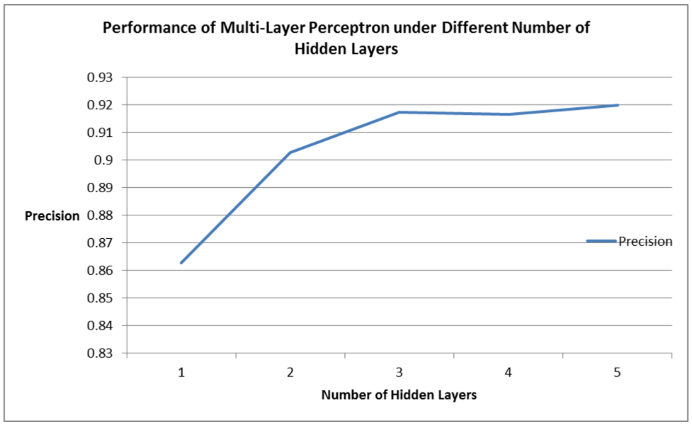


FIGURE 11. Precision values with varying number of hidden layers for MLP.

spect to given minimum and maximum values in red rectangular box). Furthermore, another interesting observation is the fact that learning based approach was able to capture the constraint that the distance between two grasping points can not be greater than the shoulder span of humans. This rule is actually captured by our learning model and applied for grasping location estimation. Additionally, our learning model was also able to capture the tendency that humans usually position their right hands higher than their left hands while lifting an object.

CONCLUSIONS

In this paper, we introduced a data-driven approach for estimating natural grasp point locations on objects that human interact with in industrial applications. The mapping between the feature vectors and 3D object geometries are dictated by grasping locations crowdsourcing from general public. Hence, our method can accommodate new geometries as well as new grasping location preferences. While our preliminary results are promising for a data-driven approach for the estimation of grasping locations, more studies on the feature selection may be required. Additionally, both the testing and training datasets needs to be increased in order cover a broader range of object families for grasp point estimations. Similarly, in this work, we assumed that the objects do not offer convenient handles for carrying purposes. A preprocessing algorithm might be implemented to check if the object contains such handles before running the data-driven estimation tool. Finally, integration of data-driven approaches with physics based models for grasping location estimation provide interesting research opportunities to incorporate material properties.

ACKNOWLEDGMENT

We thank Ulrich Raschke and Christina Cort for the valuable discussions and help for Siemens PLM Software Tecnomatix.

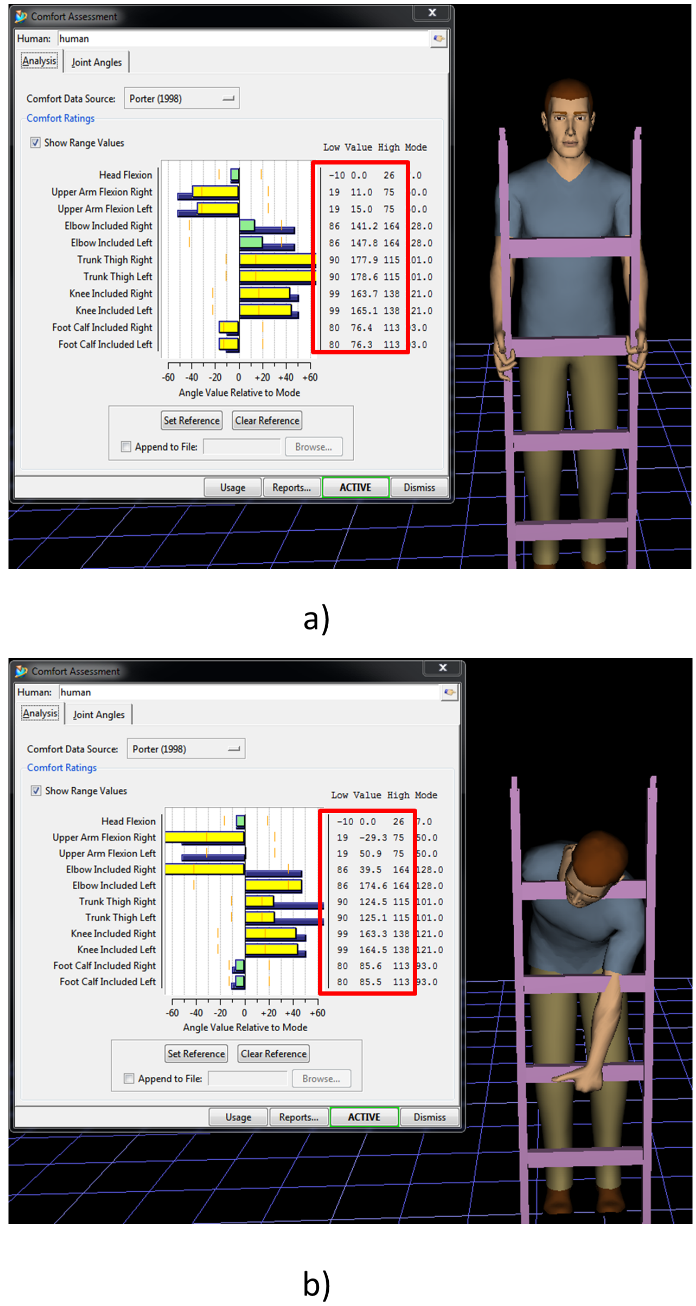


FIGURE 12. Comparison of comfort values for 2 different grasping configurations for a ladder object using Siemens Jack Software: Comfort metrics in (a) are within the identified ranges, whereas in (b) comfort metrics are outside of the predefined range by Porter [25].

REFERENCES

[1] Aydin, Y., and Nakajima, M., 1999. "Database guided computer animation of human grasping using forward and inverse kinematics". *Computers & Graphics,* *23*(1), pp. 145–

Copyright © 2016 by ASME

154.
[2] Goldfeder, C., and Allen, P. K., 2011. “Data-driven grasping”. *Autonomous Robots,* ***31***(1), pp. 1–20.
[3] Li, Y., Saut, J.-P., Cortés, J., Siméon, T., and Sidobre, D., 2011. “Finding enveloping grasps by matching continuous surfaces”. In Robotics and Automation (ICRA), 2011 IEEE International Conference on, IEEE, pp. 2825–2830.
[4] Touvet, F., Daoud, N., Gazeau, J. P., Zeghloul, S., Maier, M. A., and Eskiizmirliler, S., 2012. “A biomimetic reach and grasp approach for mechanical hands”. *Robot. Auton. Syst.,* ***60***(3), Mar., pp. 473–486.
[5] Sahbani, A., El-Khoury, S., and Bidaud, P., 2012. “An overview of 3d object grasp synthesis algorithms”. *Robotics and Autonomous Systems,* ***60***(3), pp. 326–336.
[6] Zhao, W., Zhang, J., Min, J., and Chai, J., 2013. “Robust realtime physics-based motion control for human grasping”. *ACM Trans. Graph.,* ***32***(6), Nov., pp. 207:1–207:12.
[7] Vasundara, M., and Padmanaban, K. P., 2013. “Recent developments on machining fixture layout design, analysis, and optimization using finite element method and evolutionary techniques”. *The International Journal of Advanced Manufacturing Technology,* ***70***(1), pp. 79–96.
[8] Wang, M., and Pelinescu, D., 2000. “Precision localization and robust force closure in fixture layout design for 3d workpieces”. In Robotics and Automation, 2000. Proceedings. ICRA ’00. IEEE International Conference on, Vol. 4, pp. 3585–3590 vol.4.
[9] Zheng, Y., Lin, M., and Manocha, D., 2011. “Efficient simplex computation for fixture layout design”. *Computer-Aided Design,* ***43***(10), pp. 1307 – 1318. Solid and Physical Modeling 2010.
[10] El-Khoury, S., Sahbani, A., and Perdereau, V., 2007. “Learning the natural grasping component of an unknown object”. In Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on, pp. 2957–2962.
[11] Sahbani, A., and El-Khoury, S., 2009. “A hybrid approach for grasping 3d objects”. In Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, pp. 1272–1277.
[12] Ying, L., Fu, J., and Pollard, N., 2007. “Data-driven grasp synthesis using shape matching and task-based pruning”. *Visualization and Computer Graphics, IEEE Transactions on,* ***13***(4), July, pp. 732–747.
[13] Yamane, K., Kuffner, J. J., and Hodgins, J. K., 2004. “Synthesizing animations of human manipulation tasks”. *ACM Trans. Graph.,* ***23***(3), Aug., pp. 532–539.
[14] Kim, V. G., Chaudhuri, S., Guibas, L., and Funkhouser, T., 2014. “Shape2pose: Human-centric shape analysis”. *ACM Trans. Graph.,* ***33***(4), July, pp. 120:1–120:12.
[15] Ye, Y., and Liu, C. K., 2012. “Synthesis of detailed hand manipulations using contact sampling”. *ACM Transactions on Graphics (TOG),* ***31***(4), p. 41.
[16] Zheng, Y., Lin, M., and Manocha, D., 2012. “On computing reliable optimal grasping forces”. *Robotics, IEEE Transactions on,* ***28***(3), June, pp. 619–633.
[17] Chinellato, E., Fisher, R., Morales, A., and del Pobil, A., 2003. “Ranking planar grasp configurations for a three-finger hand”. In Robotics and Automation, 2003. Proceedings. ICRA ’03. IEEE International Conference on, Vol. 1, pp. 1133–1138 vol.1.
[18] Roa, M. A., and Suárez, R., 2014. “Grasp quality measures: review and performance”. *Autonomous Robots,* ***38***(1), pp. 65–88.
[19] Ohbuchi, R., Minamitani, T., and Takei, T., 2003. “Shape-similarity search of 3d models by using enhanced shape functions”. In Theory and Practice of Computer Graphics, 2003. Proceedings, pp. 97–104.
[20] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H., 2009. “The weka data mining software: an update”. *ACM SIGKDD explorations newsletter,* ***11***(1), pp. 10–18.
[21] Quinlan, J. R., 2014. *C4. 5: programs for machine learning*. Elsevier.
[22] Breiman, L., 2001. “Random forests”. *Machine learning,* ***45***(1), pp. 5–32.
[23] Kohavi, R., 1996. “Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid.”. In KDD, Vol. 96, Citeseer, pp. 202–207.
[24] Kittur, A., Chi, E. H., and Suh, B., 2008. “Crowdsourcing user studies with mechanical turk”. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’08, ACM, pp. 453–456.
[25] Porter, J. M., and Gyi, D. E., 1998. “Exploring the optimum posture for driver comfort”. *International Journal of Vehicle Design,* ***19***(3), pp. 255–266.

Copyright © 2016 by ASME