ContactGrasp: Functional Multi-finger Grasp Synthesis from Contact

Samarth Brahmbhatt\textsuperscript{1,2}, Ankur Handa\textsuperscript{2}, James Hays\textsuperscript{1}, and Dieter Fox\textsuperscript{2}

Abstract—Grasping and manipulating objects is an important human skill. Since most objects are designed to be manipulated by human hands, anthropomorphic hands can enable richer human-robot interaction. Desirable grasps are not only stable, but also functional: they enable post-grasp actions with the object. However, functional grasp synthesis for high-dof anthropomorphic hands from object shape alone is challenging. We present ContactGrasp, a framework that allows functional grasp synthesis from object shape and contact on the object surface. Contact can be manually specified or obtained through demonstrations. Our contact representation is object-centric and allows functional grasp synthesis even for hand models different than the one used for demonstration. Using a dataset of contact demonstrations from humans grasping diverse household objects, we synthesize functional grasps for three hand models and two functional intents. The project webpage is \url{https://contactdb.cc.gatech.edu/contactgrasp.html}.

I. INTRODUCTION

Household objects are designed for use and manipulation by human hands. Humans excel at grasping and then performing actions with these objects. Enabling this skill for robots has the potential to unlock more productive and natural human-robot interaction. We take a step towards this by proposing ContactGrasp, a framework to synthesize functional grasps. Using object shape and contact demonstrations, ContactGrasp allows functional grasp synthesis for kinematically diverse hand models.

Recent work on robotic grasping of household objects focuses on using large amounts of data collected by trying random grasping actions, often using parallel-jaw or suction-cup end-effectors \cite{1, 2, 3, 4}. This approach generates lots of self-supervised data that enables training of robust grasp policies. However, the simplicity of the end-effectors has a big hand in enabling this data-collection strategy. In addition, such end-effectors are mostly suited for pick-and-place tasks and do not allow performing a post-grasp action (e.g., handing an object off, clicking the camera shutter button, switching on a flashlight, etc.). Functionality of a grasp is important because 1) it enables more natural collaboration in activities with humans, and 2) most household objects are designed with specific functional grasps in mind (e.g., spray bottle has contoured neck and squeezer, hammer has a long handle, etc.).

A large body of work addresses grasp synthesis from object geometry. These approaches model the hand as a kinematic tree of rigid mesh parts and intelligently sample its configuration space to synthesize a set of stable grasps \cite{5, 6, 7}. However, these grasps lack the notion of functionality and most of them are far from how a human would grasp the object (see Figure 5 for examples).

Other approaches have utilized human demonstrations to close this gap. The human hand pose (captured by data gloves in \cite{8}, and by visual recognition in \cite{9}) is mapped by hand-engineered transformations to a similar robot end-effector pose. Kinesthetic teaching \cite{10} can deliver demonstrations directly in the target hand model space. However, such kinematic re-targeting methods are tied to a specific end-effector and require careful analysis to develop mappings to new end-effectors. Approaches that record the human hand pose suffer from the additional difficulty of orienting the hand pose w.r.t. the object (which requires embedding an additional 6-DOF magnetic tracker in the object \cite{11}). In addition, they lack a clear objective to reproduce the demonstrated contact, as we show in Section VI. Analysis of contact during human grasping has shown that humans prefer to contact specific areas during functional grasping \cite{12}.

This motivates the development of an object-centric approach that is not tied to a specific end-effector, and that emphasizes the reproduction of demonstrated contact. Towards this end, we propose ContactGrasp, a framework which synthesizes grasps from both object geometry and contact on the object surface. Contact can be specified manually, through human demonstrations (e.g. ContactDB \cite{12}), or a combination of both. We show in Section VI that Con-
tactGrasp can be used to synthesize grasps that reproduce the demonstrated contact for multiple different hand models. Grasp synthesis from contact has the drawback that multiple hand configurations can sometimes result in the same contact pattern. To address this, ContactGrasp adopts a sample-and-rank approach which outputs a ranked set of grasps. We show qualitatively and quantitatively in Section VI that the desired grasp can be found among the top ranked grasps in this set.

To summarize, we make the following contributions in this paper:

- Develop a multi-point contact representation that supports efficient grasp synthesis.
- Propose a sample-and-rank approach for functional grasp synthesis from object shape and contact that can work with multiple hand models.

II. RELATED WORK

Grasp synthesis has been widely studied from many perspectives. Analytic approaches [13], [14], [15], [16], [17], [18], [19], [20] synthesize grasps(s) from object shape, which usually guarantee stability within their set of assumptions. These assumptions include perfect object models, rigid body approximation of the hand, Coulomb friction, simplified contact models, etc. While such approaches contribute valuable theoretical analysis of grasping, their simplifying assumptions are often quite restrictive. This makes purely analytic algorithms difficult to deploy in the real world.

In contrast, data-driven approaches rely on machine learning techniques to learn features in some representation of the object (RGB image, 3D mesh, etc.) that can be used to predict grasps. For example, Li et al [21] compute shape features for both objects and grasping hands, and use nearest-neighbor retrieval to synthesize grasps for novel objects. Newer deep-learning based algorithms typically require large amounts of training data, which is expensive to collect and label. In addition, they are limited to simple end effectors like parallel jaw grippers to enable efficient training data collection. For example, [22], [23], [24] predict grasps for a parallel jaw gripper from a dataset of manually annotated images. Some recent works like Pinto et al [4] have used self-supervision to automate data collection, again for a simple parallel jaw gripper. We refer the reader to [25] for an extensive survey of data-driven grasp synthesis.

Hybrid approaches use analytic criteria (e.g. [26], [27]) to sample a large number of grasps in a simulator like GraspIt! [28]. Real-world demonstrations in various forms are then used to process these samples: filter them or train a machine learning algorithm to predict success. For example, Mahler et al [1], [3], [2], [4] execute those grasps with a robot and record success/failure. Song et al [29] learn a Bayes Net to jointly model post-grasp task and discretized hand pose. Synthetic data generated using a grasp planner is labeled manually for post-grasp task suitability. However, the algorithm is not aware of contact and the pose discretization often results in predictions that are not in contact with the object.

Grasp Synthesis from Contact: The work of Hamer et al [30] is perhaps closest to ours. They record human demonstrations of grasping by in-hand scanning to get both object and hand pose. Contact is approximated as a single point per fingertip. Contact points are aggregated on the object surface and used to form a prior for hand pose synthesis and tracking. Varley et al [31] replace human demonstrations with synthetic data by running a grasp planner in simulation and recording the fingertip contact points. Ye at al [32] develop an algorithm to sample physically plausible contact points between the hand the object, given the wrist and object pose from a motion-capture system. These sampled points are then used to synthesize realistic-looking hand poses. In contrast, ContactGrasp allows realistic multi-point contact (see Figure 3), allows using kinematically diverse hand models, and demonstrates good performance across more complex and numerous objects.

III. CONTACT MODEL AND HUMAN DEMONSTRATIONS

As mentioned in Section I ContactGrasp can leverage human demonstrations of grasp contact to synthesize similar grasps for various hand models. It has been shown that humans contact grasped objects with not only fingertips, but also the palm and non-tip areas of the fingers. Hence our contact model and grasp synthesis algorithm supports multi-point contact.

We define the contact map $c$ as a set of $N$ points $p_i$ sampled uniformly at random on the object surface, with a contact value $c_i$ of +1 (attractive) or −1 (repulsive). Contacted points in the demonstration are marked attractive, while others are marked repulsive. Repulsive points allow ContactGrasp to exploit negative information [33]. Figure 3 shows an example of the contact map for the ‘flashlight’ object, with attractive points in green and repulsive points in red.

A. Human Contact Demonstrations

The contact model specified above supports manual specification. However, most of our experiments are performed using real-world contact maps from the ContactDB.
dataset [12]. ContactDB uses a thermal camera to observe the thermal after-prints left by heat transfer from hand to object during grasping. It is thus able to texture the object mesh surface with high-resolution contact maps. Participants grasp the objects with one of two post-grasp functional intents: handoff or use. Since ContactDB contact maps have continuous values \( t(p_i) \in [0, 1] \), we threshold them at \( \tau_t \) to determine whether a point is attractive or repulsive. Figure 3 shows an example of human contact demonstration from the ContactDB dataset.

\[
c_i = \begin{cases} 
+1, & \text{if } t(p_i) \geq \tau_t \\
-1, & \text{otherwise} 
\end{cases}
\]  

(1)

IV. HAND MODELS

![Hand models](image)

Fig. 4: Hand models used in our experiments, along with joint axes and special thumb contact point. Left: HumanHand [28], middle: Allegro Hand [34], right: Barrett Hand [35]

Given the contact map on the object surface, an articulated hand model is used to set up an optimization problem which seeks a hand pose that agrees with the observed contact map. We use three kinematically diverse hand models to show that the object-centric contact representation of ContactGrasp allows grasp synthesis with diverse hand models (See Table I and Figure 4). Each model is a kinematic tree, with parts modelled by rigid meshes. In addition to the articulation DOFs \( d_i \) the overall position of the hand in 3D space is defined by a 6 DOF rigid body transform \( T \). We denote the full pose of the hand as \( \Phi = (T, \{d_i\}_{i=1}^D) \), where \( D \) is the number of articulation DOFs of the hand. We compute a signed distance field (SDF) around each hand part and attach the number of articulation DOFs of the hand. We compute a signed distance field (SDF) around each hand part and attach the number of articulation DOFs of the hand. We compute a signed distance field (SDF) around each hand part and attach the number of articulation DOFs of the hand.

V. CONTACTGRASP: GRASP SYNTHESIS

In this section, we describe the algorithm to synthesize grasps from object geometry and a contact map. Grasp synthesis consists of estimating the full configuration \( \Phi \) of the hand, which includes the 6-DOF ‘palm’ pose as well as joint values. This is done through an appropriately initialized nonlinear optimization. Figure 5 shows an overview of the entire algorithm.

A. GRASP OPTIMIZATION

Intuitively, the objective of this optimization is to encourage contact at attractive points and discourage contact at repulsive points in the contact map. The full objective function consists of three terms, inspired from [36] (see Figure 6 for a visual example of each term):

1) GRASP TERM: The grasp term attracts (resp. repels) the closest hand segment to every attractive (resp. repulsive) point in the contact map. For a given contact map \( c \) and hand pose \( \Phi \), the term is stated as:

\[
L_{grasp}(\Phi|c) = \sum_{i=1}^{N} \lambda_a[c_i = +1]SDF_{k_i}(p_i)^2 - \lambda_r[c_i = -1]SDF_{k_i}(p_i)^2
\]  

(2)

Where \( k_i \) is the index of the hand segment that is closest to point \( p_i \), \( SDF_{k_i}() \) is the signed distance function (SDF) associated with hand segment \( k \), and \( [\cdot] \) is the indicator function. \( \lambda_a \) and \( \lambda_r \) are hyperparameters controlling the strength of attractive and repulsive points.

2) THUMB CONTACT TERM: It is well known that the thumb is especially important in human grasps [37]. To account for this, we specify a point \( p_{thummb} \) on the thumb (or the part closest resembling the thumb) in the hand model (yellow dots in Figure 4). The thumb contact term encourages \( p_{thummb} \) to be in contact with the object:

\[
L_{thumb}(\Phi) = \lambda_t SDF_{object}(p_{thummb})^2
\]  

(5)

where \( SDF_{object}(\cdot) \) is the signed distance function associated with the object, and hyperparameter \( \lambda_t \) controls the strength of this term.

3) INTERSECTION TERM: The intersection term \( L_{int}(\Phi) \) discourages intersection of the hand model with the object and self-intersection among segments of the hand, and is the same as the one used in [36]. Its strength is controlled by the hyperparameter \( \lambda_i \).

| Model       | Fingers                       | Joints | Articulation DOFs |
|-------------|-------------------------------|--------|-------------------|
| HumanHand   | 4 fingers, 1 thumb            | 15     | 20                |
| Allegro     | 3 fingers, 1 thumb            | 12     | 16                |
| Barrett     | 3 digits                      | 7      | 4                 |

TABLE I: Hand models used in our experiments.
Fig. 5: Overview of the ContactGrasp algorithm. GraspIt! [5] is used to sample random grasps for the object geometry. These are then refined and ranked for agreement with a human-demonstrated contact map to synthesize functional grasps.

Fig. 6: Geometry of the activation function for a repulsive point $p_i$ in the contact map for an object.

4) Optimization: The full objective function for grasp optimization is given by

$$L(\Phi | c) = L_{\text{grasp}}(\Phi | c) + L_{\text{thumb}}(\Phi) + L_{\text{int}}(\Phi) \quad (6)$$

We use Dense Articulated Real-time Tracking (DART) [33] to minimize Eq. 6 and get the optimized hand pose. Specifically, we modify the Contact Prior mechanism in DART [36] to 1) support repulsive points, and 2) remove the depth-map observation term. DART approximately minimizes Eq. 6 by running the Levenberg-Marquadt algorithm (see [33] for more details).

B. Initializing the Grasp Optimization

Since the Levenberg-Marquadt grasp optimization is local, and search in the high-dimensional hand pose space has many local minima, providing good initialization to the optimizer becomes important. Towards this end, we develop an algorithm to sample diverse grasps for the object geometry which are agnostic to the contact map, using the publicly available GraspIt! grasp planner [28], [5]. These are later ranked for agreement with the contact map using the residue after minimizing Eq. 6.

Specifically, we seed the planner with a coarse grasp $\phi = (a, \theta, d)$, where $a$ is the approach point on the object surface, $\theta$ is the roll angle around approach vector and $d$ is the distance from object surface. We use the negated surface normal $-n$ at the approach point $a$ as the approach vector, making since the hand approach anti-parallel to the object surface normal (similar to [38]). Approach points $a$ are sampled uniformly at random over the object surface, whereas $\theta$ and $d$ are set from the discrete sets $\{0, 90, 180, 270\}$ and $\{0 \text{ cm}, 1 \text{ cm}, 2 \text{ cm}, 3 \text{ cm}\}$ respectively. GraspIt!'s Simulated Annealing planner then runs for 45K iterations for
each seed to optimize the Contact Energy cost function [5], exploring in a cone around the specified approach vector. We pick the top 2 grasps after each run of the planner, and add them to the set of sampled diverse grasps $D$. Note that $D$ contains full grasps, since GraspIt! provides a bridge to convert coarse grasps $\phi$ to full grasps $\Phi$ through its planner. Figure 5 shows some grasps sampled in this manner for a cellphone.

The last step is to refine and rank the grasps in $D$ by running grasp optimization on each of them, and considering the residual after convergence (Eq. 6) as the negative score. Most of the grasps in $D$ result in large residuals because they are agnostic to the contact maps and hence can be out of the basin of convergence of the local optimization. Hence this ranking process causes the correct hand pose to show up among the top-ranked, from which it can be easily identified.

To summarize, the process described in this section is recover the correct full hand pose that agrees with a given contact map on an object. Manually annotating the full hand pose prohibitively expensive because of the high dimensionality.
VI. RESULTS

We use the ContactGrasp algorithm (Section V) to synthesize grasps for a diverse set of household objects for two different post-grasp functional intents: using the object, and handing it off. Grasps are synthesized for the three hand models described in Section IV. We use human contact demonstrations for functional grasps from the ContactDB dataset [12]. 25 objects in ContactDB have demonstrations for both using the object and handing it off. From these, we select a subset of 19 objects (see supplementary video for a list) for which bi-manual grasps were not observed. Note that the grasp optimization strategy described in Section V supports bi-manual grasps. However, we focus on single-handed grasps in this paper owing to lack of left-handed hand models and need for a more complex initialization strategy for the optimization.

We set the hyperparameter $\tau_t$ (threshold on contact map value) to 0.3, $\lambda_a$ (attractive contact point strength) to 150.0, $\lambda_r$ (repulsive contact point strength) to 20.0, $\lambda_t$ (thumb contact point strength) to 25.0, and $\lambda_i$ (intersection term strength) to 100.0.

A. Qualitative results

Figures 8, 9, and 10 show the synthesized grasps for the HumanHand, Allegro hand and Barrett hand respectively, for 8 objects and 2 functional intents (see supplementary video for other objects). They demonstrate functional grasps e.g., for the ‘use’ intent, the flashlight button is easily accessible by the thumb, fingers rest on mouse click buttons, knife is held by the handle. In contrast, the top-ranked grasps from GraspIt! are shown in Figure 11. They are stable but do not demonstrate functionality, especially for the ‘use’ intent e.g., flashlight button is not accessible by the thumb, finger touches knife blade, cellphone screen and wineglass opening are blocked by palm.
TABLE II: Median rank of the correct grasp (lower is better).

| Intent   | End-effector     | ContactGrasp rank L | GraspIt! rank L |
|----------|------------------|----------------------|-----------------|
| use      | HumanHand [28]   | 3(0.13%)             | 4484(34.41%)    |
|          | Allegro [34]     | 22(0.69%)            | 1289(40.28%)    |
|          | Barrett [35]     | 6(0.19%)             | 2362(74.09%)    |
| handoff  | HumanHand [28]   | 1(0.01%)             | 3751(38.37%)    |
|          | Allegro [34]     | 22(0.56%)            | 1258(38.20%)    |
|          | Barrett [35]     | 24(0.75%)            | 830(25.94%)     |

Table III shows further quantitative results in the form of the \( L_{\text{grasp}} \) value (see Eq. 3) for the synthesized grasp as well as the top-ranked grasp by GraspIt!’s Contact Energy metric. \( L_{\text{grasp}} \) measures the grasp’s disagreement with the demonstrated contact map, and is significantly lower for ContactGrasp grasps. This verifies that ContactGrasp produces grasps that are significantly closer to the human demonstrations than GraspIt!, which is agnostic to the demonstrations.

In addition, we implement the kinematic retargeting algorithm of Tosun et al. [39] for mapping the synthesized human hand grasp to the Allegro model. Each finger is treated as a separate kinematic chain (sampled with 50 points) to be retargeted. The human pinky finger is discarded. The \( L_{\text{grasp}} \) values for these mapped grasps are higher than those for the ContactGrasp Allegro grasps as well as human grasps. This shows that deterministic mapping of grasps between hand models does not reliably reproduce contact, supporting our motivation for developing ContactGrasp.

C. Failure Cases

Figure 12 shows failure cases of ContactGrasp. These occur when the target hand model does not possess the geometry to be able to achieve a complicated control pattern, or when the grasp sampling stage is not exhaustive enough to sample fine manipulation behaviors like fingers through holes.

TABLE III: Disagreement of the ContactGrasp and GraspIt! grasps from human-demonstrated contact, measured by \( L_{\text{grasp}} \) (see Eq. 3, lower is better). Human-Allegro indicates grasps mapped from human to the Allegro model using [39].

| Intent   | End-effector | ContactGrasp \( L_{\text{grasp}} \) | GraspIt! \( L_{\text{grasp}} \) |
|----------|--------------|-------------------------------------|-------------------------------|
| use      | HumanHand    | -0.07                               | 0.17                          |
|          | Allegro      | -0.06                               | 0.32                          |
|          | Human-Allegro| 0.08                                | 0.12                          |
|          | Barrett      | -0.03                               | 0.19                          |
| handoff  | HumanHand    | -0.14                               | 0.19                          |
|          | Allegro      | -0.11                               | 0.41                          |
|          | Human-Allegro| 0.05                                | 0.15                          |
|          | Barrett      | -0.04                               | 0.22                          |

Fig. 12: Failure cases for ContactGrasp. Left: Random grasp sampling is unlikely to produce a grasp with fingers in narrow holes for objects like scissors. Contact-based local optimization is then likely to get stuck in local minima. Right: Some end effectors lack the structure to produce complicated functional grasps e.g. using a mouse by resting a finger on the click buttons.

Table III shows further quantitative results in the form of the \( L_{\text{grasp}} \) value (see Eq. 3) for the synthesized grasp as well as the top-ranked grasp by GraspIt!’s Contact Energy metric. \( L_{\text{grasp}} \) measures the grasp’s disagreement with the demonstrated contact map, and is significantly lower for ContactGrasp grasps. This verifies that ContactGrasp produces grasps that are significantly closer to the human demonstrations than GraspIt!, which is agnostic to the demonstrations.

In addition, we implement the kinematic retargeting algorithm of Tosun et al. [39] for mapping the synthesized human hand grasp to the Allegro model. Each finger is treated as a separate kinematic chain (sampled with 50 points) to be retargeted. The human pinky finger is discarded. The \( L_{\text{grasp}} \) values for these mapped grasps are higher than those for the ContactGrasp Allegro grasps as well as human grasps. This shows that deterministic mapping of grasps between hand models does not reliably reproduce contact, supporting our motivation for developing ContactGrasp.

C. Failure Cases

Figure 12 shows failure cases of ContactGrasp. These occur when the target hand model does not possess the geometry to be able to achieve a complicated control pattern, or when the grasp sampling stage is not exhaustive enough to sample fine manipulation behaviors like fingers through holes.

VII. CONCLUSION

To summarize, we develop a multi-point contact model in this paper, which can plug-and-play with kinematically diverse hand models to synthesize grasps. These grasps can be modulated by hand-object contact demonstrations to be functional, supporting post-grasp actions like using the object or handing it off. We show that our approach ContactGrasp, which directly optimizes for contact, is superior than other approaches which kinematically re-target observed human grasps to the target hand model. We demonstrate the effectiveness of ContactGrasp by synthesizing functional grasps for 3 significantly diverse hand models, 19 household objects, and 2 functional intents.

REFERENCES

[1] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, “Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics,” 2017.
[2] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, “Dex-net 3.0: Computing robust robot suction grasp targets in point clouds using a new analytic model and deep learning,” arXiv preprint arXiv:1709.06670, 2017.
[3] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, “Learning ambidextrous robot grasping policies,” Science Robotics, vol. 4, no. 26, p. eaau4984, 2019.
[4] L. Pinto and A. Gupta, “Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours,” in 2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2016, pp. 3406–3413.
[5] M. Ciocarlie, C. Goldfeder, and P. Allen, “Dimensionality reduction for hand-independent dexterous robotic grasping,” in 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems.
[6] M. A. Roa, M. J. Argus, D. Leidner, C. Borst, and G. Hirzinger, “Power grasp planning for anthropomorphic robot hands,” in 2012 IEEE International Conference on Robotics and Automation. IEEE, 2012, pp. 563–569.
