2022 IEEE International Conference on Robotics and Automation (ICRA)
May 23-27, 2022. Philadelphia, PA, USA

Contact Transfer: A Direct, User-Driven Method for Human to Robot Transfer of Grasps and Manipulations

Arjun Lakshmipathy[1] Dominik Bauer[2] Cornelia Bauer[2] Nancy S. Pollard[1,2]

***Abstract*— We present a novel method for the direct transfer of grasps and manipulations between objects and hands through utilization of contact areas. Our method fully preserves contact shapes, and in contrast to existing techniques, is not dependent on grasp families, requires no model training or grasp sampling, makes no assumptions about manipulator morphology or kinematics, and allows user control over both transfer parameters and solution optimization. Despite these accommodations, we show that our method is capable of synthesizing kinematically-feasible whole hand poses in seconds even for poor initializations or hard-to-reach contacts. We additionally highlight the method's benefits in both response to design alterations as well as fast approximation over in-hand manipulation sequences. Finally, we demonstrate a solution generated by our method on a physical, custom-designed prosthetic hand.**

I. INTRODUCTION

Transfer of human grasps and manipulation demonstrations to robot hands has been a long standing and challenging problem in the robotics community. Though numerous anthropomorphic hands have been developed (e.g., [1], [2], [3], [4], [5], [6]), no current manipulator yet matches the dexterous capabilities of the human hand. In addition, simpler hands are often desired based on considerations such as weight, cost, and ease of manufacture and control.

Towards this end, many research efforts have endeavored to develop methods for robustly transferring dexterous capabilities to robot hands with characteristics that may differ from the hand of the human demonstrator. The majority of existing methods fall into pure learning, hand pose tracking and retargeting, or a hybrid of both. Pure reinforcement learning methods have produced convincing results even for complex manipulations [7], [8], [9], but occasionally yield policies with unexpected or undesirable behaviors due to the abstract nature of the reward function. Tracking methods instead capture the movements of both the human hand and the object being manipulated with the intention of grounding the learned policy [10] or directly transferring the motion to the target manipulator [11], [12]; unfortunately, occlusion and noise during the tracking process, as well as morphological differences between the human and target hand, complicate the process. Summarily, it is worth noting that these techniques focus on matching observations in hand or object space as opposed to the interaction interface between the two bodies.

Contacts instead present an alternate and often complementary means of encoding grasps and manipulations. Historical works have used contact information to prune kinematically-infeasible solutions [13], match object and hand shapes [14], [15], and drive physics based reasoning [16]. Recent works exploit entire contact regions to synthesize grasps for morphologically diverse manipulators [17], optimize coarse poses [18], and help train pose generation models [19]. The catch, however, is that most existing techniques incur one of three possible drawbacks: single point approximation [13], [14], [15], [16], strong dependence on grasp dependent hand shape priors [18], [19], or expensive computation due to exhaustive grasp sampling [17].

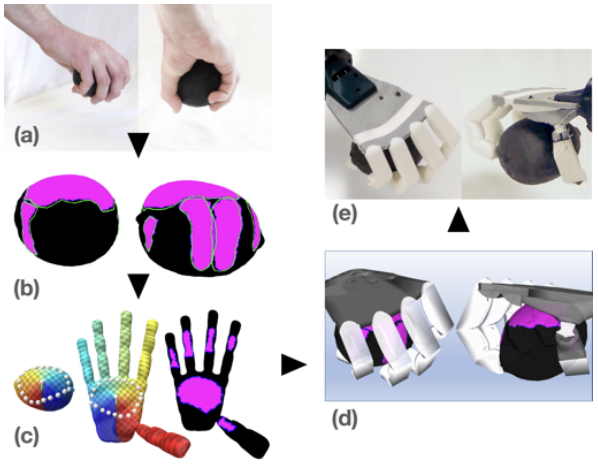


Fig. 1: High level overview of our framework. (a) Human demonstrations of grasps and manipulations are collected and (b) reconstructed using existing techniques. (c) Contact regions are then transferred from object to target manipulator, and (d) the whole hand kinematic pose is computed from the transferred contacts via optimization. (e) The robot hand then utilizes the final articulated solution.

In response to these drawbacks, we present a geometric framework for the *direct*, intuitive, and rapid transfer of contacts observed and collected from objects to morphologically diverse hands that fully preserves contact shape and is not dependent on external hand pose datasets or grasp sampling. More specifically, this paper makes the following contributions:

1) a novel approach to *directly* transfer contact areas of arbitrary shape to morphologically diverse manipulators in a fast and fully customizable manner
2) a cheap optimization strategy to quickly synthesize kinematically-feasible grasps and in-hand manipulations by exploiting correspondences generated by the transfer process

Our approach is illustrated in Figure 1. We demonstrate that our method can accommodate hands with high degrees of freedom and contact areas of widely varying shapes. We also show that our framework produces solutions that

[1]Computer Science Department, Carnegie Mellon University
[2]Robotics Institute, Carnegie Mellon University

are robust to poor initial conditions as well as difficult-to-reach contacts. The plug-and-play capability awarded by our transfer method makes it especially well suited for manipulator prototyping, which we illustrate in concluding experiments. Finally, we demonstrate a synthesized grasp solution on a physical, custom-designed prosthetic hand.

II. RELATED WORK

Numerous prior efforts have targeted the problem of recovering and optimizing manipulator pose from contacts, the majority of which can be broadly classified under either single point or area based approaches.

Single point methods use individual object points as the basis for extracting the corresponding hand configuration. Mapping individual object and hand contact points allows for the use of traditional inverse kinematics [13], [20]; however, in grasps that contain a large number of contacts, the problem is generally overconstrained and may instead be best solved using an optimization approach where the user can specify the importance of different tradeoffs (e.g., match contacts, match hand pose, etc.). Optimization approaches can additionally incorporate kinematic constraints and endeavor to find least-squares [20] or force-closure inducing [21], [16] solutions. But while these approaches are fast and straightforward, single point contacts are an idealized representation of real-world situations. Additionally, approximating contact areas as single points sacrifices useful information.

Consequently, some recent research efforts have turned instead to utilizing contact regions. Though a number of existing works have considered contacts purely through geometric reasoning, including independent contact regions [22], [23] and directly mapping individual links to mesh slices [24], recent data driven works have produced high fidelity contact maps by instead capturing contacts directly from human demonstrations [25], [26] or synthesizing them from vision based retargeting and simulation [19]. These maps have subsequently been utilized to synthesize grasps for morphologically diverse manipulators [17] as well as optimize coarse estimates [18], [19]; however, owing to the difficulty of mapping contact regions to articulated manipulators, these methods are dependent on priors or exhaustive grasp sampling, which thus limit their robustness and speed.

This paper directly addresses the mapping problem by proposing a method to transfer object contact areas to any manipulator in a manner that fully preserves relative contact point distances regardless of underlying geometry, making it possible to combine the benefits of single point and area based techniques. We show that the resulting contact area correspondence enables the use of a simple, yet robust optimization to compute the best matching, kinematically-feasible hand pose. Our technique does not require sampling or external datasets and can be used on any hand design, and is intentionally designed to allow full user control over both the transfer and optimization process, which we show leads to fast, robust, and qualitatively reasonable results for both static grasps and in-hand manipulations. We demonstrate results for a range of simulated robot hands, a graphical human hand, and a real custom-designed prosthetic hand.

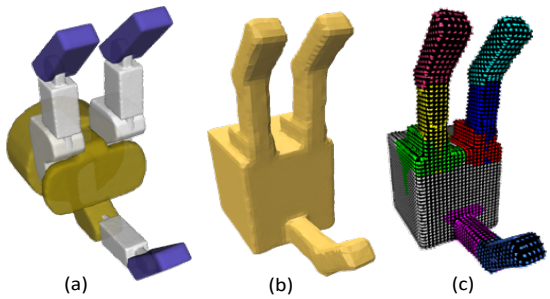


Fig. 2: The (a) articulated Barrett Hand can be (b) approximately skinned by stitching together primitive collision geometries. (c) The projection process partitions the skin into groups based on the link locations of the articulated hand.

III. PROCEDURE

Our algorithm consists of several stages, roughly illustrated in Figure 1. First, thermal traces of contact patches generated from human grasps and manipulations are captured and reconstructed onto object meshes using techniques from prior works [25], [27]. The process outlined in this paper then transfers contact patches from object surfaces onto a "skin" of the manipulator, and subsequently projects contacts from the skin onto the articulated hand. Finally, we utilize the mapping to compute grasps and manipulations with the target manipulator that best match the correspondence. Each stage is described in detail in the proceeding subsections.

A. Contact Capture, Reconstruction, and Tracing

Contact regions resulting from human grasps are first captured using a combination of thermochromic spray painted objects and RGBD imaging, and subsequently mapped onto object mesh surfaces using a reconstruction process detailed in existing works [25], [27]. Regions are then annotated and processed into *patches*, which are represented by a single *root* (PR) mesh vertex and collection of *boundary* (PB) mesh vertices. Boundaries are then down-sampled to produce *interpolation boundary* (IB) sets of 20-30 mesh vertices, which are then traced from initial to final positions in the case of manipulations. Detailed definitions and process descriptions are available in [27].

B. Skinning and Projection

Geometry processing algorithms are typically designed to operate over connected surfaces; however, the large majority of robotic systems are articulated, comprised instead of independent surface links connected by joints. These surfaces may also exhibit degenerate characteristics, including non-manifold edges, obtuse triangles, and near-zero area faces, which, combined with articulation, render many geometry processing algorithms useless.

Therefore, we instead construct a fully manifold "skin" of each manipulator and then project vertices from the skin back on to the original articulated geometry using KD tree [28] approximation. Figure 2 illustrates the skinning and

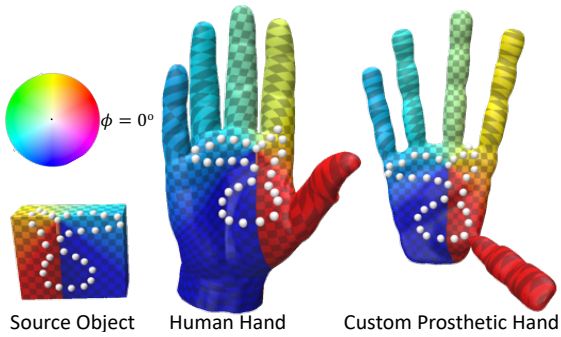


Fig. 3: Illustration of our logmap-based contact patch transfer process. Accurate relative distances and angles at all mesh vertices enable transfer of even large, irregularly shaped patches (outlined by white dots) to widely differing hand geometries.

projection process. In particular, we note that skins can be constructed from even coarse mesh approximations such as primitive collision geometries. Post projection, points on the skin which have no nearby articulated vertex below threshold ϵ are ignored during optimization. We use Blender [29] with the Phobos attachment to construct skins and the Open3d library [30] to perform the projection.

C. Contact Patch Transfer

The key idea of our transfer process is that patch shapes on the object and manipulator are equivalent, regardless of the surface on to which they are projected. More formally, the distance and direction of the IB points from the root should be preserved under a geometry independent discrete logarithmic map $\mathbf{L}_{r,\phi}$ [31], [32]. We therefore require four parameters, all of which are visually adjustable by the user: the patch's root vertex on the hand skin $\mathbf{v}_{r,h}$ and object $\mathbf{v}_{r,o}$, as well as the tangent vector direction from the hand $\vec{\mathbf{v}}_{r,h}$ and object $\vec{\mathbf{v}}_{r,o}$ in which to begin the sweep. The corresponding IB set on the hand can then be entirely computed as:

$$\begin{aligned} \mathbf{v}^*_{IB_i,h} = \arg\min_{\mathbf{v}_h} \quad & \|\mathbf{L}_{r,\phi}(\mathbf{v}_h) - \mathbf{L}_{r,\phi}(\mathbf{v}_o)\|_2^2 \\ \text{s.t.} \quad & \mathbf{L}_{r,\phi}(\mathbf{v}_h) = f(\mathbf{v}_{r,h}; \vec{\mathbf{v}}_{r,h}) \\ & \mathbf{L}_{r,\phi}(\mathbf{v}_o) = f(\mathbf{v}_{r,o}; \vec{\mathbf{v}}_{r,o}) \end{aligned} \tag{1}$$

Finding these correspondences thus requires computation of logmap coordinates (r, ϕ) for all vertices on both object and skin. Naively, these quantities can be computed by tracing geodesics from the root to all vertices (r) and propagating the root tangent vector via parallel transport (ϕ) [33]; unfortunately, these operations are notoriously expensive. The Vector Heat Method proposed by Sharp et. al. [32], however, reduces to seconds the parallel transport of *both* divergence and root tangent vectors through the use of short time heat diffusion. As Figure 3 illustrates, this method can successfully transfer patches of arbitrary shape on to widely differing surface geometries, can be performed in seconds, and guarantees both relative distance and angle preservation. We use the Polyscope viewer [34] and Geometry Central library [35] to enable user selection of $(\mathbf{v}_{r,h}; \mathbf{v}_{r,o}; \vec{\mathbf{v}}_{r,h}; \vec{\mathbf{v}}_{r,o})$, solve Eq. 1, and export the final mapping.

D. Extension to Manipulations

We note that the aforementioned procedure can only transfer static grasps; however, manipulations require dynamic contacts capable of moving, appearing, or disappearing from the hand at any time. To account for this behavior, we first transfer initial and final grasps and then adopt the procedure of Lakshmipathy et. al. [27] to evolve patches on the hand in conjunction with evolution of patches on the object.

E. Optimization Procedure

Computing the associated IB mappings between the object and hand enabled use of a straightforward, three-term optimization problem to determine the best kinematic hand pose $\boldsymbol{\theta}^*$:

$$\begin{aligned} \boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \quad & \textstyle\sum_i^M (\lambda_d \Gamma_i^D + \lambda_n \Gamma_i^N) + \sum_j^J \lambda_p \Gamma_j^P \\ \text{s.t.} \quad & \boldsymbol{\theta}_L \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}_U \end{aligned} \tag{2}$$

where $\boldsymbol{\theta}$ is the degree of freedom vector, M is the total number of corresponding contact vertices, $J = |\boldsymbol{\theta}|$, $\boldsymbol{\theta}_L$ and $\boldsymbol{\theta}_U$ are the lower and upper bounds of each degree of freedom respectively, Γ_i^D, Γ_i^N, and Γ_j^P are the distance, normal, and prior pose deviation penalty terms for each corresponding pair of points i and joint j respectively, and λ_d, λ_n, and λ_p are weighting hyperparameters. Each penalty term is elaborated upon in the proceeding paragraphs.

First, we introduce $\Gamma_{D,i}$ to minimize the L_2 distance between each pair i of corresponding hand and object contacts:

$$\Gamma_{D,i} = ||\mathbf{p}_{o,i} - \mathbf{p}_{h,i}(\boldsymbol{\theta})||_2^2 \tag{3}$$

and $\Gamma_{N,i}$ to encourage anti-alignment of vertex normals:

$$\Gamma_{N,i} = (1 + \mathbf{n}_{h,i}(\boldsymbol{\theta}) \cdot \mathbf{n}_{o,i})^2 \tag{4}$$

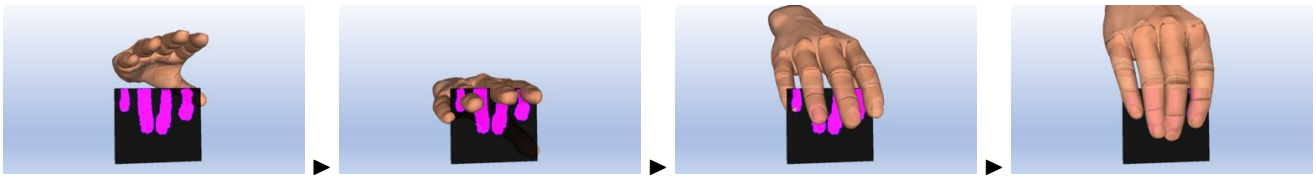
where each pair of object and hand contact points are denoted by o, i and h, i respectively, and hand point locations as well as vertex normal orientations are determined by the current hand pose $\boldsymbol{\theta}$.

We additionally introduce a third term to penalize deviation from $\boldsymbol{\theta}_P$, a hand pose prior:

$$\Gamma_{P,i} = ||\boldsymbol{\theta} - \boldsymbol{\theta}_P||_2^2 \tag{5}$$

The prior is initialized in the same way for all examples and serves as a user control for customization. At the start of the optimization, $\boldsymbol{\theta}_P$ is set to the default pose, thereby penalizing rest pose deviation. During subsequent calls, $\boldsymbol{\theta}_P$ is set to the optimal solution from the last set of iterations; however, if the user edits the default guess (e.g. moves a finger, drags the palm, etc.), $\boldsymbol{\theta}_P$ is instead set to the user edited pose. As a result, $\Gamma_{P,i}$ evolves as the optimization proceeds to reflect current progress and user direction.

It is worth noting that Eq. 2 does not contain a collision penalty term as suggested by several prior works [17], [36], [37]. Omission of this term is intentional and ultimately led to both substantial drops in solution discovery time as well



(a) (b) (c) (d)

Fig. 4: Optimization progress for finding a feasible box power grasp at (a) 0, (b) 250, (c) 800, and (d) 1,000 iterations.

as improvements in the proposal of solutions for difficult contact maps (see Figure 8).

The fully differentiable nature and cheap computation cost of Eq. 2 allows utilization of fast, gradient based solvers even using numerical gradient approximation; however, we empirically found that different solvers provided slightly different solutions. We ultimately settled on the Method of Moving Asympototes [38] for generating the majority of our results, but advocate implementing multiple solvers and allowing the user to choose between results. Each method was capped at 1,000 iterations per optimization call to intermittently update $\boldsymbol{\theta}_P$ and allow the user to edit intermediate estimates using IK. Figure 4 illustrates successive guesses over the course of a single optimization call. We used the NLOpt library [39] for defining and solving Eq. 2.

IV. RESULTS

Similar to Brahmbhatt et. al. [17], we performed a series of experiments using four different manipulators - a 20 degree of freedom human hand [40], an Allegro hand [1], a Barrett hand [41], and a custom-made anthropomorphic prosthetic hand - across several static grasps and manipulations of common household items. Items and contact maps were borrowed from several existing datasets, including ContactDB [25], Contact Tracing [27], and YCB [42]. The primary desiderata were speed and scalability, robustness, and flexibility to accommodate design or intermediate optimization modifications. Only results for qualitatively reasonable grasps are reported. All simulations were conducted using the Dynamic Animation and Robotics Toolkit (DART) [43].

A. Grasps and Manipulations

Figure 5 provides a collage of static grasps generated using our contact transfer procedure, while Figure 6 illustrates a manipulation sequence used to pull a box of sugar off a kitchen shelf. Note that in a number of cases the manipulator does not possess the required dexterity to successfully reach the contacts; however, the combination of our contact transfer and optimization procedure, along with ability to accommodate interactive user refinement during the process, results in kinematically-feasible "best effort" solutions across a variety of manipulators and objects.

B. Computation Speed

To test the speed and scalability of our algorithm, we examined the time taken by each manipulator to reach a qualitatively reasonable solution across objects containing variable numbers of contact patches. Each trial of four hands was performed with respect to a single object averaged over three separate initial object placements. The same object was used for each patch set (3: scissors, 4: wineglass, 5: box, 6: lemon). All grasps had at most one contact patch per finger and one on the palm. As a result, the three-fingered Barrett hand could not be tested for 5 or 6-contact patches, and the 4-patch Barrett hand trial was conducted using the lemon, which had a palm contact patch available. All hands were initialized to their default start positions. Parenthetical quantities indicate degrees of freedom, and no more than 6 patches were used (1 per human finger + palm). Each hand includes 6 additional degrees of freedom due to a free root joint. We model the Barrett hand's breakaway distal joints as independent degrees of freedom as a simplification due to the fact that the optimization does not model the time sequence of the grasping process. All tests were run on a single Intel Xeon W-1250 3.3 Ghz processor without the use of GPU acceleration or parallel computation.

Grasp Synthesis Speed Comparison - Favorable Initialization				
	3 patches	4 patches	5 patches	6 patches
Human Hand (26)	0.688 s / 0 edits	0.884 s / 0.33 edits	0.422 s / 1 edit	1.036 s / 1 edit
Allegro Hand (22)	0.135 s / 0 edits	0.303 s / 0 edits	0.278 s / 1 edit	N/A
Barrett Hand (13)	0.081 s / 0 edits	0.089 s / 0 edits	N/A	N/A
Prosthetic (22)	0.288 s / 0 edits	0.346 s / 0.33 edits	0.284 s / 0.67 edits	0.394 s / 1 edit

TABLE I: Speed test comparison of 4 different manipulators of varying degrees of freedom with favorable initialization.

Grasp Synthesis Speed Comparison - Poor Initialization				
	3 patches	4 patches	5 patches	6 patches
Human Hand (26)	1.115 s / 0 edits	1.752 s / 0.67 edits	1.406 s / 1.33 edits	2.032 s / 2.33 edits
Allegro Hand (22)	0.397 s / 0 edits	0.721 s / 0.67 edits	0.522 s / 1 edit	N/A
Barrett Hand (13)	0.137 s / 0 edits	0.183 s / 0 edits	N/A	N/A
Prosthetic (22)	0.441 s / 0 edits	1.443 s / 0.67 edits	0.564 s / 1.33 edits	1.267 s / 1.67 edits

TABLE II: Speed test comparison of 4 different manipulators of varying degrees of freedom with poor initialization.

Tables I and II tabulate both the average total times taken across all optimization calls as well as the number of intermediate user edits made to $\boldsymbol{\theta}_P$ during the process. Note that these reported values only consider a small sample size and that variance between different objects and regions of contact is high; however, they do reveal some interesting trends. As expected, greater computation time and interventions are

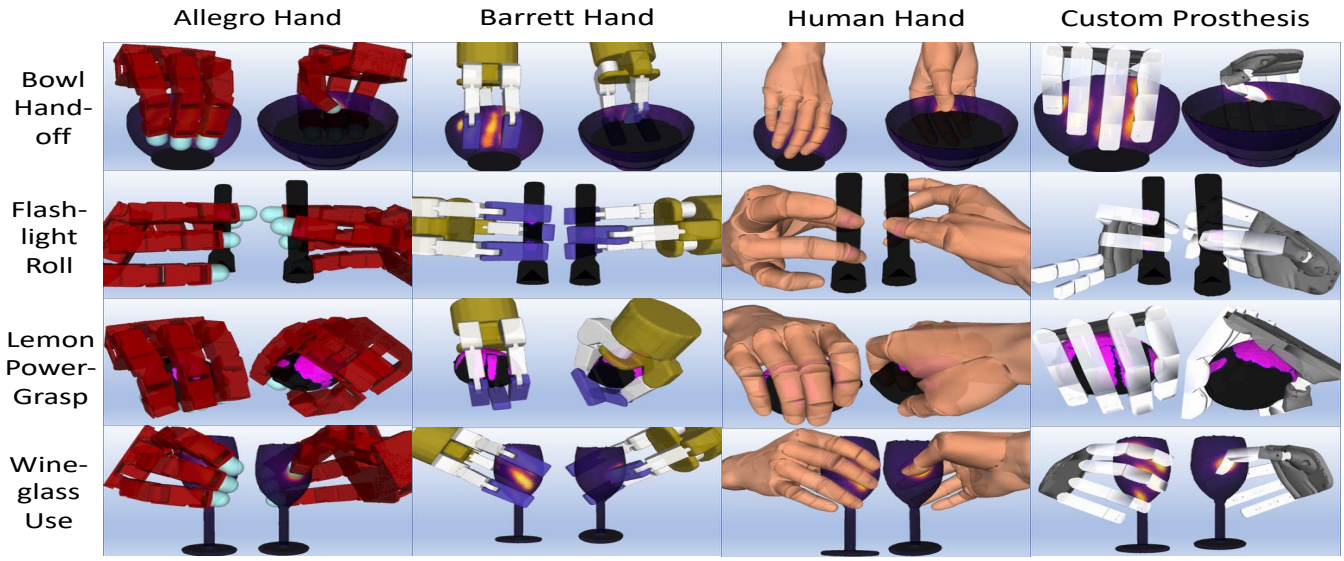


Fig. 5: Grasps synthesized as a result of our transfer procedure across a variety of kinematically diverse manipulators, objects, and grasps.

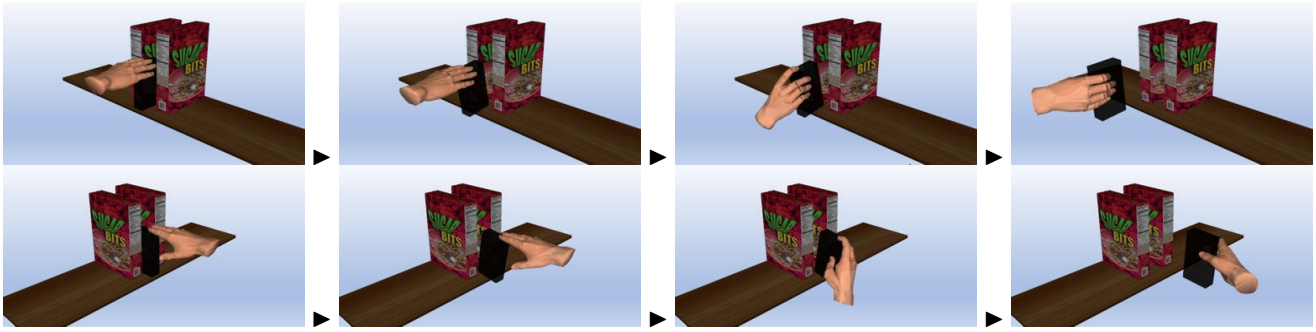


Fig. 6: Demonstration of using the method to quickly approximate a kinematic grasp time series spanning a full simple manipulation of pulling a box of sugar off a kitchen shelf. Poses for successive manipulation steps can be generated at real-time speeds due to the inclusion $\boldsymbol{\theta}_p$ and contact evolution time series data.

(a) (b)

Fig. 7: Results of our method in a configuration where the target object is inverted and placed behind the manipulator. Despite poor initialization (a), our method still finds an acceptable solution after only a few optimization calls (b).

generally required for higher degree of freedom manipulators; however, it is surprising that solution search time did not necessarily increase with more contacts. A possible explanation is that additional contacts may heavily favor certain classes of solutions, and by doing so enable faster local minima convergence; however, additional experiments over larger numbers of objects and contact distributions are required to thoroughly test the conjecture.

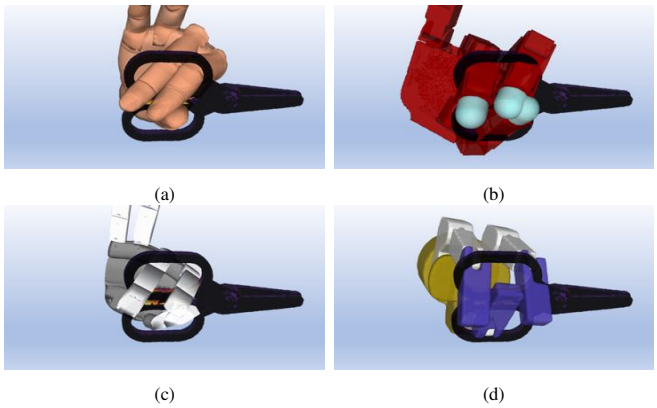
C. Robustness to Poor Initialization

Figure 7 shows the start state and ending grasp of the human hand after running our optimization framework for exactly two optimization calls comprised of 1,000 maximum iterations each, with each round converging and terminating in seconds with no edits. We repeated the procedure for several other poor object and hand initial states and found the behavior to be relatively consistent regardless of the manipulator and object used. As a result, our method is well suited to support dynamic object placement in a plug-and-play manner.

D. Robustness to Difficult Grasps

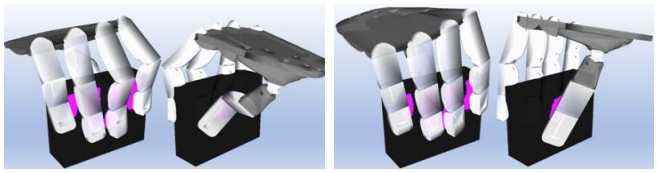
Figure 8 illustrates the outcome of synthesizing a kinematically-feasible grasp in between the holes of a pair of scissors. All solutions were found starting from default rest poses, and no edits were made to $\boldsymbol{\theta}_P$ between calls. While the human hand and custom prosthetic produced reasonable solutions, the Allegro and Barrett Hand noticeably appear too large to find a feasible solution.

Despite the grasp's irregularity and the fact that contacts are in hard-to-reach places, our framework still does a reasonable job in finding solutions quickly and reliably. In the case of the Barrett and Allegro Hand, the framework also provides useful information through fast failure, which can immediately prompt design changes during early stages of



(a) (b) (c) (d)

Fig. 8: Results of our method in finding solutions for a difficult grasp in between the holes of a pair of scissors. The (a) human hand and (c) custom prosthetic found plausible solutions, while the (b) allegro hand and (d) barrett hand failed due to the dimensions being too large for the object.



(a) (b)

Fig. 9: Custom human hand prosthesis demonstration during a box manipulation in which (a) the index finger, middle finger, and thumb joints are allowed to overextend with negative joint limits and (b) when stricter limits are imposed. Note that this alteration changes the whole hand solution rather than only the impacted joints.

prototyping. Finally, due to the irregularity of the grasp and hard-to-reach contact locations, we note that grasp sampling methods would typically fail to find a solution, in contrast to our method which still manages to do so.

E. Fast Reaction to Alterations

Figure 9 illustrates the results of running the same optimization procedure with our custom prosthesis, in which the joint limits are changed on-the-fly. In particular, we note the impact of joint limit adjustments not only on the impacted kinematic chains, but on the entire hand pose. We especially highlight the value of the latter observation within prototyping contexts since it amplifies the impact of even small design changes, which would otherwise be difficult and time-consuming to identify through existing methods.

F. Robot Demonstration

To demonstrate the utility of our approach with respect to prototyping real robot hand designs, we built a custom prosthesis using only a 3d FDM-printer, Thermoplastic Polyurethane (TPU) filament and brushless DC motors. We use a Flashforge Creator Pro 2 [44] retrofit with Flexion Extruders [45] to print the hand in one single part from soft and flexible filaments of varying shore hardness (Ninjatek Cheetah = 95A and Ninjatek Chinchilla = 75A) [46], [47]. The fingers are actuated by tendons (monofilament nylon fishing line) that are routed through channels printed inside the hand. Joint-like kinematics are achieved either by a combination of more rigid and soft materials or by creating local geometric features such as bumps or creases on the hand. For instance, a good approximation of revolute joints is given by a crease coinciding with the joint-axis that runs across the finger or the palm. To accommodate more complex joints with more than one DoF we create rigid-soft-rigid layered features where the softer material acts as cartilage to allow for deformation along multiple axes. We place these features along the kinematic chain and investigate the potential of the hand to successfully grasp or manipulate objects. Discrepancies between simulated and real prosthesis resulting from under-actuation and soft material deformation can be reduced by selecting reasonable joint limits as shown in Figure 9. We show that we can successfully transfer a grasp generated by our method to the real hand in Figure 1 (e). Further grasps are provided in the supplemental video.

V. CONCLUSIONS AND FUTURE WORK

In summary, we have presented a direct, multi-contact transfer framework that can accommodate arbitrary contact regions, objects, and manipulators. We have also presented an optimization procedure that utilizes the transferred results to produce both static grasps and manipulations quickly, reliably, and reactively to user interaction. Our method provides kinematically-feasible solutions without the use of grasp sampling or trained models, and by incorporating user feedback also enables discovery of better solutions without getting stuck in local minima. We have intentionally designed our approach to augment user capabilities, enabling full control over both the transfer and optimization process with responsive adaptation. Finally, we have shown that our method is especially useful for early stage manipulator prototyping, providing the first ground truth data-driven means of testing the impact of design parameters on the kinematics of whole hand grasps in a plug-and-play manner, while at the same time being robust to difficult to reach contacts and dynamic object placement.

However, because this paper only considers hand kinematics, it is possible that our solutions will not work in the real world. As such our next research thrust would be to extend the framework to incorporate physical properties of the hand, object, and environment in an effort to increase the sim2real fidelity of our solutions. Additionally, enabling our algorithm to work directly on point clouds, perhaps similar that of [48], would enable greater applicability to real world environments. We are confident that the foundations laid by this paper will assist in both endeavors, and are excited to showcase the results of both thrusts in the future.

ACKNOWLEDGMENTS

This material is based upon work supported by the AI Research Institutes program supported by NSF and USDA-NIFA under AI Institute for Resilient Agriculture, Award No. 2021-67021-35329, NSF award CMMI-1925130, and by a fellowship from CMU's Center for Machine Learning and Health awarded to Dominik Bauer.

REFERENCES

[1] "Wonik robotics: Allegro hand," https://www.wonikrobotics.com/research-robot-hand, accessed: 2021-09-08.

[2] "Shadow robot company: Dextrous hand series," https://www.shadowrobot.com/dexterous-hand-series/, accessed: 2021-09-08.

[3] M. Diftler, K. C. Jenks, and L. E. P. Williams, "Robonaut: a telepresence-based astronaut assistant," in *Telemanipulator and Telepresence Technologies VIII*, M. R. Stein, Ed., vol. 4570, International Society for Optics and Photonics. SPIE, 2002, pp. 142 – 152. [Online]. Available: https://doi.org/10.1117/12.454740

[4] "Prencilia s.r.l," https://www.prensilia.com/portfolio/mia/, accessed: 2021-09-08.

[5] "Technical university darmstadt: Gripper g1," http://www.thelaraproject.com/G1.html, accessed: 2021-09-08.

[6] D. Bauer, C. Bauer, J. P. King, D. Moro, K.-H. Chang, S. Coros, and N. Pollard, "Design and control of foam hands for dexterous manipulation," *International Journal of Humanoid Robotics*, vol. 17, no. 01, p. 1950033, 2020.

[7] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations," in *Proceedings of Robotics: Science and Systems (RSS)*, 2018.

[8] H. Zhu, A. Gupta, A. Rajeswaran, S. Levine, and V. Kumar, "Dexterous manipulation with deep reinforcement learning: Efficient, general, and low-cost," 05 2019, pp. 3651–3657.

[9] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[10] Z. Jia, M. Lin, Z. Chen, and S. Jian, "Vision-based robot manipulation learning via human demonstrations," *arXiv preprint arXiv:2003.00385*, 2020.

[11] D. Antotsiou, G. Garcia-Hernando, and K. T. K., "Task-oriented hand motion retargeting for dexterous manipulation imitation," *arXiv preprint arXiv:1810.01845*, 2018.

[12] G. Garcia-Hernando, E. Johns, , and T. K. Kim, "Physics-based dexterous manipulations with estimated hand poses and residual reinforcement learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[13] N. Pollard, "Parallel algorithms for synthesis of whole-hand grasps," in *Proceedings of International Conference on Robotics and Automation*, vol. 1, 1997, pp. 373–378 vol.1.

[14] Y. Li, J. L. Fu, and N. S. Pollard, "Data-driven grasp synthesis using shape matching and task-based pruning," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 4, pp. 732–747, 2007.

[15] U. Hillenbrand and M. A. Roa, "Transferring functional grasps through contact warping and local replanning," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 2963–2970.

[16] Y. Ye and C. K. Liu, "Synthesis of detailed hand manipulations using contact sampling," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 1–10, 2012.

[17] S. Brahmbhatt, A. Handa, J. Hays, and D. Fox, "ContactGrasp: Functional Multi-finger Grasp Synthesis from Contact," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[18] P. Grady, C. Tang, C. D. Twigg, M. Vo, S. Brahmbhatt, and C. C. Kemp, "ContactOpt: Optimizing contact to improve grasps," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [Online]. Available: http://www.pgrady.net/contactopt/

[19] H. Jiang, S. Liu, J. Wang, and X. Wang, "Hand-object contact consistency reasoning for human grasps generation," *arXiv preprint arXiv:2104.03304*, 2021.

[20] S. Qiu and M. R. Kermani, "Inverse kinematics of high dimensional robotic arm-hand systems for precision grasping," *Journal of Intelligent & Robotic Systems*, vol. 101, 04 2021.

[21] C. Rosales, R. Suárez, M. Gabiccini, and A. Bicchi, "On the synthesis of feasible and prehensile robotic grasps," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 550–556.

[22] M. A. Roa and R. Suarez, "Computation of independent contact regions for grasping 3-d objects," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 839–850, 2009.

[23] R. Krug, D. Dimitrov, K. Charusta, and B. Iliev, "On the efficient computation of independent contact regions for force closure grasps," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 586–591.

[24] C. Rosales, L. Ros, J. M. Porta, and R. Suárez, "Synthesizing grasp configurations with specified contact regions," *The International Journal of Robotics Research*, vol. 30, no. 4, pp. 431–443, 2011.

[25] S. Brahmbhatt, C. Ham, C. C. Kemp, and J. Hays, "ContactDB: Analyzing and predicting grasp contact via thermal imaging," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6 2019. [Online]. Available: https://contactdb.cc.gatech.edu

[26] S. Brahmbhatt, C. Tang, C. D. Twigg, C. C. Kemp, and J. Hays, "ContactPose: A dataset of grasps with object contact and hand pose," in *The European Conference on Computer Vision (ECCV)*, August 2020. [Online]. Available: https://contactpose.cc.gatech.edu

[27] A. Lakshmipathy, D. Bauer, and N. S. Pollard, "Contact Tracing: A Low Cost Reconstruction Framework for Surface Contact Interpolation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[28] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, p. 509–517, Sep. 1975. [Online]. Available: https://doi.org/10.1145/361002.361007

[29] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org

[30] Q. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.

[31] R. Schmidt, C. Grimm, and B. Wyvill, "Interactive decal compositing with discrete exponential maps," vol. 25, no. 3, p. 605–613, jul 2006. [Online]. Available: https://doi.org/10.1145/1141911.1141930

[32] N. Sharp, Y. Soliman, and K. Crane, "The vector heat method," *ACM Trans. Graph.*, vol. 38, no. 3, 2019.

[33] H. Guggenheimer, *Differential Geometry*, ser. Dover Books on Mathematics. Dover Publications, 2012. [Online]. Available: https://books.google.com/books?id=kavDAgAAQBAJ

[34] N. Sharp *et al.*, "Polyscope," 2019, www.polyscope.run.

[35] N. Sharp, K. Crane *et al.*, "geometry-central," 2019, www.geometry-central.net.

[36] A. Meixner, C. Hazard, and N. Pollard, "Automated design of simple and robust manipulators for dexterous in-hand manipulation tasks using evolutionary strategies," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, 2019, pp. 281–288.

[37] C. Hazard, N. Pollard, and S. Coros, "Automated design of manipulators for in-hand tasks," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, 2018, pp. 1–8.

[38] K. Svanberg, "A class of globally convergent optimization methods based on conservative convex separable approximations," *SIAM Journal on Optimization*, pp. 555–573.

[39] S. Johnson, "The NLopt nonlinear-optimization package," 2017. [Online]. Available: http://ab-initio.mit.edu/nlopt

[40] A. Miller and P. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.

[41] "B. tech. barrett hand," https://advanced.barrett.com/barretthand, accessed: 2021-09-08.

[42] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *2015 International Conference on Advanced Robotics (ICAR)*, 2015, pp. 510–517.

[43] J. Lee, M. X. Grey, S. Ha, T. Kunz, S. Jain, T. Ye, S. S. Srinivasa, M. Stilman, and C. K. Liu, "Dart: Dynamic animation and robotics toolkit," *Journal of Open Source Software*, vol. 3, no. 22, p. 500, 2018. [Online]. Available: https://doi.org/10.21105/joss.00500

[44] "Flashforge creator pro 2 - fdm printer with independent dual extruder (idex) system," https://www.flashforge.com/product-detail/51, accessed: 2021-09-08.

[45] "Diabase engineering - flexion extruder," https://www.diabasemachines.com/flexion-shop, accessed: 2021-09-08.

[46] "Ninjatek cheetah," https://ninjatek.com/shop/cheetah/, accessed: 2021-09-08.

[47] "Ninjatek chinchilla," https://ninjatek.com/shop/chinchilla/, accessed: 2021-09-08.

[48] L. Shao, F. Ferreira, M. Jorda, V. Nambiar, J. Luo, E. Solowjow, J. Ojea, O. Khatib, and J. Bohg, "Unigrasp: Learning a unified model to grasp with multifingered robotic hands," *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–1, 01 2020.