Contact Tracing: A Low Cost Reconstruction Framework for Surface Contact Interpolation

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

***Abstract*— We present a novel, low cost framework for reconstructing surface contact movements during in-hand manipulations. Unlike many existing methods focused on hand pose tracking, ours models the behavior of contact patches, and by doing so is the first to obtain detailed contact tracking estimates for multi-contact manipulations. Our framework is highly accessible, requiring only low cost, readily available paint materials, a single RGBD camera, and a simple, deterministic interpolation algorithm. Despite its simplicity, we demonstrate the framework's effectiveness over the course of several manipulations on three common household items. Finally, we demonstrate the use of a generated contact time series in manipulation learning for a simulated robot hand.**

I. INTRODUCTION

Dexterous manipulation remains one of the grand challenges of robotics. However, achieving dexterous manipulation is difficult in part because manipulation is complex and learning to manipulate is complicated by changing contacts.

There have been a number of impressive manipulation demonstrations utilizing human-like dexterous hands [2], [18], [15], [46] and simpler hands [20], [1], [26], as well as clever applications of extrinsic dexterity [19], [12]. However, robust and general manipulation remains a challenge.

Learning from demonstration (LfD) - i.e. translating a manipulation demonstrated by a human hand or expert operator into an autonomous control policy for a robot hand - has emerged as a popular approach to this problem especially in recent years due to rapid advances in machine learning. However, in-hand dexterous manipulation of the type we do every day is more difficult to teach, in part due to the difficulty of collecting detailed information about the evolving interactions between hand and object.

While observing surface contacts historically has proven difficult, Brahmbhatt et. al. [9] recently presented a technique to directly capture ground truth contact regions via RGBD and thermal imaging. We expand on this approach, presenting an alternative, accessible method to capture surface contacts using thermochromic pigments and a single RGBD camera, as well as a simple, but novel interpolation algorithm to estimate intermediate grasp regions using only 2 scans. Our algorithm is both geometry and contact shape independent, and is applicable to a wide range of manipulation tasks for which the initial and final grasp positions are *distinct* and *direct* (which we call *simple* manipulation). We demonstrate its use in a manipulation performed by a simulated robot hand with a different morphology, and discuss extension to more complex manipulations. Combined, we believe our framework is an important first step in constructing a dataset of contact regions over time - which would be of value for a variety of applications related to LfD, manipulation, hand design, etc. - and that its accessibility would encourage contribution by multiple research groups.

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

II. RELATED WORK

A. Determination of Grasp Regions

There have been numerous prior efforts targeted towards modeling and synthesizing grasp regions of both robot and human hands, the majority of which can be broadly divided into analytical and empirical methods [37], [7]. Analytical methods have considered independent contact regions [33], kinematic and dynamic constraints of the gripper itself [34], and physical models at the points of contact [28], among many other metrics. Data-driven methods in contrast rely on the evaluation of grasp candidates sampled from priors. Techniques have evaluated grasp wrench spaces [8], optimization of contact points [24], [45], and drawing from compiled taxonomies [25], [17], again among many others.

We note however that both approaches ultimately boil down to some form of inference via constrained optimization, and as such can only produce informed approximations of actual contact regions. Our estimates of contact evolution, informed by ground truth contact information captured at key points, can inform and complement such studies, ultimately resulting in better models.

Recently, Brahmbhatt et. al. [9] presented a novel procedure that *directly* captured ground truth human grasps using a combination of RGBD and thermal imaging, culminating in the first large object contact dataset. However, this dataset only consists of *static* grasps. Our use of thermochromic pigment removes the need for thermal imaging by exposing contact signatures to standard RGB cameras, cutting down the financial cost of scanning by an order of magnitude while enabling substantially larger scan resolutions than those of lower cost thermal cameras. In addition, we extend this technique to the estimate of contact trajectories.

B. Learning from Demonstration

Rapid advances in machine learning have driven an explosion of interest in robot LfD. Most LfD approaches can be categorized into one of three categories: kinesthetic teaching, teleoperation, and passive observation [32].

While kinesthetic teaching has proven effective for a number of robotics applications, its viable use in manipulation has proved challenging. Although recent efforts have

attempted to supplement the process [35] and analyze performance across various task sets [27], it is typically not the method selected for manipulation tasks.

Teleoperation [44], [3], [46], [14] has proven effective for large suites of manipulation tasks and provides the advantage of completely circumventing the problem of human to robot motion transfer; however, doing so requires a considerable amount of both technical and financial overhead as well as a trained human expert in the loop. Nevertheless, it remains the go-to technique when general purpose robot dexterity is required.

Passive observation methods - where an observation of human motion is used to drive learning - are a promising option for learning from demonstration, and results have been demonstrated via tracking and retargeting skeletal joints [43], employing the use of priors for object detection [21], and fusion of several trials with various probabilistic models [29]. Most passive observation methods endeavor to track hand pose; unfortunately, occlusion and the inability to place markers on contact regions limits the success of learning transfer, especially if the robot hand is not anthropomorphic. Tactile sensing gloves partially mitigate the contact problem, but the glove itself interferes with the manipulation process, success is largely dependent on the resolution, number, and placement of sensors, and full hand tactile gloves, while in existence [42], are costly[1]. By focusing on contacts, our approach provides complementary information that may be used either in conjunction with existing technologies or alone. We demonstrate one application where contact information alone is sufficient to provide a target for learning a manipulation on a robot hand with morphology different from the human demonstrator.

III. MATERIALS & OVERVIEW

A. Materials & Setup

We selected three objects inspired by the YCB dataset [11] for use in our experiments: a flashlight, a lemon, and a box. The objects were 3D printed using 1.75 mm PLA material on a Dremel 3D40.

The objects were spray painted with acrylic clear gloss mixed with a black-to-pink thermochromic pigment activated at 77°F [2]. Three coats of the mixture were applied using a Preval Spray Kit[3]. A high contrast color profile and a low activation point were chosen to provide optimize signature clarity in the vertex color mapping.

Objects were placed on a 3D printed turntable actuated by a Dynamixel XM430-W210 servo motor[4] and scanned via an Intel RealSense D415. The turntable and object were placed approximately 35 cm away from the camera. The camera was fixed to a tripod and angled to capture the principal surface(s) contacted during the manipulation. Figure 1, subfigure (c) illustrates the approximate setup.

[1]https://pressureprofile.com/body-pressure-mapping/tactile-glove

[2]https://www.amazon.com/Temperature-Activated-Thermochromic-Bi-Color-Changing/dp/B01JYOKF98/

[3]https://www.youtube.com/watch?v=YrdTj47aYhs

[4]https://www.robotis.us/dynamixel-xm430-w210-r/

No cameras were calibrated due to manufacturer provided estimates of both RGB and depth camera intrinsics and extrinsics. We intentionally selected a small turntable diameter to reduce the proximity and note that it was not possible to bring the items closer due to the minimal distance requirement of the RealSense's depth from disparity estimation.

The entire approximate cost of materials required to replicate our experiments, with plenty of materials left over, is $300, of which more than 80% stemmed solely from the cost of the camera and servo motor.

B. Definitions

Our framework uses several representations of contact regions to model traces for which we define the following terms:

Patch (P): An adjacent vertex subset of a given mesh corresponding to a physical region of contact during manipulation. The region is fully enclosed and is thus comprised of an interior (defined as a vertex being a member of P) and exterior (defined as a vertex being a member of P').

Patch Boundary (PB): A vertex subset of a single P which shares at least one edge with an exterior vertex of P and exactly 2 edges with 2 distinct vertices in PB. The set *tightly* encloses all interior points of P - it is impossible to move from any vertex $v \in P \setminus PB$ to another vertex $w \in P'$ without passing through least one vertex $x \in PB$.

Patch Root (PR): An individual vertex of a single P neighbored only by other vertices within P or PB. Any vertex $v \in P \setminus PB$ is a viable root vertex, though in practice vertices closer to the finger tips or palm center were typically selected.

Interpolation Boundary (IB): A vertex subset of PB which encloses all vertices $v \in P \setminus PB$. Unlike PB this set is $loose$ - there is no guarantee of passing through a vertex $v \in IB$ when moving from P to P' or vice versa; however, at any time t of the interpolation, it is possible to generate an approximate $PB(t)$ from $IB(t)$ by performing a shortest path search for each vertex $i \in IB$ to its $cycle\ adjacent$ neighbors.

C. Proposed Algorithm

Our algorithm rests on two core conjectures:

1) Patch movements are synchronous.
2) Assuming a simple manipulation, the growth, contraction, and movement of patch borders on surfaces can be approximated through geodesic tracing.

We note that while none of these assumptions are provably correct, they empirically capture a surprisingly large number of common simple manipulations. The first two conjectures follow naturally from the physical properties and kinematic constraints of the hand itself, while the third can be viewed as a constraint of natural motion in relation to surface geometries.

From these conjectures we propose the following simple interpolation algorithm after obtaining the initial and final signature scans:

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

Fig. 1: High level overview of our reconstruction framework. (a) Prior to manipulation, the objects are coated with thermochromic spray paint. (b) The subject first grasps the object in the initial position, performs the manipulation, and then grasps the object in its final position. (c) The initial and final objects are subsequently scanned and (d) annotated. (e) From these 2 scans our algorithm computes the corresponding interpolations.

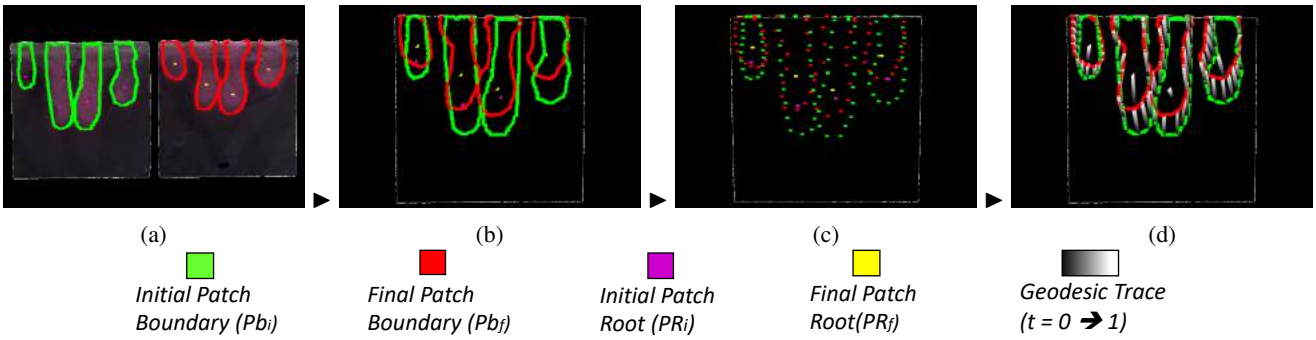


Fig. 2: Post-processing pipeline consisting of (a) annotation, (b) PR and PB computation, (c) downsampled selection of IB, and (d) tracing from initial to final position.

1) Manually annotate each contact patch in both the initial (P_i) and final (P_f) scans (palm, index finger, etc.).
2) Manually select PR and compute PB for P_i and P_f.
3) Compute IB from each PB.
4) Map each IB_i on to IB_f and each PR_i on to PR_f.
5) Trace the geodesic paths for each $i \in IB$ and each PR.
6) Divide the path into N equidistant pieces, where N is the number of frames required to perform the manipulation.
7) Move each PR and $i \in IB$ along the computed paths.
8) When reconstruction of $P(t)$ is required, use $IB(t)$ to generate $PB(t)$ and fill the interior using $PR(t)$ as the source.

Figure 2 illustrates the above steps. We note that the above algorithm reasonably accounts for scenarios in which corresponding patches are visible on both the initial and final scan; however, the existence of patch correspondences on both scans does not always hold. There are several manipulations in which patches are *created* or *deleted* mid manipulation. Consider for example the motions required to pick up and then grip a flashlight, hammer, or cell phone originally lying lengthwise. In the initial grasp only the fingers make contact with the object, but the final grip involves substantial contact with the palm. The new contact is additionally created or removed almost instantaneously, which poses a problem to the aforementioned algorithm.

To account for these manipulations, we extend our algorithm to account for patch creations and deletions by considering such initial (addition) or final (deletion) patches to contain only a PR, but no PB or IB. PR in this context is not an actual point of contact, but only a source (addition) or sink (deletion) from which the actual patch either grows from or collapses to at the time of creation or deletion. The actual time of creation or removal can then be adjusted during annotation depending on the actual time of creation or deletion relative to the start and end of the manipulation.

Additionally, we note that the above steps work only for a *simple* manipulation. However, many full manipulations are more complex, often involving temporary creation or removal of contacts. Both cases involve intermediate information which is not visible at the beginning or end of the manipulation, and would thus not be accounted for by our algorithm. We note, however, that our algorithm can interpolate between any pair of simple manipulations, and thus can naturally be extended to account for complex manipulations through keyframe subdivision. Thus, rather than 2 scans, the same interpolation can be extended to arbitrarily many scans to recreate even highly complex manipulations.

Our resulting algorithm is deterministic, makes no assumptions about surface geometry or patch shape, and is relatively simple to implement. Detailed explanations of each step, including incorporation of the conjectures and implementation, are provided in the proceeding section.

IV. DETAILED PROCEDURE

A. Manipulation and Scanning

Each trial consisted of a full scan of the subject's initial and final grasps. Additional scans were collected if a multi-step complex manipulation was being reconstructed. The ambient temperature of the room was adjusted to approximately 70°F prior to each scan.

Our scanning procedure largely mirrored that of Brahmbhatt et. al. [9]. Prior to manipulation, the subject was presented with a rechargeable hand warmer to increase the strength of the thermal signature. The subject was then asked to perform 3 tasks: to pick up the object while maintaining the initial grip, to perform the full manipulation itself, and to hold the object in the final position.

The object at the end of both the initial and final grasp was scanned using the turntable and RealSense to a ROS bagfile. Three assets were collected per frame: a fixed scale depth image, a color image, and a point cloud. The hand warmers and turntable automation proved crucial due to the transient thermal signature, which faded consistently throughout the scanning procedure due to heat dissipation.

A subset of approximately 10-12 frames sampled at roughly equal turntable increments were selected from each bagfile. Additional frames were selected if the coloration proved unsatisfactory. All videos and frames were recorded at a resolution of 720p.

B. Colored Mesh Generation

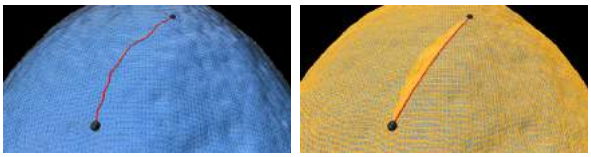
We again largely followed the methods of Brahmbhatt et. al. [9] for generating the contact-textured meshes at both the beginning and end of each manipulation. The final vertex colored mesh was generated using Open3D [48].

The point cloud was first segmented via the following algorithm to isolate the object:

1) Remove the background via z-depth culling.
2) Segment out the base table on which the turntable was placed.
3) Compute the planar convex hull of the base table and its normal.
4) Grow a bounding box from the table in the direction of the normal.
5) Extract all points beyond a specified height within the bounding box.

Post segmentation, the Iterative Closest Point Method (ICP) [6] was used to register the segmented cloud to the original mesh used to print the object. We provided an initial guess constructed from the approximate rotation matrix required to transform the mesh into its scan orientation as well as the angle of the turntable. We additionally culled faces of the mesh not visible to the scanned point cloud during a given turntable angle. The procedure was then repeated for each frame of the scan. All registration and segmentation algorithms were written in C++ using the Point Cloud Library [36].

The mesh was then projected into each snapshot using the ICP-estimated transform, from which the result was provided as the synthetic camera extrinsic per snapshot. The final vertex colored mesh was constructed from each snapshot's RGBD image, the corresponding synthetic camera extrinsic, and the manufacturer calibrated intrinsics via Color Map Optimization [47]. The procedure was repeated at least twice to produce both an initial and final grasp colored mesh. More meshes were produced if the manipulation was subdivided into multiple keyframes.



(a) (b)

Fig. 3: (a) Conversion of a naive path trace over the original triangulated surface to (b) a geodesic path by edge flips over a local intrinsic triangulation [40].

C. Patch Segmentation

Each colored mesh was next annotated to extract each visible P and subsequently labeled with its corresponding hand contact (palm, index finger, etc.).

Hand annotations typically produced errors, including missing points within P and boundary vertices that did not satisfy the definition of PB (i.e. points which had 3 or more neighboring vertices in PB). We therefore constructed a valid PB by first computing all candidate nodes and removed erroneously designated candidates by performing an exterior flood fill over P' [23]. Finally PR was selected for each P.

The procedure was repeated once for each visible P on the mesh to create a fully annotated initial and final state per set of keyframes. All annotation steps were implemented in Python using Open3D.

D. Boundary Point Mapping

To compute IB, each PB was next sampled at equal arc length increments from an annotated start point along an annotated cycle direction to produce an equal number of point-to-point correspondences. 2 neighboring points from each patch were selected to compute the start point and direction. Each corresponding pair of points was then mapped *in order* along the designated cycle. The total number of points selected for each IB depended on the sizes of P. Naturally a larger number of IB points results in greater interpolation accuracy, albeit at greater computational cost.

E. Path Tracing

Paths between initial and final points were constructed as surface-conforming geodesics. We first converted the source mesh into a path-local *intrinsic triangulation* [41], which is notable for fully preserving surface geometry. The path itself was then computed using the edge flip algorithm proposed by Sharp and Crane [40] using the Geometry Central library [5]. Figure 3 illustrates an example of the procedure over one

[5]www.geometry-central.net

of our selected objects. The variable number of points tracing each geodesic was subsequently re-sampled using piece-wise linear interpolation to produce an equal set of points for each path. The number of points selected corresponded to the amount of real time take by the subject to perform the manipulation at a frame rate of 10 frames per second.

F. Boundary Interpolation

We also computed the geodesic trace of the cycle connecting all points within each IB to clearly outline the interpolated PB at each time step while reducing the processing time during final viewing and export. IB endpoints per time step were extracted from a compact $B \times T$ matrix M, where each row corresponded to the vertex index at each point in time while each column represented the vertex index of each IB point per time step. Using M we compute the geodesics between each pair of neighboring points per time step.

G. Patch Interpolation and Reconstruction

With all geodesics pre-computed, we interpolated each P_i to P_f in real time. To reproduce each $P(t)$, we use the corresponding $IB(t)$ to synthesize $PB(t)$ via bi-directional breadth-first search [30] using each cyclic neighboring pair. $P(t)$ can then be reproduced trivially by performing an interior flood fill from $PR(t)$. We exported a collection of flood filled patches as binary vectors for manipulation targets on the simulated robot.

Since geodesics were computed over input path dependent intrinsic triangulations of the mesh, actual mesh vertices may not exist along the locations traced by the geodesic. We therefore computed closest available vertices via K-D tree search [5].

H. Soft Robot Simulation

From the contact patch time series data we extract contact trajectories that directly serve as optimization objectives for learning manipulations. Our simulated environment of a tendon-driven soft robot hand is implemented using the SOFA Simulation Framework [16] and the Soft Robots Plugin [13]. For learning the control policy we embed our simulation environment in the OpenAI gym framework [10] and use a standard Reinforcement Learning (RL) implementation from the stable-baselines framework [31]. Our experimental setup is shown in Figure 6. The hand design follows our previous works on foam robots [38], [22], [4] and features a pair of antagonistic tendons attached to each finger that can flex or extend the fingers through contraction. We consider the standard RL problem formulation in which at every timestep t an agent with state $\mathbf{s}_t$ produces an action $\mathbf{a}_t$ for which it receives a reward r_t and the future state $\mathbf{s}_{t+1}$ in return. We define the state as

$$\mathbf{s}_t = \begin{pmatrix} \mathbf{o}_t & \mathbf{x}_{t,i\in[0,\mathrm{N}]} & \mathbf{d}_{t,j\in[0,2\mathrm{N}]} & \mathbf{g}_{t,i\in[0,\mathrm{N}]} \end{pmatrix}^T$$

where $\mathbf{o}_t$ is the full 6D object pose, $\mathbf{x}_{t,i}$ refers to the fingertip position of the i-th finger and N to the total number of fingers on the hand. $\mathbf{d}_{t,j}$ is the absolute contraction of tendon j (in mm) and $\mathrm{g}_{t,i}$ is the vector pointing from fingertip i to its corresponding contact patch. The action space is defined as

$$\mathbf{a}_t = \left(\zeta_{t,j\in[0,2\mathrm{N}]}\right)^T$$

with $\zeta_{t,j} \in [-1, 1]$ the contraction applied to each tendon j. Instead of using an object pose based reward, we compute the reward r_t based on the location of a fingertip $\mathbf{x}_t$ relative to its corresponding contact patch $\mathbf{c}_t$ on the object according to

$$r_t = -\sum_{i=0}^{\mathrm{N}} ||\mathbf{x}_{t,i} - \mathbf{c}_{t,i}||_2^2$$

with N the number of fingers/patches. We train a control policy using Proximal Policy Optimization (PPO) [39].

V. RESULTS

A. Contact Tracing Manipulations

Figure 4 demonstrates an example of a manipulation in which the subject picks up and subsequently performs a sliding transition between two different grips on a box, while Figure 5 presents a reconstruction of the transition produced by our framework. Despite the complex motion, which involves multiple moving contacts, non-uniform patch shapes at the beginning, middle, and end of the manipulation, transitions over sharp geometric edges, and creation of new contact regions (in this case the pinky and palm) mid manipulation, our simple framework still manages to produce a reasonable reconstruction. We also note that our method is agnostic to manipulation type and surface geometry, and as such generalizes to several common, but distinct initial grasps and in hand manipulations. Additional simple manipulations spanning the box and flashlight, as well as a complex manipulation reconstructed from 3 scans over a lemon, can be viewed in the supplementary video.

In total the time to scan, annotate, and process a trial consisting of two keyframes is approximately 65 minutes, for which the allocations are roughly distributed as follows:

- Scanning: 15 minutes $\times$ 2
- High Quality Annotation: 15 minutes $\times$ 2
- Processing: 5 minutes

We additionally note that while the scanning and processing times are independent of object or manipulation, the annotation time varies depending on the number and spread of the contacts.

B. Robot Demonstration

We additionally exported contact trajectories of the two finger roll and applied it in our learning setup. A time sequence of the learned control policy is shown in Figure 6. The policy learns to initiate contact with the flashlight at the initial contact points and subsequently moves its fingers along the surface to follow the contact trajectory. Note that the rotation of the flashlight is not explicitly encoded in the reward function but rather implicitly given through the movement of the fingers along the object surface. In the given example the robot is not able to touch the flashlight at the designated contact patch with its ringfinger due to differences

Fig. 4: Time sequence of human subject transitioning from a precision grasp to parallel extension on a box.

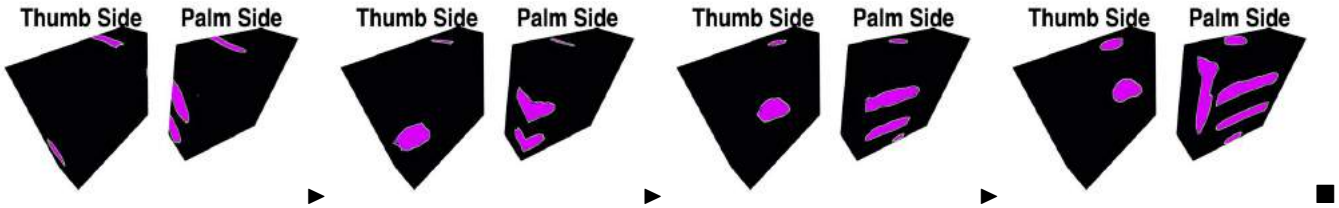


Fig. 5: Time sequence of a transition between a precision grasp and parallel extension via sliding contacts on a box from two different perspectives generated by our framework. The thumb contact (left) is observed to move in natural opposition to the remaining fingers (right), while the index finger (top) acts as the primary pivot. The palm and pinky contacts (frame 4, right mesh, far left and bottom) are generated mid-manipulation.

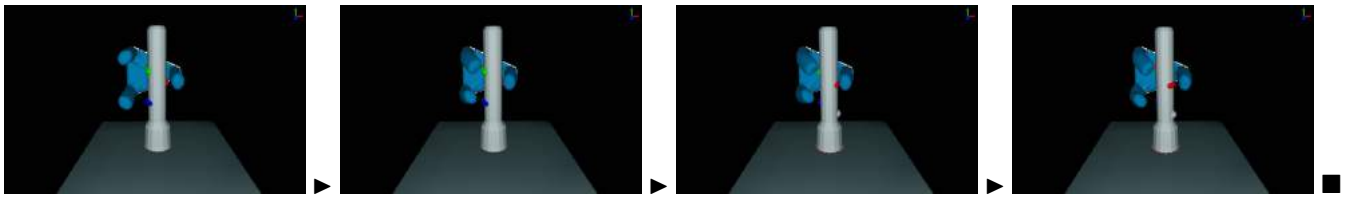


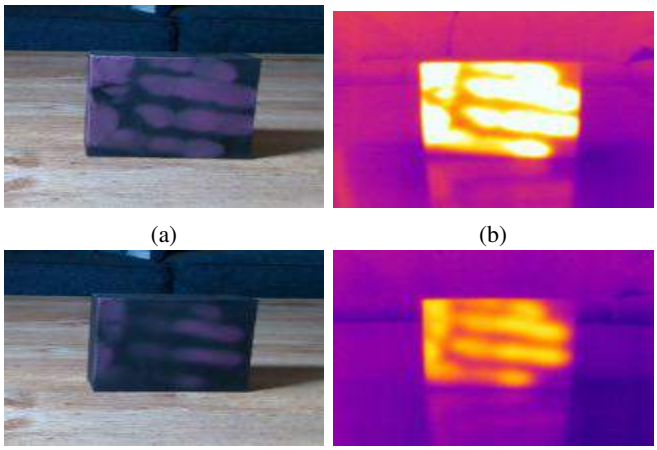
Fig. 6: Time sequence of a learned policy for a two finger roll. Goal contact patch trajectories are visualized by *red* (thumb), *green* (indexfinger) and *blue* (ringfinger) markers.

between human and robot hand morphology and tendon placement. We note that the robot hand is not expected to perfectly replicate the movement of the human subject, but rather use the points as a guideline for its own policy construction. The full demonstration can be viewed in our supplementary video.

VI. DISCUSSION

A. Comparison with Thermal Imaging

We evaluated the effectiveness of our scanning procedure against a scan produced via thermal imaging. Figure 7 shows a side by side comparison of the same patch captured at the exact same points in time as it appeared in the RGB scan produced by the thermochromic pigment as well as the thermal scan captured via a Flir A65. Despite the thermal camera's stronger signature, the distinction between each P and P' is as clearly visible in the RGB image as in the thermal image. We also note that both signatures are time sensitive due to heat dissipation and that the loss in quality was problematic in either imaging process. While using paints and RGB imaging presented some drawbacks including ambient temperature and illumination sensitivity, the large difference in cost renders our technique more accessible for research groups to build upon.



(a) (b)

(c) (d)

Fig. 7: Comparison of our thermochromic RGB imaging procedure with a Flir A65 thermal camera at full signal strength (a, b) and after heat dissipation (c, d).

B. Drawbacks

While relatively straightforward, there are two major drawbacks to our interpolation algorithm. First, we note that geodesic patch movement was a core conjecture of our algorithm. While the conjecture has proven reasonable empirically for simple, natural movements, it is nonetheless

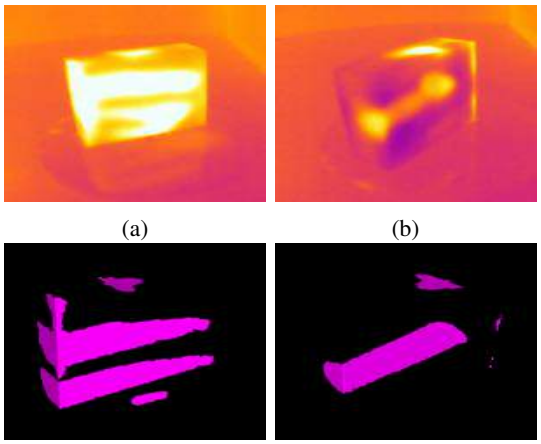


Fig. 8: Comparison of (a, b) a ground truth full trace of a precision grasp to parallel extension manipulation captured via Flir A65 from two different angles against (c, d) our geodesic interpolation reconstruction.

a strong assumption which may impact the accuracy of the reconstruction. Figure 8 illustrates a comparison of our algorithm against a captured full trace. While the reconstruction appears qualitatively reasonable, there exist noticeable divergences with regards to the shape and consistency of the paths. Additional information from tactile gloves or object tracking can directly supplement our current estimates, and, albeit at the cost of more equipment, allow for even more accurate reconstructions as well as quantitative comparisons.

Secondly, we note that annotation currently significantly delays our data generation process. While doing so ensured robustness against noise generated from the scanning procedure, improving the scan quality could easily enable the use of analogous image processing techniques (gradient processing, bilateral filtering, etc.) on the mesh to enable faster extraction of the contact patches.

Finally, we observe that our approach to reconstructing complex manipulations via keyframing, while reasonable in formulation, is difficult to effectively implement in practice and somewhat limited in accuracy within the scope of the current algorithm. While it is reasonably natural for a subject to identify beginning and final grasps, identifying intermediate grasps is unintuitive and sometimes unnatural in the context of our framework. For example, due to the current strong delineation between patch deletion versus movement events, it is not possible to model a contact that gradually disappears during sliding. Instead, the subject would need to artificially subdivide such a sequence into two intermediate grasps - neither of which accurately models the true behavior of the evolution. Additionally, our method alone is impractical to use for manipulations requiring numerous keyframes, such as those involving rapid creation and deletion of contacts. However, were additional tracking information available, keyframe subdivision could by bypassed by instead refining the consistency of the current interpolation state with force sensor readings on a glove via Kalman or Particle filtering.

VII. CONCLUSIONS AND FUTURE WORK

Our primary contributions in this paper are a novel, low cost scanning procedure to record ground truth contact regions on object surfaces, and a straightforward interpolation algorithm to reconstruct intermediate grasp regions of simple manipulations given only an initial and final scan. We have also demonstrated successful transfer of our generated interpolations in the guiding of a soft robot hand to recreate the manipulation, despite the hand having a considerably different morphology than the human hand. Finally, we have presented a means of extending our method to capture more complex manipulations through simple keyframe subdivision. Our work is the first to incorporate data-driven estimations of time evolution into multiple grasp regions on object surfaces during manipulations. Results can be used in manipulation capture, LfD, modeling of manipulation actions, robot hand design, and numerous other applications.

We next intend to use this framework to address two primary research thrusts: creating a full manipulation time series dataset spanning a large variety of objects and manipulations, and leveraging the dataset to not only design more robust control policies for existing robot hands, but also to generate and optimize new designs using contact interpolations as the input. We believe that the existence of such a dataset would enable significant advances in manipulation policies generated via a variety of learning methods, and as such be of interest to the LfD community. During this process we also intend to explore augmenting our current framework with additional tracking of manipulations and using grasp stability metrics to further refine our estimates. We are confident that these modifications will aid in the refinement of our estimates, and by extension the quality and applications of our dataset in a broad range of contexts.

ACKNOWLEDGMENTS

The authors would like to thank Ioannis Gkioulekas[1,2], Anupam Gupta[1], Nicholas Sharp[1], and Cornelia Bauer[2] for their helpful discussions and suggestions over the course of this work. This research was partially supported by the National Science Foundation award CMMI-1925130 and by a fellowship from Carnegie Mellon University's Center for Machine Learning and Health awarded to Dominik Bauer.

REFERENCES

[1] W. C. Agboh and M. R. Dogar. Real-time online re-planning for grasping under clutter and uncertainty. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pages 1–8, 2018.

[2] OpenAI: 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*, 39(1):3–20, 2020.

[3] T. Baker and M. Rolf. Tactile feedback in a tele-operation pick-and-place task improves perceived workload. In *Towards Autonomous Robotic Systems*, pages 261–273, Cham, 2020. Springer International Publishing.

[4] D. Bauer, C. Bauer, J. P King, D. Moro, K. Chang, S. Coros, and N. S. Pollard. Design and control of foam hands for dexterous manipulation. *International Journal of Humanoid Robotics*, 17(01):1950033, 2020.

[5] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, September 1975.

[6] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[7] J. Bohg, A. Morales, T. Asfour, and D. Kragic. Data-driven grasp synthesis—a survey. *IEEE Transactions on Robotics*, 30(2):289–309, 2014.

[8] C. Borst, M. Fischer, and G. Hirzinger. Grasp planning: How to choose a suitable task wrench space. *IEEE International Conference on Robotics and Automation*, 1:319 – 325, 2004.

[9] S. Brahmbhatt, C. Ham, C. C. Kemp, and J. Hays. Contactdb: Analyzing and predicting grasp contact via thermal imaging. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8709–8719, 2019.

[10] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym. *Computing Research Repository*, abs/1606.01540, 2016.

[11] 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)*, pages 510–517, 2015.

[12] N. Chavan-Dafle, A. Rodriguez, R. Paolini, B. Tang, S. Srinivasa, M. Erdmann, M. T. Mason, I. Lundberg, H. Staab, and T. Fuhlbrigge. Extrinsic dexterity: In-hand manipulation with external forces. In *Proceedings of (ICRA) International Conference on Robotics and Automation*, pages 1578 – 1585, May 2014.

[13] E. Coevoet, T. Morales-Bieze, F. Largilliere, Z. Zhang, M. Thieffry, M. Sanz-Lopez, B. Carrez, D. Marchal, O. Goury, J. Dequidt, and C. Duriez. Software toolkit for modeling, simulation and control of soft robots. *Advanced Robotics*, pages 1–26, November 2017.

[14] M. Diftler, K. C. Jenks, and L. E. P. Williams. Robonaut: a telepresence-based astronaut assistant. In Matthew R. Stein, editor, *Telemanipulator and Telepresence Technologies VIII*, volume 4570, pages 142 – 152. International Society for Optics and Photonics, SPIE, 2002.

[15] K. Fang, Y. Zhu, A. Garg, A. Kurenkov, V. Mehta, L. Fei-Fei, and S. Savarese. Learning task-oriented grasping for tool manipulation from simulated self-supervision. *The International Journal of Robotics Research*, page 0278364919872545, 2019.

[16] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, and S. Cotin. SOFA: A Multi-Model Framework for Interactive Physical Simulation. In Yohan Payan, editor, *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, volume 11 of *Studies in Mechanobiology, Tissue Engineering and Biomaterials*, pages 283–321. Springer, June 2012.

[17] T. Feix, J. Romero, H. Schmiedmayer, A. M. Dollar, and D. Kragic. The grasp taxonomy of human grasp types. *IEEE Transactions on Human-Machine Systems*, 46(1):66–77, 2016.

[18] 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)*, pages 9561–9568, 2020.

[19] Y. Hou, Z. Jia, and M. T. Mason. Manipulation with shared grasping. In *Robotics: Science and Systems (RSS '20)*, June 2020.

[20] E. Huang, Z. Jia, and M. T. Mason. Large-scale multi-object rearrangement. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 211–218, 2019.

[21] Z. Jia, M. Lin, Z. Chen, and S. Jian. Vision-based robot manipulation learning via human demonstrations. *ArXiv*, abs/2003.00385, 2020.

[22] J. P King, D. Bauer, C. Schlagenhauf, K. Chang, D. Moro, N. S. Pollard, and S. Coros. Design. fabrication, and evaluation of tendon-driven multi-fingered foam hands. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pages 1–9. IEEE, 2018.

[23] M. Levoy. Area flooding algorithms. *ACM Transactions on Graphics*, 1982.

[24] 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*, 13(4):732–747, 2007.

[25] J. Liu, F. Feng, Y. C. Nakamura, and N. S. Pollard. A taxonomy of everyday grasps in action. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 573–580, 2014.

[26] R. R. Ma and A. M. Dollar. An underactuated hand for efficient finger-gaiting-based dexterous manipulation. In *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, pages 2214–2219, 2014.

[27] J. Morrow, A. Kothari, Y. H. Ong, N. Harlan, R. Balasubramanian, and C. Grimm. Using human studies to analyze capabilities of underactuated and compliant hands in manipulation tasks. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2949–2954, 2018.

[28] R. M. Murray, S. S. Sastry, and L. Zexiang. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., USA, 1st edition, 1994.

[29] E. Pignat, J. Silvério, and S. Calinon. Learning from demonstration using products of experts: Applications to manipulation and task prioritization. *International Journal of Robotics Research (IJRR)*, 2021.

[30] I. Pohl. Heuristic search viewed as path finding in a graph. *Artificial Intelligence*, 1(3):193–204, 1970.

[31] A. Raffin, A. Hill, M. Ernestus, A. Gleave, A. Kanervisto, and N. Dormann. Stable baselines3. https://github.com/DLR-RM/stable-baselines3, 2019.

[32] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard. Recent advances in robot learning from demonstration. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1):297–330, 2020.

[33] M. A. Roa and R. Suarez. Computation of independent contact regions for grasping 3-d objects. *IEEE Transactions on Robotics*, 25(4):839–850, 2009.

[34] 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*, pages 550–556, 2012.

[35] E. Ruffaldi, A. Di Fava, C. Loconsole, A. Frisoli, and C. A. Avizzano. Vibrotactile feedback for aiding robot kinesthetic teaching of manipulation tasks. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 818–823, 2017.

[36] R. B. Rusu and S. Cousins. 3d is here: Point cloud library (pcl). In *2011 IEEE International Conference on Robotics and Automation*, pages 1–4, 2011.

[37] A. Sahbani, S. El-Khoury, and P. Bidaud. An overview of 3d object grasp synthesis algorithms. *Robotics and Autonomous Systems*, 60(3):326–336, 2012. Autonomous Grasping.

[38] C. Schlagenhauf, D. Bauer, K. Chang, J. P King, D. Moro, S. Coros, and N. S. Pollard. Control of tendon-driven soft foam robot hands. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pages 1–7. IEEE, 2018.

[39] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *Computing Research Repository*, abs/1707.06347, 2017.

[40] N. Sharp and K. Crane. You can find geodesic paths in triangle meshes by just flipping edges. *ACM Transactions on Graphics*, 39(6), 2020.

[41] N. Sharp, Y. Soliman, and K. Crane. Navigating intrinsic triangulations. *ACM Transactions on Graphics (TOG)*, 38(4):55, 2019.

[42] S. Sundaram, P. Kellnhofer, Y. Li, J. Zhu, A. Torralba, and W. Matusik. Learning the signatures of the human grasp using a scalable tactile glove. *Nature*, 569(7758):698–702, May 2019.

[43] E. Tosello, S. Michieletto, A. Bisson, E. Pagello, and E. Menegatti. A learning from demonstration framework for manipulation tasks. In *ISR/Robotik 2014; 41st International Symposium on Robotics*, pages 1–7, 2014.

[44] D. Whitney, E. Rosen, E. Phillips, G. Konidaris, and S. Tellex. Comparing robot grasping teleoperation across desktop and virtual reality with ros reality. In *Robotics Research*, pages 335–350, Cham, 2020. Springer International Publishing.

[45] Y. Ye and C. K. Liu. Synthesis of detailed hand manipulations using contact sampling. *ACM Transactions on Graphics*, 31(4):1–10, 2012.

[46] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.

[47] Q. Zhou and V. Koltun. Color map optimization for 3d reconstruction with consumer depth cameras. *ACM Transactions on Graphics*, 33(4), July 2014.

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