Contact Areas for Dexterous Manipulation and Beyond

Arjun Sriram Lakshmipathy

CMU-CS-26-110

May 2026

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Nancy S. Pollard, Chair
Jessica K. Hodgins
Keenan Crane
Zackory Erickson
C. Karen Liu (Stanford University)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Copyright © 2026 **Arjun Sriram Lakshmipathy**

This research was supported by: The National Science Foundation award CMMI-1925130, the AI Research Institutes program supported by NSF and USDA-NIFA under AI Institute for Resilient Agriculture, Award No. 2021-67021-35329, an AI Research and Mentorship award jointly administered through Meta Platforms Inc. and Carnegie Mellon University, and the National Institute Of Biomedical Imaging And Bioengineering of the National Institutes of Health under Award Number R01EB036842. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government, or any other entity.

Keywords: contact modeling, robotics, animation, dexterous manipulation, geometry processing, artist tools, retargeting, optimization, surface parameterization, dense correspondence, model-predictive control, robot bathing, soft hands, tendon-driven actuation

To my past, present, and future family



Abstract

HUMANS use their hands to effortlessly manipulate objects of arbitrarily complex geometries and physical properties every day; however, adapting these behaviors to dexterous robots and virtual characters is an extremely difficult task. Understanding the ways in which humans exploit contact to perform these manipulations has the potential to greatly advance progress towards this goal.

Unsurprisingly, research efforts have analyzed contact in the context of dexterous manipulation for decades. We now have numerous metrics for evaluating the quality of dexterous grasps in terms of contacts, sophisticated models of contact states, efficient means of computing contact in physical simulation, and countless strategies that exploit contact correspondences between hands and objects to synthesize grasps and manipulations. But the majority of existing works fundamentally characterize contact in the same way: as points, lines, or planes of interaction between surfaces.

Although this simplification has proven reasonable for a number of applications, contact in the real world is much more complicated. Instead, real bodies interface with one another via *areas* of contact which greatly vary with the geometries of the contacting surfaces. If we wish to improve our understanding and model the complexities of manipulations as they actually occur, then we must progress beyond such simplifying assumptions and deal with the messy nature of reality. Yet surprisingly, there have been relatively few research efforts in this direction.

This thesis aims to change the current contact modeling narrative by presenting foundational frameworks and algorithms for the modeling, capture, mutation, and exploitation of contact areas. Our intention is to establish the foundations necessary to elevate contact regions to first-class primitives and demonstrate the inherent value they provide across a range of practical applications in both dexterous manipulation and adjacent domains.

First, we introduce three novel models of contact areas alongside a collection of operations supported by each model fundamentally designed to run on real discrete geometries rather than primitive shapes. Next, using area-based contact primitives, we introduce: a set of intuitive artist tools for digitally drafting high quality grasps, a kinematic motion retargeting pipeline for dexterous manipulations, a contact-driven control framework for dexterous robot hands in physical simulation, and two practical extensions of our contributions to different domains. We then shift our focus to the real world by introducing two approaches for capturing and reconstructing contact regions during human-object and human-human interactions. Finally, we present an end-to-end system architecture framework for constructing fully functional robot systems from contact-rich human demonstrations. The contributions in this thesis are not intended to be the last words, but rather important first steps designed to promote future research efforts in contact area modeling and utilization.

Acknowledgments

Doing a PhD is a long, stressful, but rewarding adventure. It is also very, very difficult to get through without a solid support system, and I've got quite a number of people to thank for being a part of mine. However...on the slight chance I miss someone and risk living with permanent guilt after submitting this big ol' document for publication, I'm going to largely address my acknowledgements to not-necessarily-mutually-exclusive groups rather than individuals. So if any individuals in these groups happens to find this document, you probably already know who you are and where you fit in.

First and foremost, I don't think I can start with anyone besides my advisor Nancy Pollard. Beyond our discussions, I greatly appreciated the level of autonomy I had to both explore ideas I was genuinely interested in and set my own schedule. It really gave me the freedom to treat this journey as a marathon rather than a sprint, and also grow both as a researcher and an individual. I also want to acknowledge the incredible support offered especially during some of the lowest points in my career. There was one instance in particular following a failed project collaboration where I really questioned my ability to do research at all. I can't thank you enough for giving me the support I needed to pull myself out of that hole. I'm proud to become an alumnus of the Foam Hands Lab and to carry your influence forward to wherever life takes me.

I would also like to extend a major additional thank you to two other significant mentors in my research journey: Jessica Hodgins and Demetri Terzopoulos. Jessica, thank you for taking me under your wing at Meta and continuing to support the retargeting paper efforts even after you transitioned to RAI. I know your schedule is beyond packed and I really appreciate the time you took to help me out as well as the research and career advice you offered as I neared the end of my PhD. Demetri, without you, I wouldn't even be at CMU to begin with. I've come a long way since my MS days, but I will never forget the chance you took on me way back in 2017 and the doors it has opened for me ever since. More than anything, it helped me find the career path I was ultimately seeking but struggled to find.

Next, I'd like to thank all of my collaborators over the years. It's been incredible working with people from all kinds of different backgrounds (digital animation, geometry processing, mechanical design, AI, computer vision, and all kinds of hybrids!). More than anything, I love hearing about how all of you think about problems — it has had a major impact on research philosophy. Gaining the ability to confidently search across different domains for open problems and ideas is the skill I am perhaps the most proud to have gained from my PhD, and I have all of you to thank for that.

Next comes my thesis committee, many of whom have either been direct collaborators or indirect collaborators via students. You are all some of the most accomplished people I have ever met and your work throughout the years has had a major influence on my research interests. Thank you for taking the time to meet with me, writing recommendation letters, and being flexible with some of the

changes introduced between the defense and proposal.

Moving on, I'd next like to acknowledge my lab / group mates spanning the Foam Hands Lab and the CMU Graphics Group. Thank you all for being such a welcoming group of people, whether that involved research collaborations, general talks about life over lunches, talks at our respective dinner tables, or doing fun activities in and around Pittsburgh. Thank you for giving me a home base and local community that I genuinely looked forward to seeing and being around throughout the years. Beyond these groups, I'd also like to give a shout out to my GHC office mates and the folks at Meta I met during my time there who I hopefully will continue to see moving forward at conferences.

Elmer Street, thank you for being my first and longest standing group of friends both in the department and in Pittsburgh overall. Whether it involved (re-)watching Avatar in the evenings, brunches, exchanges of savory for sweet dishes, spontaneous long distance trips, or ranting about the (perhaps too many) failed dates in the early days before I finally found the one, thank you.

And lastly, but certainly not least, thank you to my current and soon-to-be family. Thanks for always being the rock I needed while navigating the highs and lows of grad school. I don't think it's possible to express the gratitude I feel in words, but do know that it has mattered far more than you can imagine. And as for my soon-to-be wife Sindhu, I can't wait to start our adventure together. We've made it to the end of our respective PhD journeys and I'm confident we can handle whatever life throws at us after dealing with what it already has.



Contents

1 Introduction **1**
- 1.1 Motivation . . . 1
- 1.2 Contributions . . . 2

2 Background & Related Works **4**
- 2.1 Single Point Contact Analysis . . . 4
- 2.2 The Growth of Interest in Contact Regions . . . 5
- 2.3 Dexterous Manipulation . . . 7

3 Preliminaries **9**
- 3.1 Geometry vs. Connectivity . . . 9
- 3.2 Manifolds . . . 9
- 3.3 Tangent Spaces . . . 10
- 3.4 Riemannian Structure . . . 11
- 3.5 Isometries . . . 11
- 3.6 Geodesics . . . 11
- 3.7 Parallel Transport . . . 11
- 3.8 Shape Space Coordinates . . . 11
  - 3.8.1 Barycentric Coordinates on Triangle Faces . . . 12
  - 3.8.2 Linear Coordinates on Edges . . . 12
- 3.9 Exponential and Logarithmic Map . . . 12

4 Representing Contact Areas **13**
- 4.1 Operation Definitions . . . 13
  - 4.1.1 Translation . . . 13
  - 4.1.2 Rotation . . . 14
  - 4.1.3 “Isometric" Deformation . . . 14
  - 4.1.4 Non-Isometric Deformation . . . 15
  - 4.1.5 “Linear" Interpolation . . . 15
  - 4.1.6 Hierarchical Composition . . . 15
  - 4.1.7 Transfer . . . 15
- 4.2 Boundary Model . . . 16
  - 4.2.1 Overview . . . 16

4.2.2 Packing . . . 16
4.2.3 Reconstruction . . . 17
4.2.4 Supported Operations . . . 17
4.2.5 Benefits & Downsides . . . 19
4.3 Single-Point Embedded (SPE) Model . . . 19
4.3.1 Overview . . . 19
4.3.2 Packing . . . 19
4.3.3 Reconstruction . . . 20
4.3.4 Supported Operations . . . 20
4.3.5 Benefits & Downsides . . . 21
4.4 Curve / Axis-Embedded (AE) Model . . . 23
4.4.1 Overview . . . 23
4.4.2 Packing . . . 24
4.4.3 Reconstruction . . . 25
4.4.4 Supported Operations . . . 25
4.4.5 Benefits & Downsides . . . 26
4.5 Summary . . . 26

5 Artist Tools for Grasp Drafting 28
5.1 Motivation . . . 28
5.2 Contributions . . . 29
5.3 Workflow and Assumptions . . . 29
5.3.1 Sourcing Contact Areas . . . 29
5.3.2 Contact Editing . . . 31
5.3.3 Contact Transfer . . . 32
5.3.4 Grasp Computation . . . 32
5.4 Results . . . 33
5.4.1 Static Grasps . . . 33
5.4.2 Difficult Grasps on Higher Genus Topologies . . . 35
5.4.3 Rapid Grasp Transitions . . . 36
5.4.4 Drafting Animations . . . 36
5.4.5 Ablation Studies . . . 40
5.4.6 Comparison with Single Point Contact Model . . . 40
5.5 User Studies . . . 42
5.5.1 Subject Recruitment and Distribution . . . 42
5.5.2 Setup . . . 42
5.5.3 Study Results . . . 44
5.6 Qualitative Assessments . . . 45
5.6.1 Subject Recruitment and Distribution . . . 45
5.6.2 Setup . . . 45
5.6.3 Assessment Results . . . 47
5.7 Drawbacks and Limitations . . . 47
5.8 Summary . . . 49

6 Kinematic Motion Retargeting **50**
6.1 Motivation . . . 50
6.2 Approach . . . 51
6.2.1 Inputs . . . 53
6.2.2 Dense Contact Pairing . . . 53
6.2.3 Hand Shape Matching . . . 54
6.2.4 Motion Retargeting . . . 59
6.3 Experiments and Results . . . 62
6.3.1 Extension: Object Substitution . . . 66
6.3.2 Comparisons . . . 66
6.4 Discussion . . . 69
6.5 Drawbacks and Limitations . . . 71
6.6 Summary . . . 72

7 Online Adaptive Control **73**
7.1 Motivation . . . 73
7.2 Proposed Approach . . . 74
7.2.1 Problem Formulation . . . 75
7.2.2 Online Evaluation . . . 76
7.3 Experiments and Results . . . 76
7.4 Discussion . . . 78
7.5 Summary . . . 79

8 Applications **80**
8.1 Rapid Prototyping of Hand Designs . . . 80
8.2 Scalable Dataset Annotation . . . 81

9 Real World Capture **84**
9.1 Thermochromic Paint Imaging for HOI . . . 84
9.1.1 Motivation . . . 84
9.1.2 Process . . . 85
9.1.3 Textured Mesh Reconstruction . . . 86
9.1.4 Thermal Camera Comparison . . . 86
9.2 Tactile-Augmented Motion Capture for HHI . . . 87
9.2.1 Motivation . . . 87
9.2.2 Capture . . . 88
9.2.3 Shape and Motion Reconstruction . . . 89
9.2.4 Experiments and Results . . . 90
9.2.5 Data Insights and Lessons Learned . . . 95
9.3 Summary . . . 96

10 From Contacts to Systems **97**
10.1 Motivation . . . 97
10.2 Approach . . . 98
10.2.1 Conceptual Design . . . 98
10.2.2 Design and Control of a Tendon-Driven Soft Hand . . . 98
10.2.3 Arm Motion Retargeting and Control . . . 102
10.3 Experiments and Results . . . 104
10.3.1 Soft Hand Design and Control . . . 104
10.3.2 Real World Deployment . . . 105
10.4 Discussion . . . 108
10.4.1 Lessons Learned . . . 108
10.4.2 Limitations . . . 109
10.5 Summary . . . 110

11 Conclusion **111**
11.1 Summary of Contributions . . . 111
11.2 Open Problems and Future Directions . . . 112
11.3 Final Remarks . . . 113

Bibliography **114**

List of Figures

1.1 Sample human hand imprint during a box grasp. Notice the large, irregularly shaped, and geometry-conforming areas of contact generated by the fingers. . . 2

2.1 Early visualizations of human hand contact areas generated during various object grasps. 5
2.2 Various grasp imprints from the ContactDB [Brahmbhatt et al. 2019a] thermal imaging dataset. 6

3.1 Illustrations of (a) manifold and (b) non-manifold geometries. 10

4.1 Illustration of the operations targeted by our contact model representations. . . 14
4.2 Illustration of the boundary model and its associated parameterization entities, including the (yellow) root in relation to (a) the original areas in green and (b) the boundary points in red. 16
4.3 Post-processing pipeline consisting of (a) annotation, boundary, and interior point extraction, (b) overlaid initial (S) and final (T) contact areas on the same mesh, (c) downsampled selection of boundary points, and (d) interpolation from initial to final configuration. 18
4.4 Time sequence of human subject transitioning from a precision grasp to parallel extension on a box. 18
4.5 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. 18
4.6 Illustration of our logmap-based contact patch transfer process, with the single point outlined in black. Consistently parameterized relative distances and angles at all mesh vertices on the source object domain enables transfer of even large, irregularly shaped patches (outlined by white dots) to widely differing hand geometries. 22
4.7 SPE reconstructions are robust to degenerate or completely unconnected domains such as (a) non-manifold meshes or (b) point clouds. 22

4.8 Reconstruction can completely fail if the logmap parameterization fails to be adequately constructed on the target domain. In such instances, the contact area can experience “foldover" by crossing large discontinuities in parameterized space. 23
4.9 Illustration of the AE model packing procedure. 24
4.10 Transfer comparison of a tentacle contact to a cylinder using (a) the axis-embedded contact model and (b) its generated solution against (c) the single-point-embedded contact model from and (d) its generated solution 27

5.1 High level overview of our artist tools workflow. Starting from a given hand and object, the artist typically starts by (a) creating contact areas on the hand and manipulating them into their final positions before (b) transferring them to the object to generate a corresponding set. Artists can then (c) compute the manipulator pose from the contact areas and (d) use the result as feedback to edit the contact areas on the hand and object in an iterative feedback loop. Artists can alternatively (e) first create a (often coarse) pose using traditional posing tools and designate contact areas afterword. (f) Refinement continues until the final result is acceptable. 30
5.2 Optimization progress for finding a feasible box power grasp at (a) 0, (b) 250, (c) 800, and (d) 1,000 iterations. 33
5.3 Poses created using our contact edit framework across kinematically and morphologically diverse manipulators in realistic in-situ contexts. 33
5.4 Additional poses generated by our solver across various manipulators and geometries. 34
5.5 Grasps synthesized as a result of our transfer procedure across a variety of kinematically diverse robotic manipulators, objects, and grasps. 34
5.6 Bulk transfer of a hierarchically organized identical set of contacts on the hand complies to local geometric features on the head (left) and body (right) of the Stanford bunny. 35
5.7 Challenging grasps generated using our method (a) curling the fingers around the cluttered interior of a genus-1 mug handle and (b) between the two holes of a pair of genus-2 scissors. 36
5.8 Our technique can create smooth transitions between multiple baseball pitching grips, including a (a) knuckleball, (b) two-seam fastball, and (c) curveball. 37
5.9 Demonstration of a manipulator adapting its grip in response to variance in object scale. 38
5.10 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 quickly due to the inclusion $\boldsymbol{\theta}_p$ and contact area interpolation. 38
5.11 Result of applying our method to designate contacts on a full body (a) in preparation for sitting in a chair and (b) settled into the final pose. 39
5.12 Catastrophic failures resulting from the removal of optimization terms, including (a) distance, (b) normal, and (c) prior penalties. 39

5.13 Contact sets used as the basis for pose solution comparisons using (a) our axis based area model (b) a curve-only model and (c) a single point model. 40
5.14 Poses computed using (a) Our axis-based area model (b) A curve-only model and (c) A single point model 41
5.15 All user study items 43
5.16 Subject results generated using our plugin in a substantial capacity. 46
5.17 (a) Our transfer method can produce large unwanted distortion if the axis (magenta) is drawn such that the patch parameterization depends on geodesics expected to travel outside of a non-convex domain. (b) However, desired results can still be obtained by simply choosing the axis to lie entirely within the patch region. 48
5.18 Although the geodesic tracing step of our transfer process is not directly applicable to nonmanifold and disconnected surfaces, contact transfer can still be achieved between such surfaces via the use of a manifold offset mesh. Parameterization and transfer occurs between the offset surfaces, and patches are transferred between the original and offset surfaces via projection. Offset surfaces in this figure were constructed using the "shrinkwrap" modifier in the open-source 3D software Blender. 48

6.1 A collection of diverse hand models and some interesting motions synthesized for them. 51
6.2 Overview of our retargeting framework. (a) Our approach requires inputs of accurate meshes of the original object and source hand, per-frame contacts on either the object or source hand, and a complete motion sequence of the source hand. (b) To perform the retarget, we require a skeleton-driven target hand mesh as well as a set of artist-annotated corresponding virtual markers and axial curves. (c) After recovering a dense set of contacts between the object and source hand, we transfer contacts across the entire time series and (d) use the virtual markers and transferred contacts to synthesize motion for the target hand from scratch. 52
6.3 Virtual markers can be configured as traditional single-point one-to-one, or alternatively (a) heterogeneous many-to-one, or (b) dense, area-based configurations. Configuration (a) can be utilized to model uncertainty between virtual marker locations between differing source and target hands, which can be useful when finger link lengths are different sizes. Configuration (b) can be used to weight the importance of matching the deformed hand states over large regions, which can be useful when deformation behaviors diverge despite similar link lengths. . 55
6.4 (a) When examining geodesic distance and outgoing tangent vector direction from a landmark point on the palm, clusters of contact points are clearly separable. (b) Viewing the same distribution from a landmark on the thumb tip cleanly isolates thumb contacts from those of remaining fingers, allowing us to easily introduce discontinuities to filter out the remaining contacts and subsequently perform arbitrary transformations exclusively on the isolated distribution. . . . 56

6.5 Cover generated from a set of landmarks (dark blue), chart boundaries (black), and example corresponding landmarks on each surface (red, orange, teal, light blue) obtained from annotation. A sample contact distribution (cyan) within the boundary of embedded chart U_i is parameterized against the chart's affiliated landmark using logmap transform ψ_i. Inverting ψ_i allows the distribution to be reconstructed from the corresponding landmark on $\mathbb{A}_{\Omega_T}$, while V_i determines the location of the contact's new embedding. 58

6.6 Illustration of isometry-preserving-as-possible contact transfer between two widely varying hand shapes. While unnatural squishing of contacts on the target domain initially fails to model semantic equivalence between fingers, these artifacts can be mitigated by adjusting λ_s and λ_a 60

6.7 All hands used in our experiments, including the (a) source MANO hand, (b) an alternate human hand, (c) a witch hand, (d) an alien hand, (e) a custom prosthetic hand, and (f) the Allegro Hand. (g) We retarget demonstrations performed by the source hand to all these hands by procedurally transferring contact areas over the entire time series via shape matching. 63

6.8 Single point virtual marker configuration on the source MANO hand and area based corresponding marker sets on all other hands. We use a manifold wrapper of the Allegro Hand for contact processing. 63

6.9 One-time axis annotation (top) enables scalable retargeting of original contacts (magenta) to customized configurations per target hand (cyan). Axis colors indicate matching annotations. 64

6.10 (Top Row) Object contacts, (Second Row) source hand contacts, (Third Row) computed target hand contacts, and (Bottom Row) source and retargeted hand motion for four different stages of a complex phone manipulation: (First Column) table pickup, (Second Column) in-hand dialing, (Third Column) holding for use, and (Last Column) movement back towards the table for release. Although poses and contact distributions vary dramatically during the manipulation, our method can successfully produce target hand motion by using source hand contact areas as a foundational retargeting medium. 65

6.11 Plots of all hand-object (green), self (red), and table (blue) intersection. 66

6.12 Our method can be extended to accommodate retargeting generic motions between different objects. Hand grasps successfully make subtle, but important adjustments to adapt to simple shapes (left) as well as more dramatic adjustments for more complex features (right). 67

6.13 Approximate atlases used to retarget contacts of a handoff manipulation between a simple apple and a more geometrically complex gargoyle figurine. Axial curves corresponding to each finger are denoted by different curve colors, while black spheres denote the start point of each curve. Curve placements on the target surface can be arbitrary and be used to implicitly modify grips on the target surface to any extent desired. 68

6.14 (Top row) Functional pose equivalence in terms of keypoints or keyvectors visibly struggle encouraging hand-object contact and motion alignment especially when source and target hand geometries greatly diverge. (Bottom row) Using contact areas greatly reduces such artifacts. 69
6.15 When optimized from the same start position with the same marker and contact distribution targets, (a) a 2-DOF MCP joint converges to a solution which penetrates through the hammer while (b) a 3-DOF MCP joint is able to conform to the hammer's handle with much less intersection. We found that the torsional DOF was utilized in this particular grip across all hands that included a 3-DOF MCP joint. 70

7.1 Overview of our proposed framework. Starting with a single motion capture sequence with corresponding dense contact areas, we retarget both the motion and contact disbributions to obtain a baseline kinematic trajectory. Contact regions are then downsampled to maintain realtime speeds, while the retargeted motion sequence is decomposed into PCs. The motion baseline seeds the controller, while corresponding contact distributions are used to evaluate sampling costs. . 74
7.2 Detailed overview of our sampling-based controller overview. In our case, samples are used to adjust control points of the B-Splines representing PC-curves. Final joint torques τ are computed by re-assembling the PC values back into DOF space and executing the desired joint angles via PD control. 75
7.3 Illustrations of our evaluated physically simulated control tasks. 77
7.4 Plots of time-to-failure metrics across all 4 tasks expressed as a percentage of time between the first and last expected frames of contact per task. Tasks are ordered from left to right by relative complexity in relation to system dynamics, with the doorknob being the simplest and the stapler being the most complex. . 77

8.1 Iterative hand design prototyping framework enabled by contact area transfer and grasp computation 81
8.2 Anthropomorphic hand designs featuring a variety of different thumb placements (*column 1*) are evaluated for their ability to grasp a bowl (*column 2*), a box (*column 3*), a lemon (*column 4*), and a wine glass (*column 5*). 82
8.3 Based on the results in the design study shown in Figure 8.2, candidate (d) is selected for fabrication. All grasps are successfully completed in a pick and place setup using key-framed open-loop poses that match the results obtained in our simulation. 82
8.4 Sample of generated human pose estimation results grounded via corresponding contact regions. 83

9.1 Capture setup used for thermochromic paint imaging along with (upper left) pigment and applicator materials. 85
9.2 Comparison of our (bottom) thermochromic RGB imaging procedure with top a Flir A65 thermal camera at full signal strength (a, b) and after heat dissipation (c, d). 87

9.3 Optical marker configurations used on (a) human subjects and (b) tactile force gloves. (c) Cameras are floor mounted and positioned in a semi-circle configuration. 88
9.4 Comparison film strip of a leg bathing reconstruction using (top) MoSh++ [Mahmood et al. 2019] and (bottom) our reconstruction pipeline. Black arrows illustrate subsequent movement directions. Our method effectively resolves finger contortion and gap artifacts without any significant parameter tuning. 89
9.5 MoSh++ [Mahmood et al. 2019] shape reconstructions of all subjects. 91
9.6 Sample stills of captures generated using our joint tactile and motion reconstruction pipeline. 91
9.7 Contact coverage (yellow) over all captures of a representative (a) male and (b) female subject. 93
9.8 Visualizations of the most commonly (a) contacted taxels and (b) applied pressures across all captures. 93
9.9 Visualization of a bathing “stroke" from the (green) bottom to the (red) top of the back. The subject is stabilized by a contact distribution at the left shoulder during the stroke. 94
9.10 Illustrations of 8 representative contact trajectories, separated into (red) X, (green) Y, and (blue) Z channels. All positions are in world coordinates. Each band is comprised of 65 individual contact point trajectories. Inactive taxels are zeroed out. 94

10.1 Our contributions in this chapter outline how to we can utilize contact areas across the entire development stack to translate human demonstration into fully operational robotic systems. 98
10.2 Overview of our pipeline. Starting with captures of motion and tactile forces from human bathing demonstrations, we digitally reconstruct time-synchronized data across shape, movement, and contact force with high accuracy. We then utilize these reconstructions to both design and control an anthropomorphic soft hand, as well as retarget results to a robot arm. We demonstrate the final system’s efficacy by replicating bathing tasks on a mannequin in the real world. 99
10.3 (a) The robot-assisted bathing setup, consisting of a robotic arm [UFACTORY 2026], DexKit robotic hand [King et al. 2025], and mannequin. (b) The DexKit hand is outfitted with a tactile sensing glove [Pressure Profile Systems 2026] with grip material [3M™ 2026] to increase friction. 100
10.4 Visualizations of improved surface contact following hand pose correction, comparing the original (*top*) and corrected (*bottom*) states. In the first example (*left*), the red arrows highlight a significant increase in finger contact. In the second (*right*), the green arrows indicate the hand flattening against the table as the heel moves closer to the surface. 101
10.5 Illustrations of the (a) mannequin input image and (b) extracted SMPL-X proxy mesh from TRAM [Wang et al. 2024]. 102
10.6 (a) xArm calibration poses collected in the real world are (b) superimposed for mannequin positioning in simulation. 103
10.7 Complete MAE loss curve over the parameter sweep of $|S|$. 104

10.8 The DexKit hand is comprised of a cast foam hand with a textile "skin". Tendons are sewn into the skin (green), and routed along the palm through captive PTFE sheaths (red). The hand shown has six tendons, five flexors on the fingers and thumb, as well as an adductor on the thumb which is routed along the back (shown with dotted lines). The hand interfaces with a motor housing (black) which contains an array of six servo motors. Inside the motor housing the free tendons (orange) route to pulleys (blue) on the motors which reel in the tendons to actuate the hand. 105
10.9 Sample poses used to build $\boldsymbol{M}_{\boldsymbol{\phi}(S')}$. Active contacts in each pose (pink) are simulated by pressing against the hand. 106
10.10 Film strip of back bathing demonstration retargeted to the xArm 7 (top) in simulation using the clinician's shape-fitted hand as the end effector. The computed joint positions and pre-computed hand motor commands are (bottom) deployed on the real world setup. Orange and pink arrows illustrate subsequent movement directions. 106
10.11 Overlaid plots of the (blue) original human, (orange) robot open loop, and (green) robot closed loop tactile glove pressures during a back bathing task. We provide (a) absolute sums, (b) normalized sums, and (c) median intensities calculated across the full taxel array per time step. Note that the spike in pressure at the start of closed loop data collection is due to initialization noise rather than a consequence of trajectory execution. 107

List of Tables

5.1 λ weighting values used to generate results in Figure 5.14. 41
5.2 Qualitative study results 47

9.1 Distribution of all captures in our dataset segmented by body part, assistance level, and pressure intensity. 92
9.2 *L2* Contact distance metric comparison between MoSh++[Mahmood et al. 2019] and our reconstruction. 95

Chapter 1

Introduction

1.1 Motivation



ontact is one of the most fundamental, yet complex physical phenomena that governs interaction with the world around us. Consequently, the ability to estimate, analyze, and plan with contacts has been a research cornerstone of robotics, computer graphics, and computer vision for decades. As humans, we use our hands to perform activities in our everyday lives such as carefully maneuvering dishes out of cluttered spaces in kitchens, wielding a wide variety of tools to perform chores around the house, and interacting with other people, all while giving little thought to the incredibly complex contact dynamics required to perform such tasks. The difficulty in translating such tasks to virtual characters and robots, even under highly simplified settings, is one of the driving reasons behind why animation of such tasks remains extremely time consuming and why autonomous dexterous robots have been difficult to deploy in real world operations rather than carefully controlled lab settings.

To illustrate how dramatically contacts can evolve in a realistic setting, let us consider the mundane task of taking a single glass out a fully loaded dishwasher. First, we must tightly envelop our fingers around the dishwasher handle to create a grip that is strong enough to overcome the friction exerted by the door latch. Next, assuming the glass is facing downward and packed between several other dishes, we must use our fingers to pull out the dish rack and then pinch the sides of the glass that are exposed and carefully pull it out of the rack. Then, we must reorient the glass to face upward and grip it tightly enough so that it won't fall while walking over to its appropriate cabinet. Finally, we must gently release it in the cabinet. To perform this task alone, we had to create a large region of contact between our hand and the dishwasher's door handle that, furthermore, had to conform to the door handle's geometry, utilize firm pinching grips which again had to conform to the geometries of the dish rack and glass, stabilize the glass using our pinching contacts while simultaneously deploying the other fingers to reorient and pull the glass into a firm grip which tightly conformed to the glass structure, and finally carefully release our grip to let go of the glass.

Performing this task thus required us to consistently utilize *large areas* of *geometry-conforming* contacts and, furthermore, drastically change our intermediate grips in an effort to transform less stable, small area contact configurations into more stable, large area configurations. If our goal was to program a robot to replicate this task for us autonomously or, alternatively, create

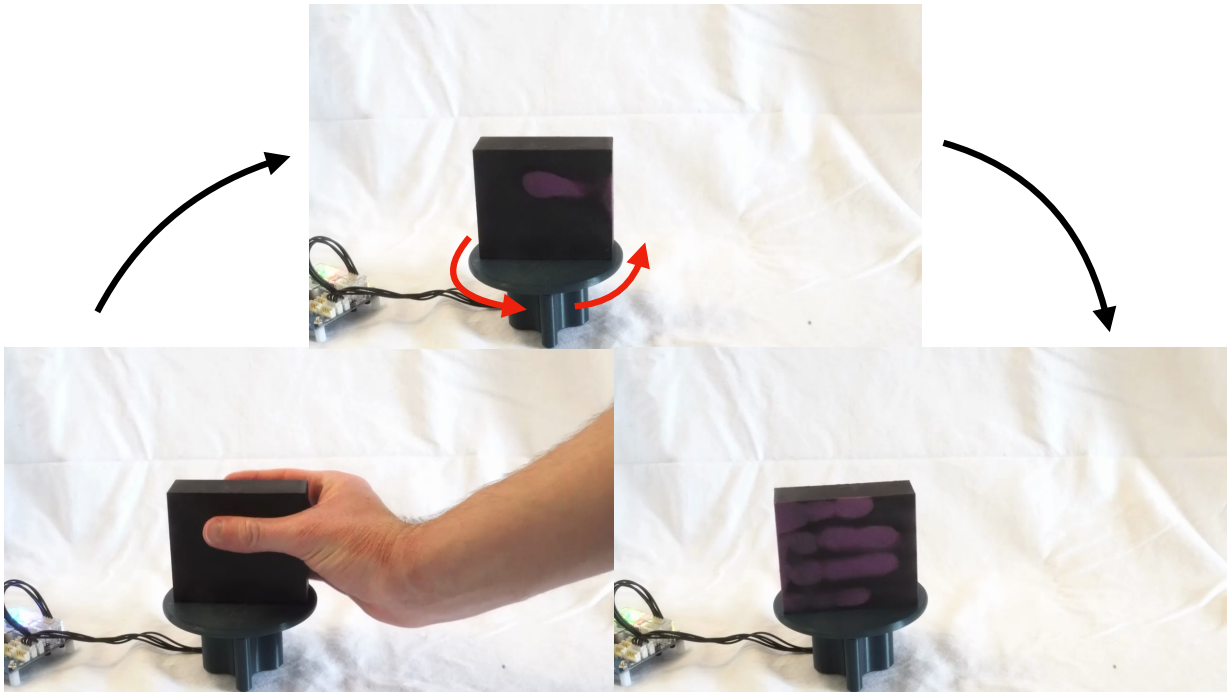

Figure 1.1: Sample human hand imprint during a box grasp. Notice the large, irregularly shaped, and geometry-conforming areas of contact generated by the fingers.

such an animation sequence for a virtual character, instructing the agent to replicate the contact patterns in addition to the hand motions we performed seems like a reasonable starting point.

But surprisingly, despite overwhelming evidence that humans consistently rely on large area contacts to perform everyday tasks (e.g. Figure 1.1), area-based representations have rarely been utilized in frameworks or algorithms for dexterous grasping and manipulation. Instead, many classical and even modern techniques largely focus on analysis of single point contacts. Single point contact models are significantly easier to model and analyze from a computational standpoint, and in the case of multiple complex systems such as full body character interactions [Zhang et al. 2023] and legged locomotion [Bishop et al. 2024], have proven largely sufficient for various applications. Unfortunately, this simplification has proven much less effective for dexterous manipulation — it is simply too far removed from the way contact actually happens in reality. Area-based contact models offer the potential to re-shape the way we characterize manipulation today, and, as we will show, offer multiple benefits over their single point counterparts in multiple verticals.

1.2 Contributions

This thesis aims to establish some of the first foundational models, algorithms, and frameworks designed to elevate contact areas to first-class processing primitives. Specifically, it makes the following contributions:

- Three novel computational models of contact areas, as well as operaters defined on these

models (Chapter 4).

- Novel artist tools for intuitive drafting and control of contact areas, as well as an optimization pipeline capable of synthesizing grasps using the drafted regions (Chapter 5).
- A novel framework for kinematic motion retargeting of complex manipulations between different hands which uses contact areas as the primary retargeting medium (Chapter 6).
- An online adaptive control framework for physically simulated dexterous manipulation tasks involving large areas of contact (Chapter 7).
- Two novel methods for high fidelity contact area capture and reconstruction from the real world (Chapter 9).
- An end-to-end system architecture framework for constructing functioning robot systems from human demonstrations that utilizes contact areas at multiple layers of the development stack (Chapter 10).

We also present several useful applications enabled by our contributions in the context of rapid prototyping and dataset annotation at internet scale (Chapter 8).

The contributions in this thesis are based on the findings in five core publications [Lakshmipathy et al. 2021; 2022; Lakshmipathy et al. 2023; Lakshmipathy & Pollard 2024; Lakshmipathy et al. 2025], one submission under review [Lakshmipathy et al. 2026], and two collaborative works [Bauer et al. 2022; Cseke et al. 2025]. Readers are encouraged to refer to these publications for further details not covered in this document.



Chapter 2

Background & Related Works

2.1 Single Point Contact Analysis

Analysis of single points of contact has been used for decades as the basis for a wide range of applications.

Among the most long-standing uses of individual point analysis in robotic grasping is the search for force-closure grasps [Prattichizzo & Trinkle 2008] – a configuration that, at least theoretically, should be able to resist any external wrench applied to the system via a combination of permissible forces applied at the points of contact. The force closure criteria has lead to a number of follow up works, including the famous maximal wrench space ball metric proposed by Ferrari and Canny [Ferrari & Canny 1992] – the most widely used grasp quality metric even today. There have also been numerous additional quality metrics proposed and re-discovered over the years [Mirtich & Canny 1994; Ding et al. 2001; Kim et al. 2001; Miller & Allen 1999] and even into modernity [Chen et al. 2024b]. A survey of such metrics is available [Rubert et al. 2019], and textbooks detailing these metrics and many others provide a much more comprehensive overview to both grasp quality evaluation and manipulation as a field overall [Mason 2001; Murray et al. 1994; Roa & Suárez 2015]. Additional efforts have used the proposed metrics to search for feasible grasps [Hang et al. 2017; Kim et al. 2013; Li et al. 2007] or maximal independent contact regions [Roa & Suarez 2009].

As locations of interaction between different objects, contact points have also been historically endowed with “modes" that dictate both the types frictional forces applied to the bodies (or lack thereof if friction is neglected) as well as their respective motion manifolds. These modes typically include elastic collision, sticking, sliding, rolling, or torsional twisting. In robot manipulation planning, the assumption of forces being applied at individual points on objects has formed the basis of planners that reason about enumerating and carefully switching modes of contact at the right times to achieve highly non-trivial tasks. These tasks include maneuvering items out of shelves or cluttered spaces [Cheng et al. 2022; 2023; Pang et al. 2023], on-palm manipulation via controlled sliding and rotation [Yang & Posa 2024], enabling simple grippers to achieve object reorientation by exploiting extrinsic dexterity [Hou et al. 2020; Zhou & Held 2022; Zhou et al. 2023]. Even highly complex systems such as quadrupedal robots and even humanoids rely on point contact analysis on the feet as a critical building block for state estimation and gait control.

In computer graphics and vision, points of contact have also been used within the context of

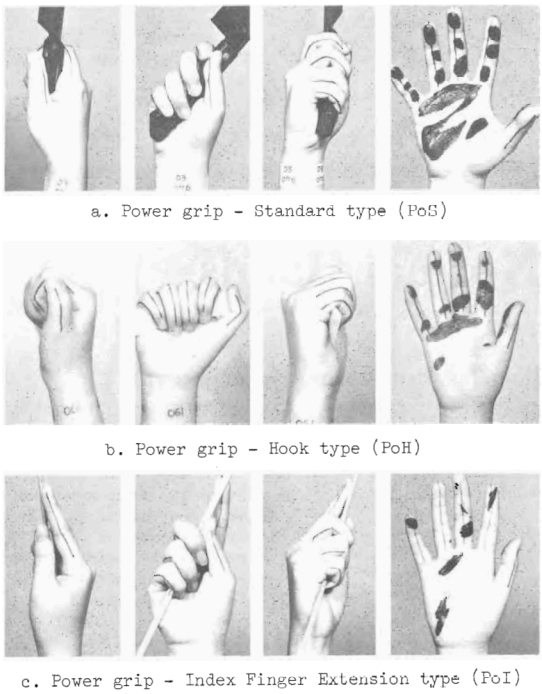

Figure 2.1: Early visualizations of human hand contact areas generated during various object grasps.

learning [Starke et al. 2019; Zhang et al. 2023] and reconciling motion capture reconstructions [Ye & Liu 2012]. Contact points also are useful primitives for identifying correspondences between different bodies, which makes them particularly useful in tasks such as contact planning from images [Yang et al. 2021] or video [Rempe et al. 2020]. Additionally, single point analysis forms a critical building block of solvers for physical simulation — when two bodies collide, solvers must often determine at what point to either apply the necessary repulsive force to push the bodies out of intersection or, alternatively, at which location to forcibly constrain the motion such that the bodies do not intersect in the first place. It is worth noting that accurate contact resolution is a particularly challenging problem in simulation and one of the chief contributors to the simulation to real (Sim2Real) gap in robotics.

2.2 The Growth of Interest in Contact Regions

The idea of considering grasping and manipulation in terms of contact regions rather than individual points has existed for quite some time. Although it has been conjectured for decades, one of the earliest works to provide detailed visualizations of these areas was Kamakura et. al. [Kamakura et al. 1980]. Figure 2.1 depicts some of these visualizations, which were generated by coating objects with a solution of ink and glue. Notably, the shape and distribution of these areas are quite complex.

Since then, there have been numerous efforts to physically simulate contact regions by the computer graphics community for applications in computer animation [Kry & Pai 2003; Jain & Liu 2011; Li et al. 2020], VR / AR [Jacobs & Fröhlich 2011; Romero et al. 2022], and medical simulation [Pai et al. 2018]. Many of these and other works have primarily targeted soft-body

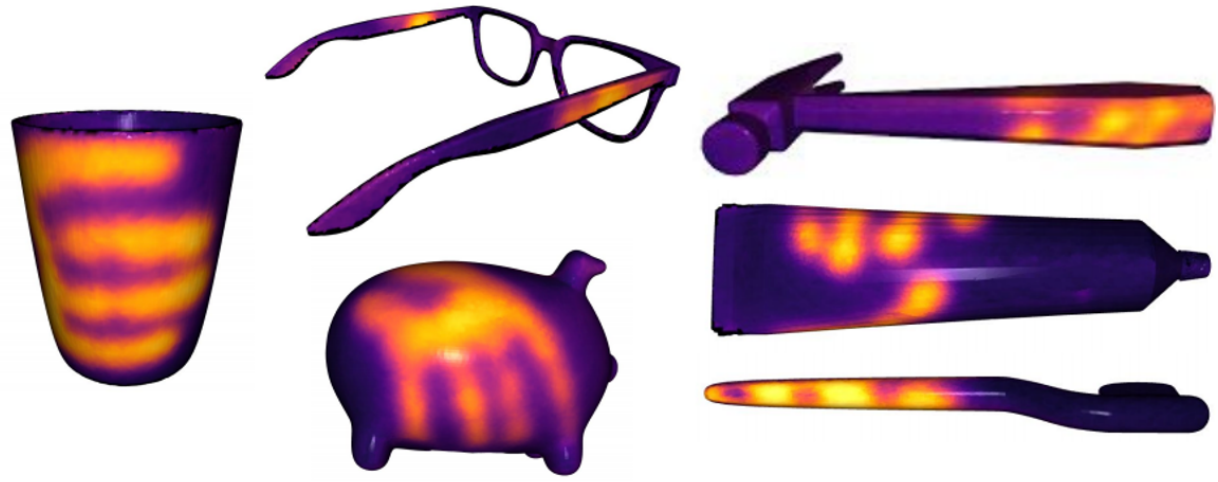

Figure 2.2: Various grasp imprints from the ContactDB [Brahmbhatt et al. 2019a] thermal imaging dataset.

simulation in high fidelity contexts, and as such have primarily resorted to high quality, albeit computationally expensive, approaches such as finite element methods (FEM). In the realm of computational rigid body dynamics, however, one of the most significant improvements to dynamically simulated contact modeling has been the introduction of forces being computed as volumetrically integrated queries over a pre-computed internal pressure field [Elandt et al. 2019]. But a common theme in all of these works is that they primarily target passive contexts, or contexts in which a system is only expected to respond to stimuli.

A more recently realized benefit, particularly in the computer vision community, is the significantly stronger degree of correspondence contact areas provide over their single point counterparts. In our case, we define correspondence as an association defining a mapping between contacting entities. A single point correspondence can thus be concretely defined as a mapping of a single point on the hand (e.g. a finger tip) to an exact single point on the object (strong) or a candidate set of points (weak). Areas by design provide an abundance of correspondences, and can therefor be used to associate an entire region of the hand with a region of the object. The ContctDB [Brahmbhatt et al. 2019a] (examples in Figure 2.2) and ContactPose [Brahmbhatt et al. 2020] datasets, which are comprised of thermally imaged heatmaps of human grasps, are being rapidly adopted for research efforts in grasp synthesis [Jiang et al. 2024], pose refinement [Grady et al. 2021], and contact map generation [Zhang et al. 2024b] particularly for their ground truth annotations of sizeable contact regions on objects. The goal of the aforementioned works and many others is then to generate corresponding contact regions on the hand in order to extract the hand pose which best brings the corresponding regions close to one another in space. Areas are also becoming increasingly popular in the full-body human pose estimation literature as well both due to their ability to assist in pose optimization pipelines and, perhaps more interestingly, for their ability to be easily obtained from ground truth annotations at scale even from non-expert annotators [Tripathi et al. 2023]. The popularity of area-based contacts has considerably increased due to the release and subsequent widespread adoption of

human template meshes, particularly the SMPL [Loper et al. 2023] and MANO [Romero et al. 2017] models. Such standardized templates have allowed research in both vision and graphics to progress beyond skeletal estimations to complete shape reconstructions, which generally benefit from dense contact correspondences. Importantly, these works target *active* contexts. Rather than simply generating responses to externally provided stimuli, these contexts instead involve *exploiting* contact areas to *generate* or *refine* results.

But a notable drawback in all existing area-based works is that correspondences are weak — in other words, given two areas, existing works in the correspondence space only consider a general mapping between discretized point sets obtained by mapping all points in one set on to all points in the other. Additionally, no previous works provide solutions for changing or evolving contact regions with respect to time or user input. This thesis proposes solutions to both problems, and by doing so paves the way for powerful frameworks and applications targeting more active contexts.

2.3 Dexterous Manipulation

As one of the long standing grand challenges in robotics, there has been an extensive history of research efforts in dexterous grasping and manipulation. Building controllers for dexterous hands to manipulate objects is an incredibly difficult problem both due to the high degrees of freedom involved and the natural discontinuities that contact events introduce in system states over time.

Historical works typically considered grasps analytically by examining point contact models with Coulomb friction [Murray et al. 1994; Siciliano & Khatib 2007]. Other techniques include performing global searches for kinematically feasible grasps meeting certain constraints [Pollard 1997], leveraging passive compliance to exploit automatic contact dynamics from physics simulation [Pollard & Zordan 2005], kinematic and dynamic constraints of the gripper itself [Rosales et al. 2012], physical models at the points of contact [Murray et al. 1994], and targeting optimal independent regions within which point contacts can be placed [Roa & Suarez 2009], among many other research efforts.

In the spirit of classical methods that consider explicit models of system dynamics and control, several modern efforts have endeavored to tackle the contact discontinuity problem head-on. These include the use of fully differentiable surrogate functions of discontinuous events [Pang et al. 2023], contact-implicit planning which does not require the explicit determination of modes [Le Cleac'h et al. 2024; Yang & Posa 2024], as well as mechanisms that make the system well-conditioned to solve with long standing techniques in optimal control theory such as differential dynamic programming [Jiang et al. 2024]. Alternatively, other methods, in an effort to alleviate the need for gradients completely, have demonstrated the ability to solve certain dexterous manipulation tasks by using sampling based approaches instead [Howell et al. 2022]. However, it is worth noting that many classical techniques tend to be difficult to deploy on real systems due to the complexities of real world dynamics and, perhaps more importantly, the lack of reliable state estimation techniques for tracking hands and objects together.

More recent approaches to both grasping and manipulation are typically data-driven. Rapid advances in machine learning have driven an explosion of interest in robot learning from expert demonstrations or reward functions. A common source for expert demonstrations of

dexterous grasping and manipulation is from teleoperation, where a human operator generates a sequence of hand poses that gets mapped (retargeted) to corresponding robot hand poses and subsequently deployed to the manipulator in real-time [Handa et al. 2019; Sivakumar et al. 2022; Mannam et al. 2023; Qin et al. 2022; Arunachalam et al. 2023]. The retargeting procedure in such scenarios often must be rudimentary in an effort to maintain real time speed, and thus usually involves simple heuristics such as fingertip keypoint matching [Qin et al. 2022], whole hand keyvector matching [Handa et al. 2019; Sivakumar et al. 2022; Mannam et al. 2023], or direct joint mapping [Arunachalam et al. 2023], although some methods have proposed using data-driven procedures from larger scale human video resources to improve retargeting quality [Sivakumar et al. 2022; Shaw et al. 2022]. The assumption at this stage is that by maintaining real time speeds, even large discrepencies in the hand pose retargeting can be interactively corrected online by the human operator. Then, once a sufficiently large number of demonstrations have been collected, a policy is trained to reproduce the task [Qin et al. 2022; Shaw et al. 2022]. These methods effectively trade off the difficulty of modeling contact dynamics with data-driven priors. But a major drawback of such pipelines in practice, in addition to the massive upfront engineering costs and data collection requirements, is that the type of tasks which can be demonstrated is often limited to pick-and-place or manipulation of environmentally constrained objects.

An alternative strategy for policy construction is to instead use reinforcement learning (RL). Rather than rely on expert demonstrations, this approach instead encourages the robot to search for sequences of actions capable of completing the task on its own using a mathematical framework that balances random exploration and maximization of a prescribed “reward" function. In some cases a reference trajectory is explicitly incorporated as part of the reward function [Rajeswaran et al. 2018; Dasari et al. 2023; Lin et al. 2024b; a], whereas in others simple rewards may suffice [Qi et al. 2022; 2023; Ma et al. 2024]. Contact planning sometimes plays a role in such pipelines, but is often neglected given the exploratory nature of the approach. RL approaches have shown impressive results that cannot feasibly be acquired from teleoperation demos [Chen et al. 2022; Qi et al. 2023; Ma et al. 2024], and are notable for being much more scalable due to less dependence on data collection. However, the incredibly poor sampling efficiency and difficulty of task resetting in the real world mean that RL approaches necessarily require massive computing resources and strong dependence on simulation. These realities mean that the development cycle is slow and inevitably must deal with Sim2Real policy transfer problems. Additionally, choosing good reward functions is arguably more of an art than a science. Finally, the strategy can sometimes result in unexpected learned behaviors that can deviate substantially from expectations.

This thesis proposes methods that build more so on the classical line of work with model-based approaches. However, the use of contact areas in general is agnostic to the choice of solver or controller.



Chapter 3

Preliminaries

Many of the contributions in this thesis borrow techniques from discrete differential geometry, and thus will make repeated use of a number of terms that readers may find unfamiliar. We provide a brief overview of relevant terminology in this chapter, but importantly focus more on informal and intuitive understanding rather than mathematical rigor. A more comprehensive treatment of all concepts discussed this chapter is available in the seminal textbook of Crane [Crane 2018].

3.1 Geometry vs. Connectivity

First, it is important for us to differentiate between the *geometry* and *connectivity* of a shape. A shape's geometry concretely defines how the shape is embedded in, for our purposes, 3D Euclidian space. This includes information such as the position and orientation of its elements with respect to its embedding space. A shape's connectivity, on the other hand, can be viewed as an abstract relationship that simply defines what parts, if any, of the shape are connected to one another and how. Certain operations are capable of impacting one but not the other. Additionally, as we will touch on later, geometry is typically considered and *extrinsic* property whereas connectivity is *intrinsic*. Finally, while every concrete shape is required to have geometry, it is not necessarily required for it to also have connectivity.

To provide some concrete examples, let's consider a shape that is discretized as a triangle mesh. The triangle mesh's vertex positions and normals would be considered extrinsic properties, where as the adjacency list for its edges (or any alternate representation) would define its connectivity. A deformation of a triangle mesh would only alter its geometry, but not its connectivity. In contrast, a "perfect" remeshing – one that perfectly preserves the original structure – can alter the connectivity but not the geometry. In practice that tends to not quite hold since triangles resulting from the remeshing must be planar, which unfortunately also induces a change in geometry. A point cloud, on the other hand, is a representation without any connectivity.

3.2 Manifolds

Manifoldness is a useful property which provides certain guarantees, such as the ability to trace geodesics anywhere on the surface unambiguously. We mainly consider manifolds in terms of surfaces — a 2D structure embedded in 3D space. In this domain, a *smooth* surface would

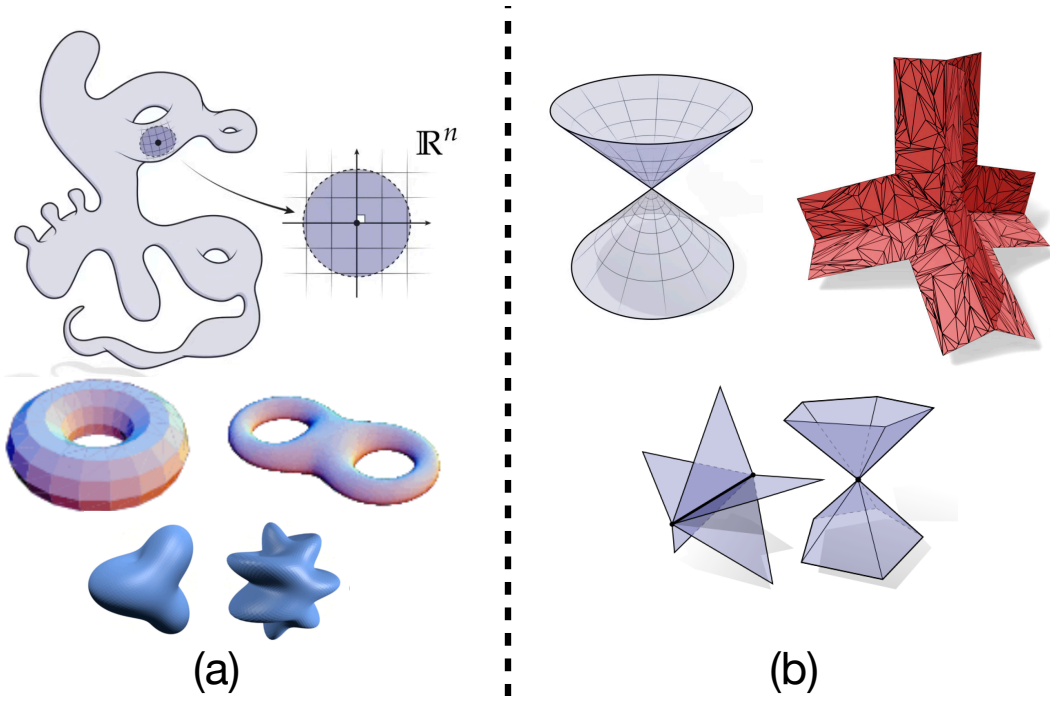

Figure 3.1: Illustrations of (a) manifold and (b) non-manifold geometries.

be considered manifold if every point on the surface can be mapped to a planar disk. Under this informal definition, spheres, tori, and even unusual but locally smooth shapes would be considered manifold. In contrast, as illustrated in Figure 3.1, a pair of cones oriented such that the tips touch one another would be considered non-manifold since no neighborhood around the contacting tips can be mapped to a disk — it therefore has a non-manifold vertex. Similarly, the geometry generated by conjoining more than 2 triangles together along a shared edge (assuming a surface) would also be considered non-manifold, but in this case due to its edges. However, this definition becomes slightly muddled in the discrete setting since even shapes with sharp spikes can still be considered manifold. In this setting, an alternate way of evaluating manifoldness is evaluating whether a vertex appears in only one *boundary loop*. In other words, if we were to start from a vertex, move to one of its neighbors, and then traverse around the vertex visiting each neighbor sharing an edge with it until we run out of eligible neighbors to visit, have we successfully visited every single neighbor the vertex actually has? The double cone and plus-sign shapes would still fail under this definition. This definition also provides an easy algorithm to check if a given shape is manifold. It is also worth noting that certain shapes (e.g. point clouds) cannot be evaluated for manifoldness due to their discretization.

3.3 Tangent Spaces

A tangent space can be conceptually represented by a plane tangent to the shape evaluated at a particular point. Tangent spaces provide a means of linearly approximating the space at that point, and as such can be thought of as the span of a linearly independent basis comprised of a set of tangent vectors defined in the space. For convenience, we will assume that the basis

vectors are orthonormal and that the origin of the tangent space is the same as the point around which it is defined.

A major caveat to note here is that a shape can have many tangent spaces, and that defining a “consistent" tangent space across all points is highly non-trivial. This caveat underscores a major difference between computations in the plane vs. on a surface: a surface has no global coordinate system. This means that any measures on the surface must be defined with respect to the tangent space they are computed on, and that transmitting information between tangent spaces requires computing the equivalent (or rather closest) approximation of that measure with respect to the new tangent space.

3.4 Riemannian Structure

A manifold endowed with Riemannian structure implies the existence of a Riemannian metric g defined on every tangent space of the manifold. The Riemannian metric endows each tangent space with an inner product operation, which for our purposes means that we can perform operations such as measuring lengths and angles for curves defined in the basis of the tangent space. In this work, we assume that every connectivity-endowed discrete geometry we operate on is inherently endowed with a Riemannian structure since we can easily build a tangent basis using outgoing connecting edges from any vertex.

3.5 Isometries

An isometry is a mapping that attempts to preserve the Riemannian metric, which in practice refers to distances and angles between different tangent spaces.

3.6 Geodesics

A geodesic is the natural generalization of a straight line to a curved surface: it is a trajectory of zero acceleration / unit speed, or equivalently, a path of locally minimal length [Sharp et al. 2019].

3.7 Parallel Transport

Intuitively, parallel transport of a vector constitutes moving a vector along a locally shortest geodesic as “straightly" as possible. Our interest in parallel transport is primarily in the discrete setting. In this case, performing a parallel transport essentially boils down to maintaining a “global" tangent vector orientation as it is being transmitted through different tangent spaces along the locally shortest geodesic. A comprehensive overview of how to do so is available in the work of Sharp et. al. [Sharp et al. 2019].

3.8 Shape Space Coordinates

When performing geometric computation, having an exact representation for coordinates anywhere on the shape is highly beneficial. In the case of the shape being a point cloud, each

sampled point uniquely defines a coordinate. Meshes, however, are comprised of vertices, faces, and edges. Locations of vertices, similar to points in point clouds, are definitive; however, additional coordinates are required to identify locations within faces and on edges.

3.8.1 Barycentric Coordinates on Triangle Faces

Barycentric coordinates are a classic coordinate system in computer graphics that allow for interpolating vertex-defined values into a triangular face. Given a triangle ijk with vertex values v_i, v_j, v_k, the three barycentric coordinates c_i, c_j, c_k describe the interpolated value at point p as $v_p = c_i v_i + c_j v_j + c_k v_k$. This coordinates must also satisfy the properties:

1. $c_i, c_j, c_k \geq 0$
2. $b_i + b_j + b_k = 1$.

3.8.2 Linear Coordinates on Edges

In addition to using barycentric coordinates on triangles, we will often use linear coordinates to interpolate values along edges. For convenience, we will also require that the coordinates must sum to 1. Therefore, given an edge ij with vertex values v_i, v_j, we only require a single coordinate t and can determine the interpolated value at point p as $v_p = tv_i + (1-t)v_j$.

3.9 Exponential and Logarithmic Map

The exponential map $\exp_x(u)$ (expmap) of a tangent vector at a point computes the point reached by following a geodesic starting from the point in the direction of the tangent vector for some distance. For compactness, the tangent vector can be decomposed into a unit direction and magnitude to obtain both quantities. The logarithmic map $\log_x(p)$ (logmap) of an arbitrary point when viewed from the origin point of a tangent space gives the smallest tangent vector required to reach the arbitrary point from the origin point under the expmap computation. The maps are therefore inverses of one another and can be thought of as a generalization of a planar polar coordinate system to arbitrary surfaces. A major caveat here, however, is that while for any point p and starting point x, we have that $\exp_x(\log_x(p)) = p$, it is *not* always the case that, for any tangent vector v, $\log_x(\exp_x(v)) = v$. We will illustrate a concrete example of how such breakdowns occur later in some of our experiments.



Chapter 4

Representing Contact Areas

Although we can naively represent contact areas as sets of points either on a surface or in space, it is challenging to perform meaningful operations on such a representation. Instead, we introduce and discuss three *parameterized* geometric representations of contact regions: a boundary model, a single-point-embedded (SPE) model, and a curve-embedded model. We will introduce each representation with an overview of parameters, present packing and reconstruction methods for each representation, discuss operations supported by each method, and will conclude with an analysis of both benefits and drawbacks of each representation. For the sake of consistency, we will assume that all shapes are discretized and that contact regions start out as sets of discrete points in shape space that can be represented either as vertices, face barycentric coordinates, or edge linear coordinates in the case of triangle meshes, or just points in the case of point clouds. Reconstruction therefore requires unpacking from the parameterized representation into discrete points in shape space.

4.1 Operation Definitions

Before we can discuss operations supported by each model, we must first define the operations themselves. We consider 6 operations: translation, rotation, “isometric" deformation, non-isometric deformation, hierarchical composition, and transfer, all of which are illustrated in Figure 4.1. Collectively, these operations enable a wide range of geometric transformations we can reasonably expect contact areas to undergo as they move across a single surface or jointly across multiple surfaces in contact. We define each operation in the proceeding subsections, detail which operations are supported by each model, and how the representations are updated to do so.

4.1.1 Translation

We define patch translation as rigidly moving a patch along a specified path with as little rotation as possible with respect to the path. Mathematically, this definition corresponds to parallel transport (Chapter 3.7) of the patch along the path of translation.

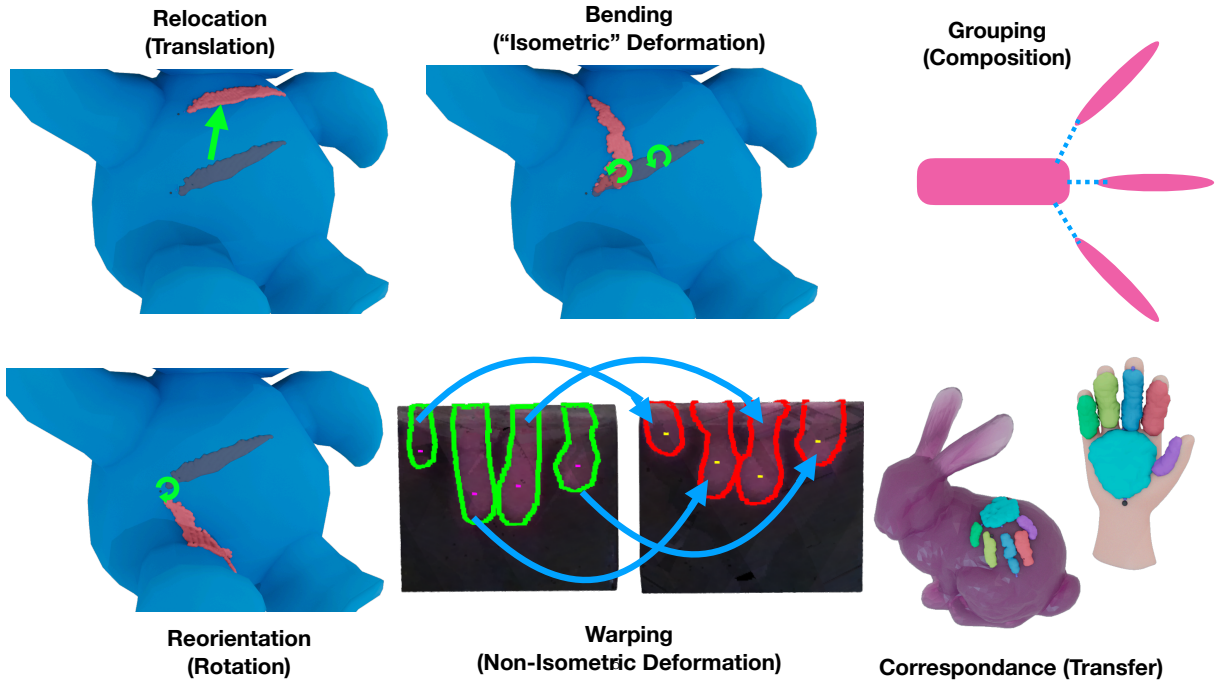

Figure 4.1: Illustration of the operations targeted by our contact model representations.

4.1.2 Rotation

Rotation is the task of reorienting the whole patch "rigidly" around a designated pivot point. If the model also supports translation, then the pivot point can be arbitrarily chosen because a rotation around any pivot point can be decoupled into a rotation around the arbitrarily chosen point followed by a translation. A caveat is that in order for a rotation to be well defined, the model must implicitly be able to assign an orientation to the encoded contact region.

4.1.3 "Isometric" Deformation

"Isometric" – short for "isometric-as-possible" – deformation refers to transformation instances where relative distances are preserved. Put a slightly different way, the canonical shape of the contact region should remain (approximately) the same. As a practical example, consider a snake slithering on the ground. If we were to examine the contact area between the snake's underbelly and the ground, we would notice that it would bend and twist over time; however, the canonical shape of the region would be the same (e.g. if the snake was straight).

In practice, these deformations are not truly isometry-preserving; for example, a geodesic ball in a region with positive curvature will have a smaller area than a geodesic ball of the same radius in a region of zero curvature. The embedding will only be isometry-preserving if and only if the source region and target region on the underlying shape are themselves isometric, which will almost never occur. However, we consider such inaccuracies acceptable for our purposes.

4.1.4 Non-Isometric Deformation

A non-isometric encapsulates all remaining shape changes (e.g uniform and non-uniform scaling, topology changes, etc.).

4.1.5 “Linear" Interpolation

Interpolation is a tertiary operation built on top of the previously discussed operations. At a high level, we define interpolation as the ability to reconstruct an intermediate state of a contact area between a pre-determined initial and final parameterized state. More concretely, given a starting configuration S, final configuration T, and a normalized time step $t \in [0, 1]$, a model capable of supporting interpolation can generate an intermediate state $Q(t)$ such that $Q(0) = S$ and $Q(1) = T$. Although there are numerous ways to perform such an interpolation, in this thesis we only consider “linear" intermediate reconstructions.

4.1.6 Hierarchical Composition

Hierarchical composition concerns the structured grouping of multiple parameterized contact areas together for the sole purpose of performing previously discussed operations on the group *without the introduction of new elements or removal of existing elements from constituent member areas that cannot be easily tracked, added back, or removed.* We impose such constraints because compositions are intended to be temporary relationships – it should be possible to trivially un-group contact regions at will. Removing the constraints would imply that contact regions can simply be permanently merged together before being paramatereized under any model, which defeats the purpose of a composition operation in the first place.

4.1.7 Transfer

Transfer is perhaps the most useful and powerful operation introduced in this thesis. We define transfer as the task of computing an “as-isometric-as-possible" embedding of a contact area on a source shape into another target shape. In the context of this thesis, the most practical and compelling use case would be transferring a contact area parameterized on an object geometry to the hand performing the manipulation, or vice versa.

Different shapes can have extremely different shape space coordinates. For example, the object may be encoded as a dense triangle mesh obtained from a high resolution 3D scan, while another object might be represented as a point cloud collected from a depth sensor, and a hand might be a coarsely triangulated mesh created by an artist. Additionally, the actual encoded shape of all of these discretizations can be drastically different from one another. In order for transfer to be possible between such geometric variations, we must have a way of computing over such geometries using a consistent representation.



Figure 4.2: Illustration of the boundary model and its associated parameterization entities, including the (yellow) root in relation to (a) the original areas in green and (b) the boundary points in red.

4.2 Boundary Model

4.2.1 Overview

As the name suggests, the boundary model represents contact areas using a collection of points on the boundary of the region. However, a simple unordered set of boundary points provides insufficient information to support accurate reconstruction. We also need to know:

1. How is the boundary *oriented*? If I am at a particular boundary point, how do I know what is the *next* boundary point in the loop?
2. What regions are *inside* and *outside* the boundary?

Fortunately, these ambiguities are easy to resolve. Our final parameterized representation thus consists of:

1. An *ordered* set of boundary points
2. A single extra *interior* point, sometimes called the “root"

Figure 4.2 illustrates the proposed model and its associated parameters.

4.2.2 Packing

We start by clamping all face barycentric coordinates and edge linear coordinates to vertices. Extracting an unordered set of boundary points for a contact region is fairly simple on a triangle mesh if we consider only vertices – we can simply select all vertices that have at least one neighbor that is not a part of the original contact area group. In practice, however, this does not work if the original contact area contains holes — a common artifact from imprecise annotation of the original area. We can resolve holes by performing a flood fill starting from some vertex exterior to the contact region and removing any boundary candidates not visited by the fill.

By construction, a candidate boundary point's "next" point can only be any of its immediate neighbors on the triangle mesh that are also boundary candidate points. We can therefore select any boundary candidate point and pick any of its immediate neighbors as that starting direction of the loop. We can then simply repeat the procedure until a full loop has been extracted. In the event of multiple candidate neighbors being available, we can perform a branched search and discard the path which terminates early (which in practice is viable because the search trees are of limited depth). After a boundary loop has been extracted, we can also arbitrarily *downsample* the boundary by keeping only every kth point.

Finally, we can select an interior point from any of the non-boundary points by any means, be it randomly, manually, or through some heuristic (e.g. largest average geodesic distance from all boundary points).

4.2.3 Reconstruction

To reconstruct a full contact area, we simply perform a flood fill starting from the stored interior point with termination conditions on the boundary points. Note that because the boundary may not be "watertight" (e.g. if it was downsampled), we must perform a preliminary step to convert the currently stored "sparse" boundary points into a dense, watertight boundary. We can do so by simply computing a shortest path from each stored boundary to the next using Dijkstra's algorithm [Dijkstra 1959].

4.2.4 Supported Operations

Translation We can compute the direction and distance of translation in the tangent space of the interior point, parallel transport the computed tangent vector to the all the boundary points, and then execute the operation by updating the boundary and interior point locations to the endpoints of the resulting traced geodesics.

Non-Isometric Deformation Any update to the boundary points that is not a translation is considered a non-isometric deformation. Such transformations enable the boundary model to vary its shape arbitrarily, which allows fitting to virtually any contact distribution — including those captured from the real world (see Chapter 9). Additionally, by endowing the model with an indicator function, we can also model events such as sudden making and breaking of contacts.

"Linear" Interpolation A linear interpolation can be performed under either a translation or non-isometric deformation operation by interpolating the traced geodesic path of each boundary point. Interpolation of a non-isometric deformation is a particularly interesting case due to its ability to capture the complexities of real world contacts. We found that it was possible to obtain surprisingly high quality intermediate reconstructions even using linear path assumptions.

As illustrated in Figure 4.3, if we assume an equal number of boundary points in S and T and the correspondences between the boundary point sets are known, then non-isometric deformation can be performed under the same principle as translation interpolation. Although such a simplification may be reasonable for a relatively simple interpolation (e.g. Figure 4.3),

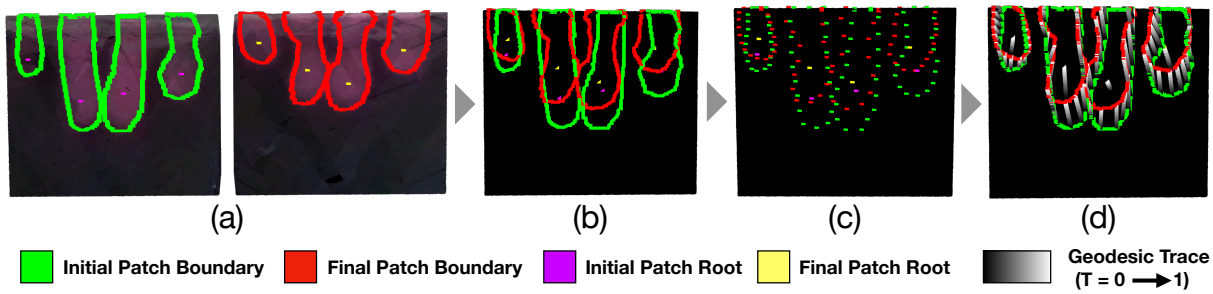

Figure 4.3: Post-processing pipeline consisting of (a) annotation, boundary, and interior point extraction, (b) overlaid initial (S) and final (T) contact areas on the same mesh, (c) downsampled selection of boundary points, and (d) interpolation from initial to final configuration.

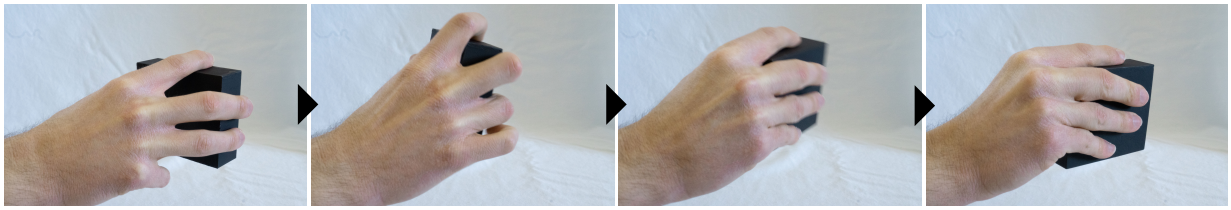

Figure 4.4: Time sequence of human subject transitioning from a precision grasp to parallel extension on a box.

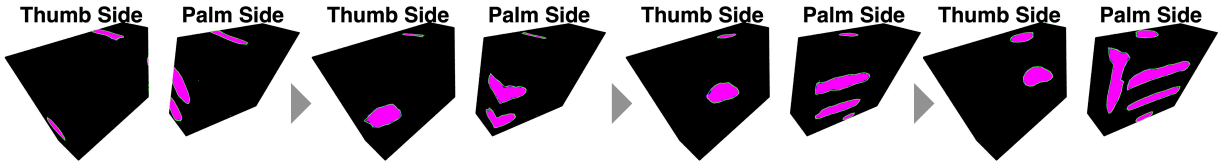

Figure 4.5: Time sequence of a transition between a precision grasp and parallel extension via sliding contacts on a box from two different perspectives generated by our framework. The thumb contact (left) is observed to move in natural opposition to the remaining fingers (right), while the index finger (top) acts as the primary pivot. The palm and pinky contacts (frame 4, right mesh, far left and bottom) are generated mid-manipulation.

Figure 4.5, which endeavors to reconstruct the manipulation shown in Figure 4.4, illustrates a more complex scenario in which the contact shapes of the middle and ring finger vary dramatically through the manipulation sequence.

Figure 4.5 also illustrates a scenario in which contacts — in this case induced by the palm and pinky finger — are generated mid manipulation. The contact “making" operation can be modeled under the translation principle by instead having the geodesics of the boundary points originate from the interior point and terminate at the final corresponding locations. Then, using a time-indexed indicator function, the contact can be generated immediately at an arbitrary time step mid-manipulation. The same operation can be run in reverse for a breaking contact.

4.2.5 Benefits & Downsides

By construction, a boundary-based representation ensures that no holes will be generated in the interior during reconstruction, even if the original region contains such artifacts. Additionally, the representation can accommodate complex contact region shapes as well as a means of deforming the shape in non-trivial ways via updates to the boundary point locations.

However, this representation is completely dependent on the underlying shape being endowed with connectivity – it would be unusable on unconnected representations such as pointclouds. It is also sensitive to shape discretization due to the vertex clamping pre-processing step, which means that, practically, reconstruction fidelity would degrade considerably on coarsely triangulated meshes. Both the embedding and reconstruction processes are also quite slow, which renders the model impractical for real time applications. Finally, this model supports only a small subset of the operations we are interested in, particularly because it does not assign an orientation to the contact area or a method of reconstruction on any surface other than the one it was generated on.

4.3 Single-Point Embedded (SPE) Model

4.3.1 Overview

The single-point embedded (SPE) model takes advantage of the logmap parameterization to represent all points that comprise a contact region. To obtain logmap coordinates, we first parameterize the underlying shape. More specifically, we must impose a logmap chart at some designated single origin point on the surface, which consequently is how the model’s name is derived. This model thus stores:

1. A single point represented in shape space coordinates that serves as the the logmap origin
2. A tangent vector direction from the single point that determines the reference angle
3. Logmap parameterizations of all contact points

4.3.2 Packing

For the packing process to start, the single point and tangent vector direction must first be selected. The logmap parameterization of the shape is then computed using the Vector Heat

Method (VHM) [Sharp et al. 2019]. All contact area points, originally represented in shape coordinates, are converted to logmap coordinates under the imposed shape parameterization.

4.3.3 Reconstruction

To begin reconstruction, the underlying shape is first converted to its logamp parameterization using the provided single point location and direction. A search is then performed over all discrete elements in the underlying shape, and the element with the closest parameterized values to each parameterized contact point is selected. The search for the best matching point for contact point c_i can be more compactly stated as:

$$\begin{aligned} c_i^* = \arg\min_{y} \quad & \| log_x(y) - log_x(c_i) \|_2^2 \\ \text{s.t.} \quad & log_x(\cdot) \quad = \quad f(x; \vec{x}) \end{aligned} \tag{4.1}$$

where x is the single point, $\vec{x}$ is the reference tangent vector direction, and y is a candidate element. The contact point is then converted back to shape coordinates by adopting the shape coordinates of the optimal candidate element. We elect vertices as candidate elements for meshes and all samples for point clouds.

4.3.4 Supported Operations

Translation Translation is trivial because, unlike the boundary model, only the single point must be updated — the representation of the parameterized contact points does not change whatsoever, and the patch can be fully reconstructed from the newly updated location of the translated single point. This operation can also be performed on point clouds as long as a suitable Laplacian is available [Sharp & Crane 2020].

However, we must be careful to account for the implicit change of basis that occurs between the starting and ending location of the translation (see Chapter 3.3). Ignoring this step can otherwise induce an unwanted change in “global" orientation at the destination. Therefore, we must also parallel transport the starting tangent vector direction to the destination point and update it to the local tangent basis at the destination.

Rotation The SPE model automatically assigns an orientation to the contact area via logmap parameterization. Therefore, we can change the orientation simply by designating the single point as the arbitrary pivot for convenience and altering the outgoing tangent vector direction at said point. Because this direction change only impacts the reference angle, no changes need to be made to the parameterized points — they can be reconstructed as-is following the new reference angle. Rotations about any pivot point can also be generated by combining rotation with translation.

“Linear" Interpolation The SPE model supports interpolations of both translations and rotations, albeit by different means. We will discuss each briefly.

Translation. Translations only involve movement of the single embedding point; therefore, we can simply compute a geodesic between the point's initial and final location and interpolate along the geodesic. We must also parallel transport the starting tangent vector and update its representation to be in the basis of $Q(t)$. Because we are forced to compute the path geodesic, however, translation interpolation can only be performed on domains with connectivity.

Rotation. The change in outgoing tangent vector between the initial and final configurations can simply be interpolated with respect to the turning angle to produce intermediate states.

Hierarchical Composition Contact regions can be temporarily grouped simply by converting the single point embedding source of the "child" parameterized contact region into a temporary pseudo-contact point parameterized by the "parent" contact region. Any operations performed on the parent contact can simply be applied to the child pseudo-contact point, and the child can be reconstructed using its existing stored parameterization. A caveat is that, in the event of a rotation, the parent must also parallel transport its modified outgoing tangent vector direction to the child source point in order for the child to update its reference angle. The relationship can be uncoupled by removing the removing the pseudo-contact relationship.

Transfer The logmap provides a unified representation that can be used to move SPE contacts between surfaces. As long as the elements of both shapes are endowed with logmap-parameterized coordinates, we can embed a contact region on the source shape and transfer it to the target shape without modifying the parameterized values of the contact points. Furthermore, because the reconstruction strategy is search based, we can even embed on a mesh and reconstruct on a point cloud or any shape representation for which the VHM's required Laplacian can be computed (or vice versa). Figure 4.6 illustrates a sample transfer operation, where the contact is first parameterized on the object and then reconstructed on two different hand shapes. The overlaid logmap illustrates the consistent parameterization across the domains, while the black squares indicate the respective single embedding points in each domain.

4.3.5 Benefits & Downsides

This method guarantees that there will be an equal number of points comprising the contact area during reconstruction — if a contact area is represented with 30 points, there will always be 30 points during any reconstruction call. An even more compelling benefit is that this method does not rely on connectivity and thus does not necessarily need a manifold mesh as input — as illustrated in Figure 4.7, it is capable of operating on degenerate non-manifold meshes, point clouds, or really any shape representation for which a Laplacian can be computed. The representation can also support "rigid" transformations through updates to the single point location or reference tangent vector direction. Finally, it is a significantly more reliable model in terms of parameterization compared to the boundary representation because the caveats of the contact area (e.g. holes) are not important.

But there are multiple downsides. First, similar to the boundary method, reconstruction fidelity still depends on the resolution of the underlying surface due to candidate selection.

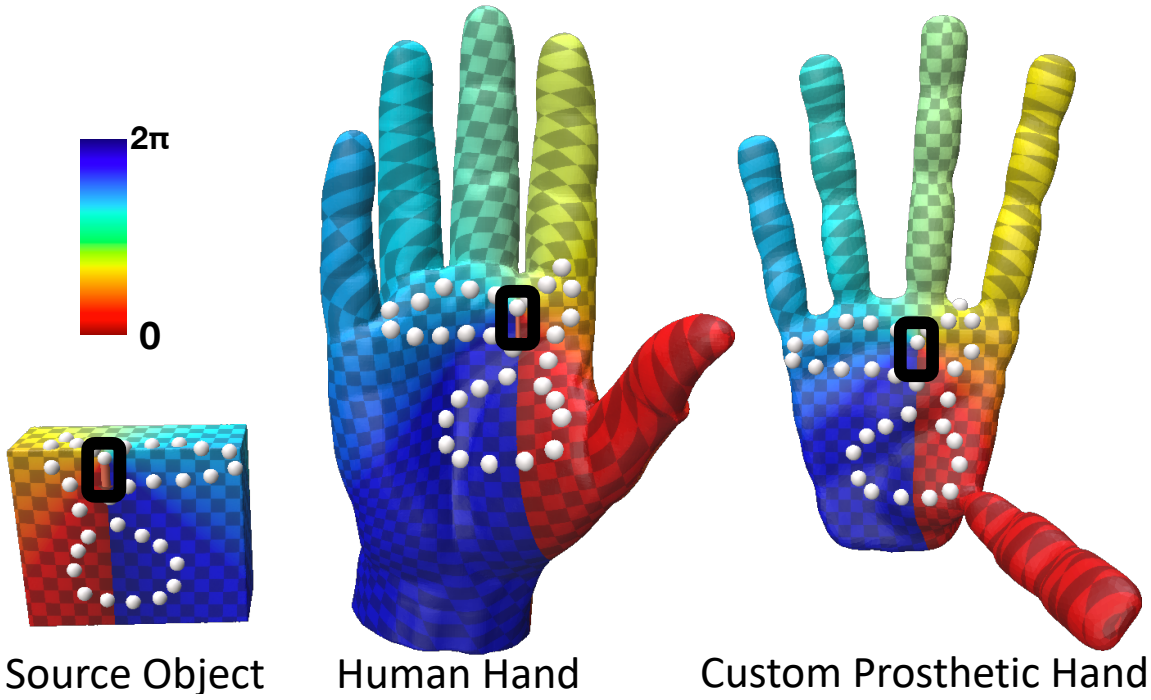

Figure 4.6: Illustration of our logmap-based contact patch transfer process, with the single point outlined in black. Consistently parameterized relative distances and angles at all mesh vertices on the source object domain enables transfer of even large, irregularly shaped patches (outlined by white dots) to widely differing hand geometries.

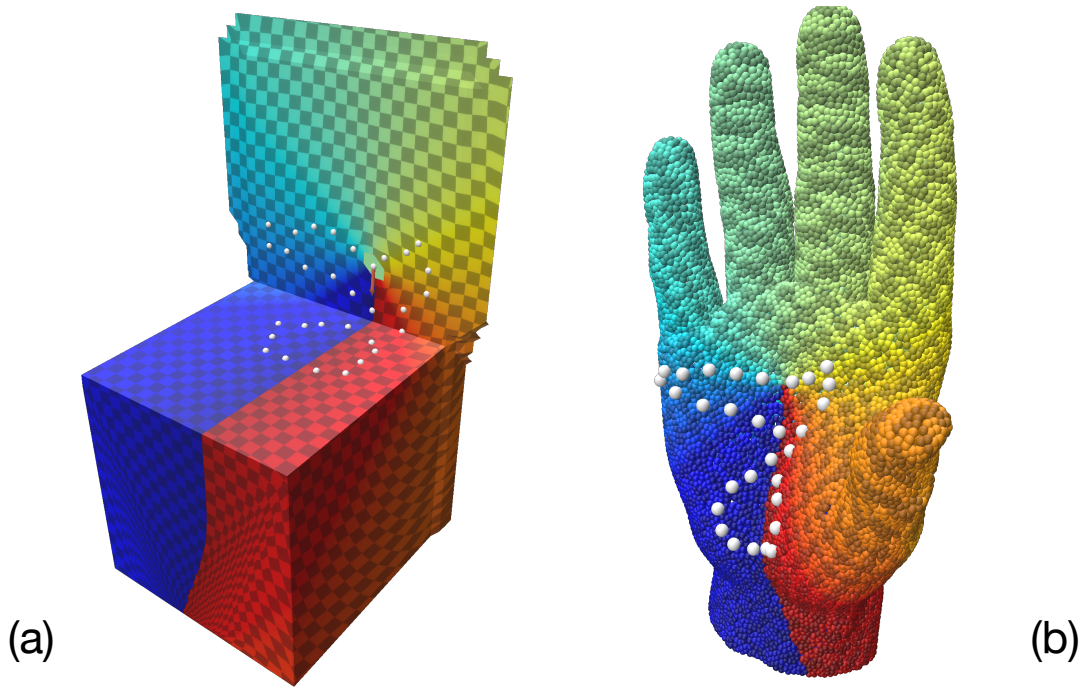

Figure 4.7: SPE reconstructions are robust to degenerate or completely unconnected domains such as (a) non-manifold meshes or (b) point clouds.

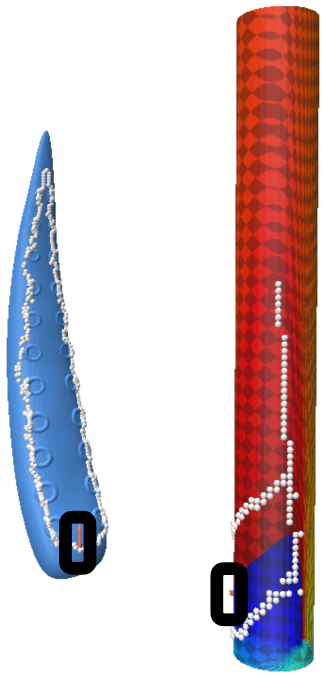

Figure 4.8: Reconstruction can completely fail if the logmap parameterization fails to be adequately constructed on the target domain. In such instances, the contact area can experience “foldover" by crossing large discontinuities in parameterized space.

Second, the logmap parameterization tends to degrade at distances far away from the origin, which again can result in reconstruction fidelity issues. In some cases the parameterization might even break down completely. Consider, for example, a contact area of a tentacle spiraling around a pole. As illustrated in Figure 4.8, in such situations the map can result in large discontinuities for nearby points that just happen to cross the angle singularity, which can result in highly inaccurate reconstruction. Additional sources of reconstruction noise include noisy geometry or contacts parameterized on non-convex shape regions. Third, this model cannot handle any non-isometric changes in contact shape whatsoever. Finally, reconstruction can be extremely slow if many candidate elements are available.

4.4 Curve / Axis-Embedded (AE) Model

4.4.1 Overview

The curve-embedded model was originally introduced to address multiple limitations of the SPE model. The modification is straightforward: rather than using a logmap originating from a single point, we instead distribute the embeddings over a curve. Because the model draws inspiration from the idea of medial axes [Guay et al. 2013], we instead refer to it as the “axis" embedded (AE) model — the axis in this case just being a piece-wise approximation of a curve. We define the axis as:

1. a finite set of points $\{a_1, \ldots, a_m\}$, with a *shortest geodesic* g_i connecting each pair of adjacent

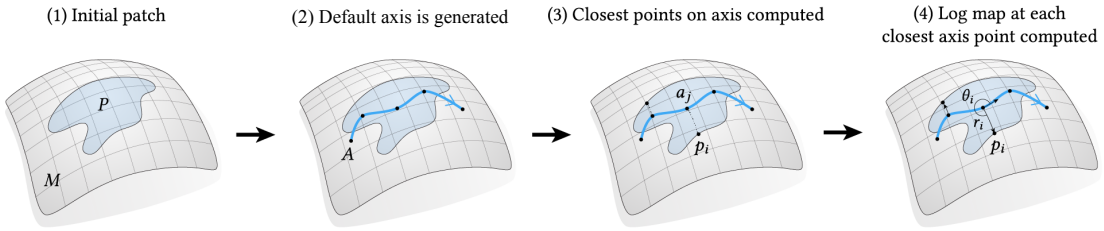

Figure 4.9: Illustration of the AE model packing procedure.

points (a_i, a_{i+1}) for $i = 1, \ldots, m-1$

2. *turning angles* $\{\phi_i\}_{i=2}^{m-1}$, where each ϕ_i is the angle of rotation from the ending direction of g_{i-1} to the initial direction of g_i, expressed in the tangent space of a_i

Then for every discrete contact area point p, we store:

1. its closest axis point a^*

2. its logmap parameterization $\log_{a^*}(p)$, with the angle parameter θ_p taken relative to the outgoing tangent direction of a^*. If $a^* = a_m$ for some i, then θ_p is taken relative to ending direction of g_{i-1}

Note that the outgoing tangent vector direction of all axis points except the first is not stored explicitly, but rather is computed relative to the incoming tangent vector from the previous point via the turning angle. The first point in the axis is special in that it is the only point which requires an explicit shape space coordinate and outgoing tangent vector direction. All other points on the axis can be fully reconstructed from the parameterized model.

4.4.2 Packing

Figure 4.9 provides an illustrated overview of the axis model packing procedure. Similarly to the SPE model, an axis must first be designated. We can naively generate a default axis by selecting the two points in the contact area furthest apart by heat distance, drawing a geodesic between the two points, and selecting all edge crossings as intermediate axis points. Additionally, by explicitly tracing out the geodesic, we can easily extract turning angles and, by doing so, convert the axis to the representation required by the model.

Once the axis is available, we must next compute the closest axis point to each contact point. We can do so by performing a short time scalar diffusion of all the axis point index values over the underlying surface, again using the Vector Heat Method [Sharp et al. 2019]. We then simply evaluate the diffused scalar field at the contact point's shape space coordinate, round to the nearest whole number, and select the axis point with the computed index as a^*. From there we can compute the logmap parameterization of the contact point in exactly the same way as the SPE model.

4.4.3 Reconstruction

Unlike the SPE model, we simply invert the logmap representation to recover the expmap in the axis model. Concretely, we trace geodesics using the converted expmap parameters computed during packing. The contact point is then converted back to shape coordinates by adopting the point reached at the end of the geodesic trace.

4.4.4 Supported Operations

Translation The AE model is also trivial to translate because only the starting point of the axis must be updated — the representation of the parameterized contact points does not change whatsoever, and the patch can be fully reconstructed from the newly updated location of the translated point. Similarly to the SPE model, we must also account for the tangent vector direction change by performing a parallel transport of the starting tangent vector direction to the final destination.

Rotation Also similar to the SPE model, the AE model also automatically assigns an orientation by logmap parameterization. Therefore, we can change the orientation simply by designating the first point in the axial curve as the arbitrary pivot for convenience and alter the outgoing tangent vector direction at said point. Because this direction change only impacts the reference angle in both models, no changes need to be made to the parameterized points — they can be reconstructed as-is following the new reference angle. Rotations about any pivot point can also be generated by combining rotation with translation.

“Isometric" Deformation An “isometric-as-possible" deformation can be induced simply by altering any of the intermediate turning angles of the axis. The method of deformation bears strong similarity to the way kinematic chains are used to control the shape of characters, with the only difference being the domain of application (e.g. on an arbitrary manifold vs. Euclidian space).

“Linear" Interpolation The AE model supports interpolations of translations, rotations, and isometric deformations albeit by different means. We will discuss each briefly.

Translation. Translation interpolation is performed identically to the SPE model, but instead with respect to the first axis point rather than the single embedding point.

Rotation. Rotation interpolation is also performed identically to the SPE model, but again with respect to the first axis point rather than the single embedding point.

“Isometric" Deformation. Isometric deformation interpolation uses the same principle as rotation interpolation, but applied to an intermediate axis point rather than the start point.

Hierarchical Composition The axis of the child contact region can simply be appended to the axis of the parent. In such an instance, the parent's terminal point and the child's start point must be converted to intermediate points by means of computing a turning angle; however, no re-parameterization is necessary. Translation, rotation, and isometric deformation can then be performed normally on the combined grouping. The composition can thus be uncoupled by discarding the turning angles and reverting the converted intermediate points back to ending and starting points respectively.

Transfer The axis model is capable of supporting transfer operations provided both domains are endowed with connectivity. As with the SPE model, only the first axis point and its associated tangent vector are stored using explicit shape coordinates. Therefore, if the corresponding location and tangent vector direction are provided on the target domain, the entire remaining contact area can be constructed using the existing parameterization. Although the AE model cannot accommodate unconnected domains like point clouds due to its reconstruction via the expmap, it instead can provide high quality reconstructions on coarse meshes and is much faster to reconstruct on fine meshes.

4.4.5 Benefits & Downsides

The AE model, true to its original design objectives, addresses many of the shortcomings of the SPE model. Like the SPE model, it also guarantees preservation of the original number of contact points; however, it does not rely on element clamping. It is therefore the only model of the three discussed that is suitable for coarse meshes. Furthermore, the reconstruction is extremely fast by comparison. The model also supports "isometric" deformation by enabling contact areas to "bend" via updates to the turning angles of intermediate points. Finally, it is more reliable than the SPE model in terms of parameterization because it can also handle complex instances such as the tentacle pole example discussed earlier (see Figure 4.10).

However, the model suffers from many similar drawbacks as well. The model reconstruction still tends to degrade at regions far away from the axis; however, this drawback tends to be largely addressable in practice because the axis can simply be adjusted to be close to every point in the region. But reconstruction error on noisy geometry or contacts parameterized on non-convex shape regions still are a problem. This model also cannot handle any non-isometric changes in contact shape. Finally, unlike the SPE model, this representation is confined to meshes due to its heavy reliance on geodesic tracing. While it is possible to accelerate the SPE model by using expmap reconstruction when connectivity is available, it is not possible to use the axis model on point clouds whatsoever.

4.5 Summary

We have presented three different parametric models of contact areas, including discussions regarding the packing and unpacking methods for, and operations support by, each model. However, none of the models presented is a silver bullet — they all come with benefits and tradeoffs. The best model is thus dependent on the use case. We also note that all representations are purely geometric and lack notions of physical quantities such as force or associations to

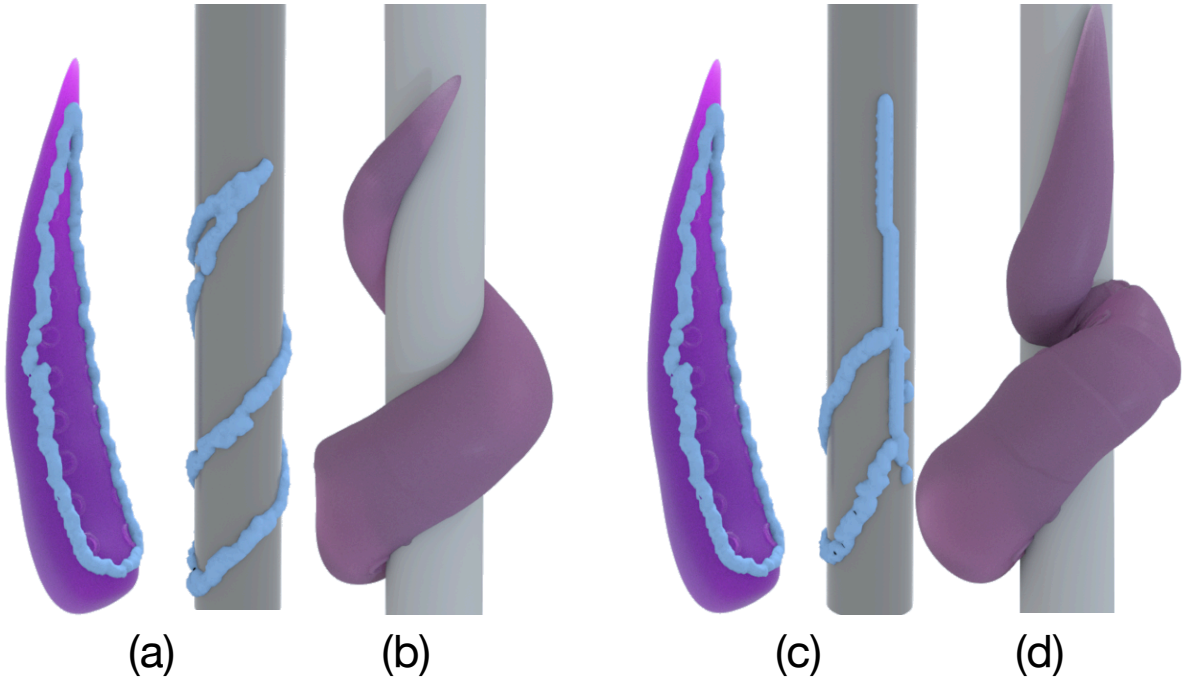

Figure 4.10: Transfer comparison of a tentacle contact to a cylinder using (a) the axis-embedded contact model and (b) its generated solution against (c) the single-point-embedded contact model from and (d) its generated solution

mass-inertial properties from the systems (e.g. fingers, objects) that induce them. We explore augmenting areas with physical parameters in Section 9.2.

Chapter 5

Artist Tools for Grasp Drafting

By taking advantage of our proposed models and their respective operators, we propose a comprehensive suite of tools that can be used to draft contact-rich interactions between dexterous hands and object either from previously compiled data or from scratch. In doing so, we will highlight practical and realistic use cases which significantly benefit from one or several of our proposed models.

5.1 Motivation

With numerous research efforts today aiming to automate the grasp generation pipeline [Turpin et al. 2023; Zhang et al. 2024a; Wan et al. 2023; Wang et al. 2023; Chen et al. 2024b; Chen et al. 2024a; Li et al. 2007], many of which have demonstrated very high quality results, we may be tempted to ask a simple question: why are artist tools even necessary? Why can't we just invoke a state of the art grasp solver with the hand and object we are interested in, and why have such approaches not already been widely adopted?

One problem with fully automated data-driven solutions [Zhang et al. 2024a; Wan et al. 2023; Chen et al. 2024a] is that they often trade off customization in favor of scale. For example, a grasp solver trained over a large dataset may provide high quality solutions to scenarios where the hand, object, and desired grasp are adequately represented in the dataset; however, they may struggle to generalize to objects or desired grasp poses that significantly deviate from the dataset distribution or fail entirely if the setup does not conform to the assumptions the dataset is built on (e.g. the desired hand does not match the hand which is assumed by the dataset). It is also infeasible to capture the space of all possible hand assets, especially since many assets are custom (e.g. developed internally by a studio for a particular media production or by a designer for a particular robot).

Other automated methods are intended to operate on specific instances [Turpin et al. 2023]; however, in order to make up for the lack of human supervision, they often must employ more sophisticated tools such as representing objects as signed distance fields (SDFs) or using differentiable simulation to optimize over non-trivial events such as hand-object and self contact. Although powerful, these tools are often expensive to integrate in terms of software engineering overhead, which is problematic in situations where additional development time cannot be

strongly justified. Such methods also tend to be slow in comparison to the time taken by an artist and may not even necessarily generate the desired result. There is therefore still a strong need for tools that can work in conjunction with artists and deliver fast and reliable results which can be quickly iterated upon.

5.2 Contributions

This chapter aims to provide the foundations necessary for integrating grasp drafting from contact areas into production workflows directly, intuitively, and robustly. To do so, we present *end-to-end art-directable (EAD)* tools for defining contact areas, directly editing contact areas, transferring contact areas between surfaces, and using the resulting correspondences compute the optimal corresponding hand poses automatically.

Our focus on EAD tools, which enable animators and designers to create results either from scratch or previously collected data entirely under the tool framework, is intended to address the challenges of practical real world situations. In particular, they are intended to work robustly on a wide range of custom geometries such as those collected from reconstructed 3D scans, operate at real-time interactive speeds whenever possible, and, above all, be transparent and easy to both work with and understand. Additionally, we will also show how these tools can be used to draft out simple manipulation animations.

5.3 Workflow and Assumptions

Figure 5.1 illustrates the workflow we aim to enable, which requires the ability to create and manipulate contact areas, transfer areas between surfaces, and compute the appropriate pose. We assume that the desired hand and object have already been sourced, and that the kinematics of the hand are known and well defined. To this end, we will discuss the creation of contact areas, editing operations artists can perform on the areas and the interfaces for doing so, and the pose computation pipeline after contacts are drafted. As noted in Figure 5.1, we intend for this process to be iterative either in response to input from a project stakeholder (e.g. art director) or until a result of desirable quality is achieved. We also assume that no geometries besides the hand and desired grasp object are present in the scene.

5.3.1 Sourcing Contact Areas

Contact areas can be sourced from scratch by the artist through direct painting, or alternatively from the real world (Chapter 9). In both cases, we require contacts to only be sourced on a single entity (i.e. either the object OR the hand). We discuss both options below and assume the extracted contact areas are initially represented as an unordered set of shape space coordinates.

Direct Painting Contacts can be directly painted either using off-the-shelf brush tools already provided by existing 3D modeling software (e.g. Maya, Blender, etc.) or by our custom paint brush. We leave the choice up to users but briefly discuss use cases for both options.

On fine meshes or regions with dense sampling, users may find it beneficial to select existing discrete vertices. Vertex selection is flexible, inexpensive, has no parameters beyond the brush

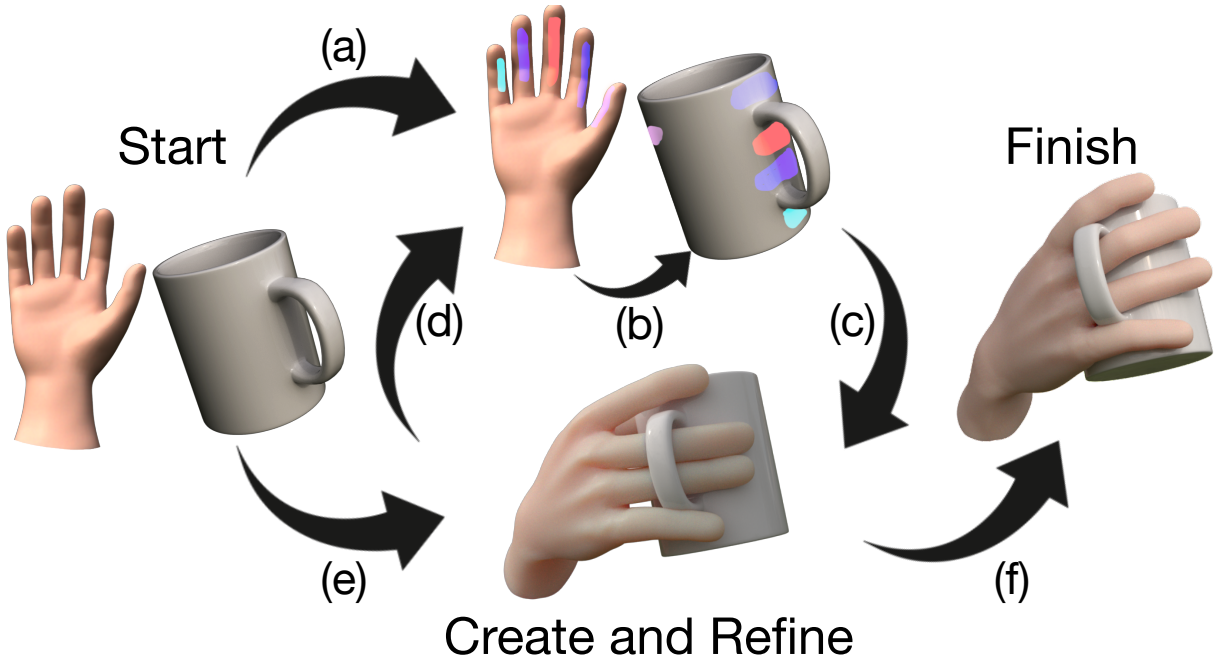

Figure 5.1: High level overview of our artist tools workflow. Starting from a given hand and object, the artist typically starts by (a) creating contact areas on the hand and manipulating them into their final positions before (b) transferring them to the object to generate a corresponding set. Artists can then (c) compute the manipulator pose from the contact areas and (d) use the result as feedback to edit the contact areas on the hand and object in an iterative feedback loop. Artists can alternatively (e) first create a (often coarse) pose using traditional posing tools and designate contact areas afterword. (f) Refinement continues until the final result is acceptable.

size, and is a method familiar to most artists. The drawback, however, is that this method does not enable selection of arbitrary points such as those within faces or in between edges, which leads to problems creating dense contacts on coarse meshes.

In response, we created our own brush tool which enables arbitrary point selection by discretizing a NURBS curve drawn in screen space and projecting the points into mesh space via standard ray-mesh intersection. Our custom brush offers a solution to the coarse mesh problem; however, it requires users to designate possibly unintuitive parameters such as the discretization step size and does not provide variable brush stroke sizes.

For these reasons, we allow heterogeneous creation of contacts using both brushes, which combined permit users to work directly with the original desired triangulated mesh, even if the mesh is irregularly sampled or has poor triangulation in general.

Real World Capture Chapter 9 thoroughly details techniques for imaging and reconstructing contact areas as textures. After the final textured mesh has been generated, artists can simply examine the texture and paint over the imaged contact regions.

5.3.2 Contact Editing

The extracted contact regions may not necessarily be generated exactly where or how the artist intended. We therefore support editing operations that can transform the regions into the desired configurations.

Due to the focus on artist-driven contexts, we assume that both the hand and object are represented by triangulated meshes and that the coarseness of each mesh is unknown. Therefore, we select the AE model as the representation for editing. We begin by parameterizing all extracted contacts into AE representations and assign a default axis by tracing a geodesic between the furthest separated points by geodesic distance.

We support all AE model operations outlined in Chapter 4 for contact editing. We also support a fourth operation which permits the user to replace default axis with one of their own choosing. We detail the interface for each editing operation in this section. Create Figure

Translation We support translation through a drag interface. For convenience, we only allow translation of the first axis point. After selecting the appropriate axis point, the user can hold and drag the axis point to a new location on the mesh. The target point is determined through standard screen-space ray intersection. After the destination is determined, we parallel transport the outgoing tangent vector direction and simultaneously move the axis point to the selected location. We also reconstruct and visualize the axis while it is in transit to provide user feedback; however, we refrain from reconstructing the entire contact to reduce computations and maintain realtime speeds. After the user releases the drag, the axis and full contact are reconstructed at the final destination to complete the edit.

Rotation To support rotation, we render a disk at the starting axis point which can be re-sized arbitrarily. The user initiates a rotation by selecting any location on the disk and beginning a drag selection. The total arc length of the disk covered during the drag in screen space is then converted back to world space, after which the corresponding angle is extracted. The tangent vector is then incrementally rotated by the extracted angle. Users can therefore adjust the size of the disk to select larger (smaller disks) or smaller (larger disks) step sizes for contact rotation. As with translation, a preview of the intermediate reconstructed axis during the drag operation, and the full axis and contact are reconstructed upon release to complete the operation.

Isometric Deformation We enable isometric deformation in the same way as rotation via the disk interface. However, rather than being applied to the start point, the disks are instead applied to all intermediate axis points. Rotation of any such intermediate disk thus alters the turning angle parameter of the corresponding axis point, thereby inducing an isometric deformation under the axis model.

New Axis Selection To generate a new custom axis, we simply allow users to select two points. We then trace a geodesic between the selected points and take all triangle edge crossings as intermediate axis points. The first point in the selection is used as the starting axis point, while the second point serves as the last point in the curve. We then re-parameterize the existing contact area to the new axis.

5.3.3 Contact Transfer

Similar to new axis selection, contact transfer is also initiated by the user selecting two points on the target shape. The first point is used to determine the starting point of the transferred axis, while the second point is used to determine the direction of its associated tangent vector. The transferred contact can then be fully reconstructed from the existing parameterized representation, with the caveat that the area must be mirrored. A dense bijective pairing is automatically generated between the source and transferred contact area and visualized via paired lines. The user can also edit the transferred contact at will without breaking the correspondence, or alternatively manually break the correspondence if desired.

5.3.4 Grasp Computation

After all contact areas have been transferred, we automatically compute the desired hand pose using optimization-based inverse kinematics. Specifically, we solve the following equation:

$$\begin{aligned} \boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \quad & \textstyle\sum_{i=0}^{N} \ (\lambda_d \Gamma_{D,i} + \lambda_n \Gamma_{N,i}) + \sum_{j=0}^{J} \ \lambda_p \Gamma_{P,j} \\ \text{s.t.} \quad & \boldsymbol{\theta}_L \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}_U \end{aligned} \tag{5.1}$$

where $\boldsymbol{\theta}$ is the degree of freedom vector, N is the total number of corresponding contact vertices, $J = |\boldsymbol{\theta}|$, $\boldsymbol{\theta}_L$ and $\boldsymbol{\theta}_U$ define the lower and upper bounds of $\boldsymbol{\theta}$ respectively. We define the remaining terms as follows.

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

$$\Gamma_{D,i} = ||\mathbf{p}_{o,i} - \mathbf{p}_{h,i}(\boldsymbol{\theta})||_2^2 \tag{5.2}$$

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

$$\Gamma_{N,i} = (1 + \mathbf{n}_{h,i}(\boldsymbol{\theta}) \cdot \mathbf{n}_{o,i})^2 \tag{5.3}$$

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:

$$\Gamma_{P,i} = ||\boldsymbol{\theta} - \boldsymbol{\theta}_P||_2^2 \tag{5.4}$$

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. We formulate and solve Eq. 5.1 using the NLOPT optimization library [Johnson 2017]. Figure 5.2 illustrates successive guesses over the course of a single optimization call.

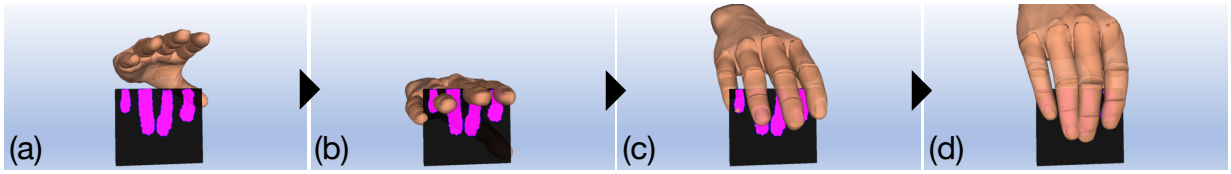

Figure 5.2: Optimization progress for finding a feasible box power grasp at (a) 0, (b) 250, (c) 800, and (d) 1,000 iterations.

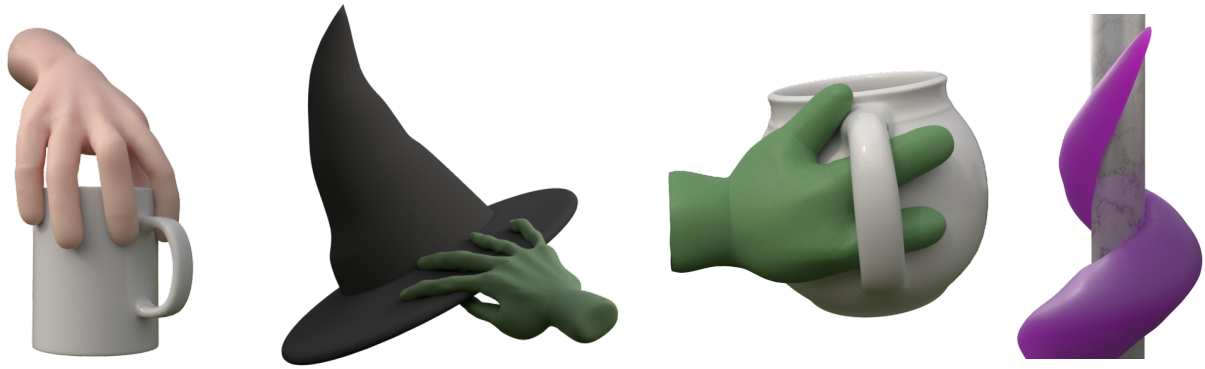

Figure 5.3: Poses created using our contact edit framework across kinematically and morphologically diverse manipulators in realistic in-situ contexts.

It is worth noting that Eq. 5.1 does not contain a collision penalty term as suggested by several prior works [Brahmbhatt et al. 2019b; Meixner et al. 2019; Hazard et al. 2018; Grady et al. 2021]. Omission of this term is intentional. Empirically, we found that doing so led to both substantial drops in solution discovery time and improvements in the proposal of solutions for difficult contact maps.

5.4 Results

We built our artist tools in Autodesk Maya due to the tool's wide existing familiarity with animators. The tool was distributed to an animator, who subsequently used it to generate all results illustrated below.

5.4.1 Static Grasps

Figures 5.3 and 5.4 illustrate some high quality poses generated by an animator using our framework. Generating poses such as these using the traditional process of manually adjusting joints is time consuming and iterative; however, our approach allows users to quickly set up the desired shot by working directly with contact areas, after which the manipulator pose is automatically calculated in a manner that accounts for both rig kinematics and skin deformation.

Figure 5.5 also illustrates some results generated with fully articulated robot hands as well as a rigid approximation of a human hand. Although these hands are comprised of disconnected



Figure 5.4: Additional poses generated by our solver across various manipulators and geometries.

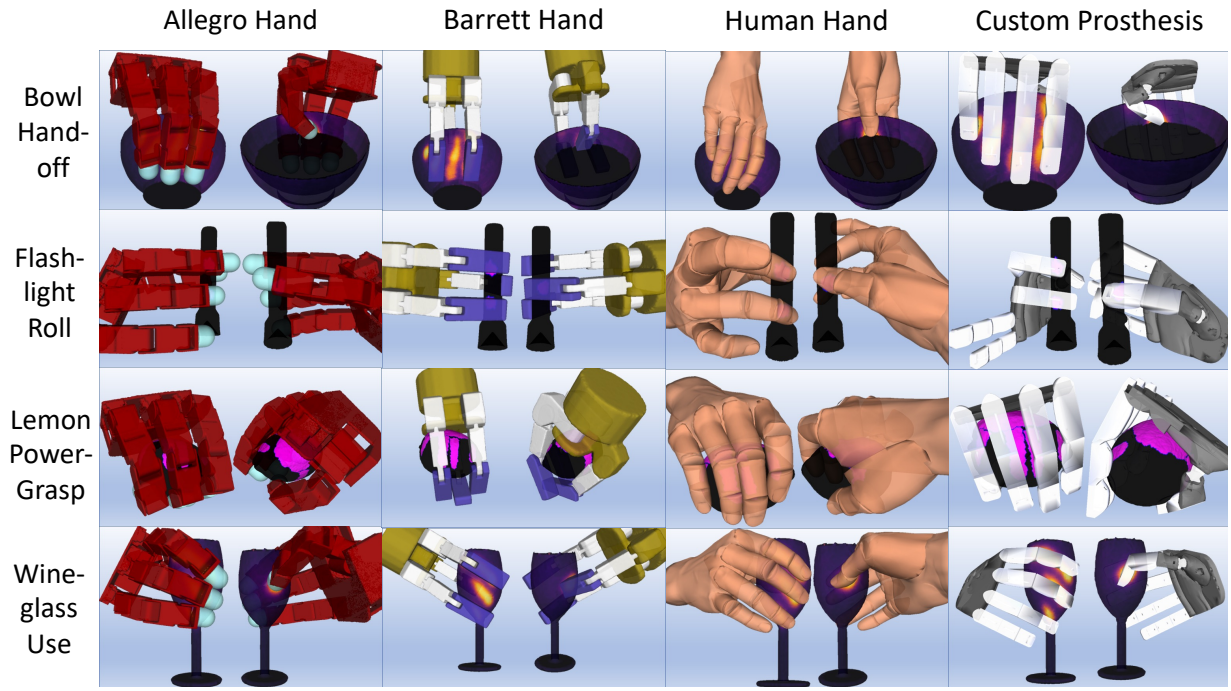

Figure 5.5: Grasps synthesized as a result of our transfer procedure across a variety of kinematically diverse robotic manipulators, objects, and grasps.

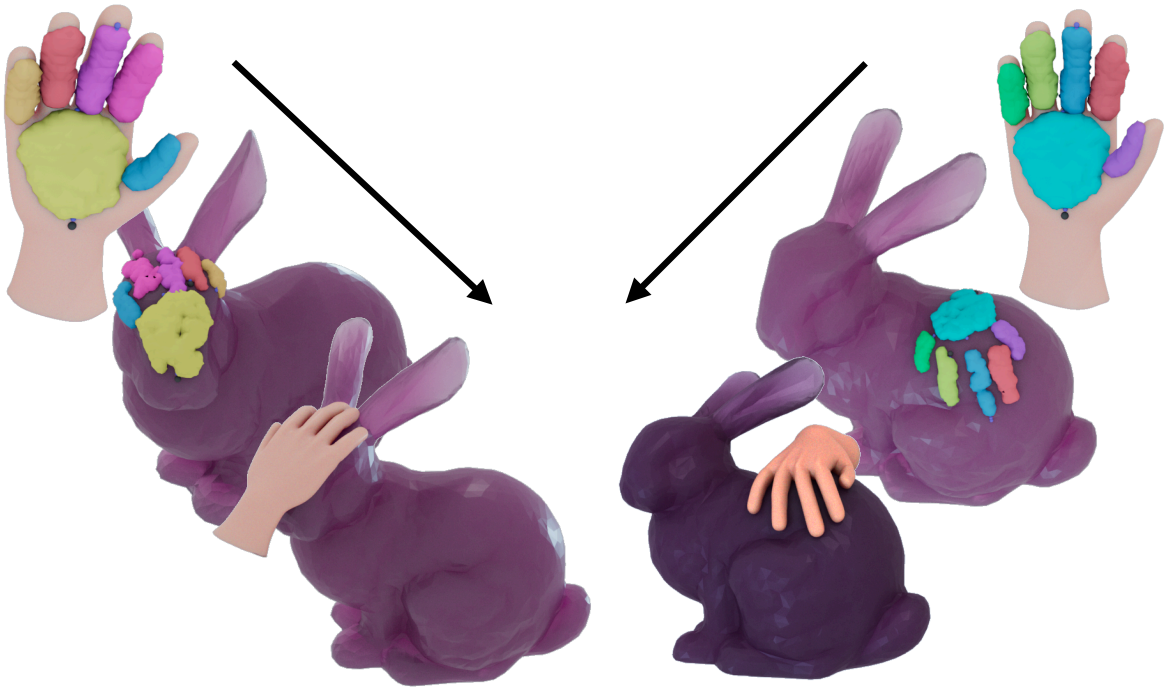

Figure 5.6: Bulk transfer of a hierarchically organized identical set of contacts on the hand complies to local geometric features on the head (left) and body (right) of the Stanford bunny.

geometries, we can generate a manifold wrapper and perform contact editing on the wrapper as normal. Once complete, the points can be projected back to the articulated meshes via closest point queries [Sawhney 2021]. The resulting optimization pipeline can then be performed normally.

Figure 5.6 shows an example of quickly drafting multiple grasps on different parts of the bunny quickly by taking advantage of hierarchical composition. In this case, multiple contacts were painted on to the hand, composed together, and then mapped to different areas of the bunny with a few mouse clicks. Our as-isometric-as-possible transfer process adapts well to the different geometric features on the bunny surface.

5.4.2 Difficult Grasps on Higher Genus Topologies

Routing multiple fingers through holes and around handles is an especially challenging task due to the large number of inter-object and self-intersections generated during traditional posing, many of which would require movement of the palm and a subsequent total repositioning of the fingers in the updated state to resolve. Figure 5.7 shows two examples. Using contact areas allows the animator to designate which parts of the handles are desirable for contact and automatically generate a pose for the palm optimized for the desired grasp. Different fingers can also be routed towards specific handle regions to reduce self-intersections, while automated computation of the palm location can determine the optimal base pose position required to produce such a routing.

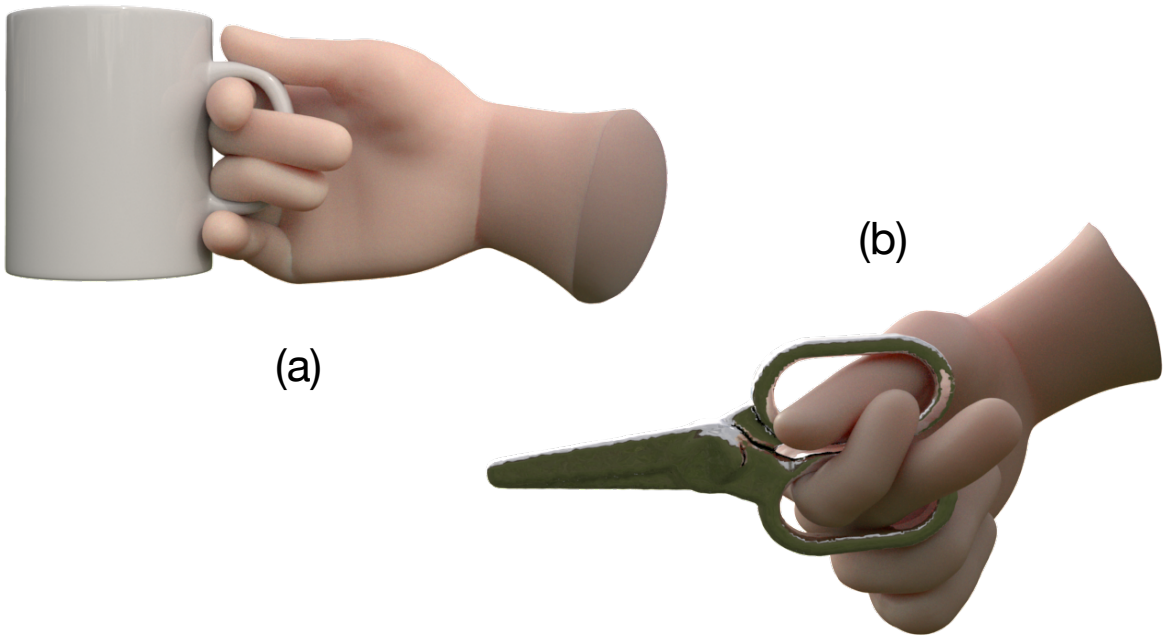

Figure 5.7: Challenging grasps generated using our method (a) curling the fingers around the cluttered interior of a genus-1 mug handle and (b) between the two holes of a pair of genus-2 scissors.

5.4.3 Rapid Grasp Transitions

Figure 5.8 illustrates a sequence of transitions between multiple pitching grips, each of which was constructed through direct manipulation of contacts on the surface of a baseball. Each transition required exactly three manipulation operations: two translations and one rotation to go from the knuckleball to the fastball, and three translations to move from the fastball to the curveball. The entire sequence was generated by an untrained animator using reference photos in less than an hour after initial scene setup. Subtle details, such as closing the gaps between fingers, were made by creating corresponding contact patches on different fingers of the same mesh.

5.4.4 Drafting Animations

We can also use the discussed static grasp framework for creating simple animations through joint-space linear interpolation. We discuss three interesting use cases: changing the hand pose in between a fixed set of contacts, changing the hand pose with respect to object scale, changing both the hand pose and the contacts together, and finally extending grasp synthesis to full body posing.

Just as we can typically keyframe poses, we can also keyframe parameterized contact distributions. For ones that do, we can simply create new contacts at a desired later keyframe. In such instances where the number of necessary areas to draft are relatively small and joint space interpolation between computed grasp solutions is limited to “nearby" configurations (e.g.

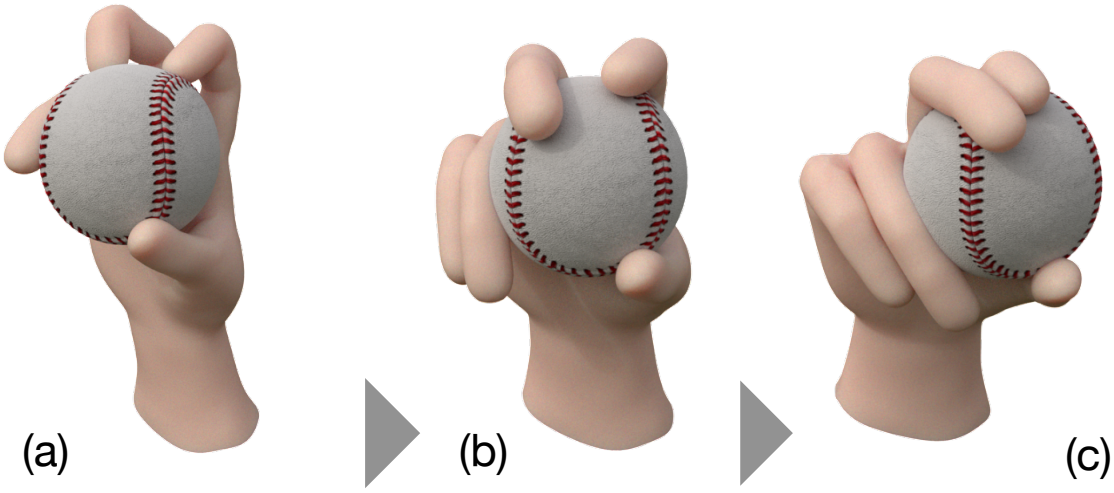

Figure 5.8: Our technique can create smooth transitions between multiple baseball pitching grips, including a (a) knuckleball, (b) two-seam fastball, and (c) curveball.

transitioning between grips on a baseball), this type of workflow is highly effective. Figure 5.8 illustrates a compelling example of such a use case.

Figure 5.9 illustrates and example showing adaptation to changes in object scale. This sequence was constructed using 4 keyframes in under 45 minutes by an inexperienced animator, and in particular highlights the advantage of hierarchical composition. To do so, we transferred a fixed set of hand contacts – the same ones in Figure 5.6 – to an identical starting point and axial direction of the sphere at each scale, and use the solution from the previous keyframe as a prior for the subsequent frame to reduce optimization time.

Figure 5.10 illustrates a manipulation sequence used to pull a box of sugar off a kitchen shelf. Since contact distributions are strongly associated with a particular time of the object trajectory, it is easier to simply interpolate the contact regions and solve for the grasps automatically rather than keyframe contacts directly. Contact area interpolation also allows finger-grained control over hand posing during intermediate solves.

Finally, although we have primarily considered manipulator-object interaction in this paper, the general-purpose nature of our contact formulation permits application to arbitrary contact-rich contexts. Figure 5.11 illustrates an example in which various parts of a full body are designated to make contact with the chair and floor. To animate this sequence from the initial standing position after specifying the contact sets, we start by fixing all joints and solving for the position of the root joint at the center of the pelvis. Next, we unlock the lower body and solve for the leg positions which align the feet with the floor. Finally, we unlock the left and right arm subsystems and solve in order to bring the arms inward. Another use case is to minimize foot movement during the in-betweens of the sitting procedure, which we can do by solving for the foot contacts while the remainder of the upper body continues with the sitting interpolation.

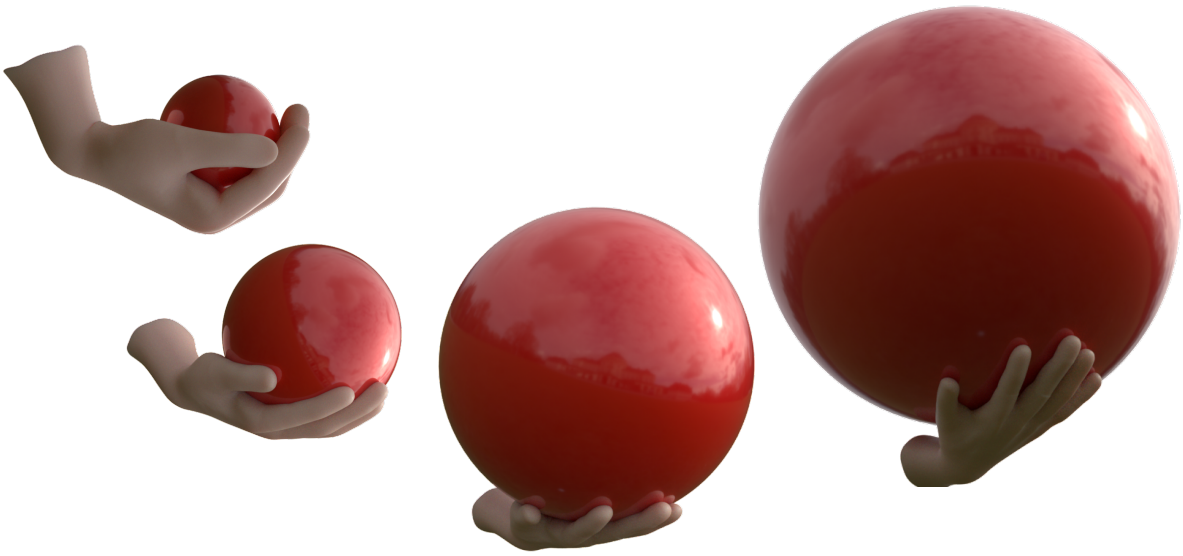

Figure 5.9: Demonstration of a manipulator adapting its grip in response to variance in object scale.

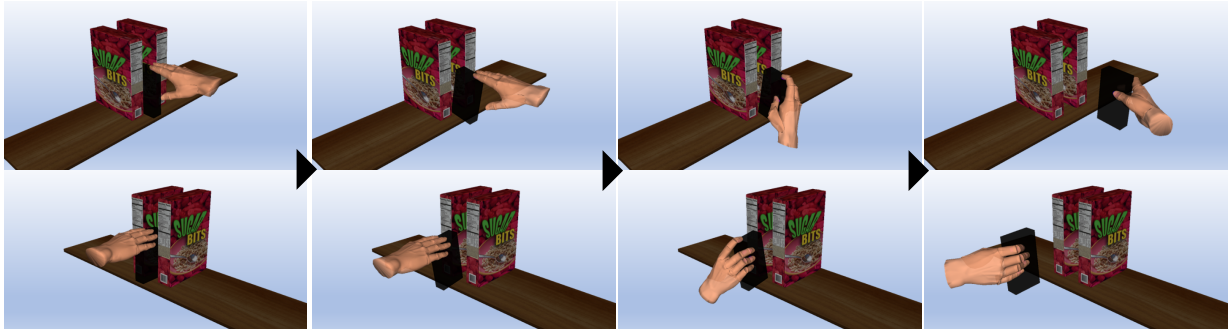

Figure 5.10: Demonstration of using the method to quickly approximate a kinematic grasp time series spanning a full simple manipulation of pulling a box of sugar off a kitchen shelf. Poses for successive manipulation steps can be generated quickly due to the inclusion θ_p and contact area interpolation.

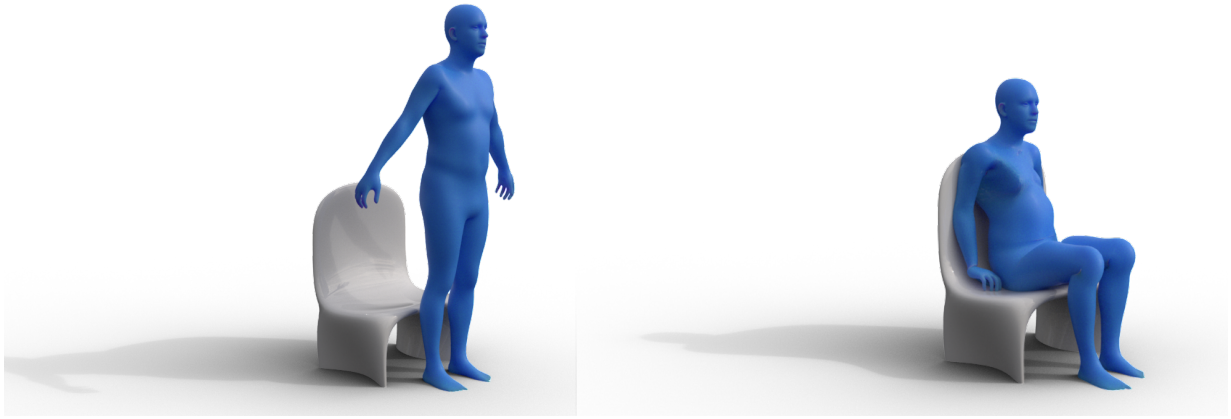

Figure 5.11: Result of applying our method to designate contacts on a full body (a) in preparation for sitting in a chair and (b) settled into the final pose.

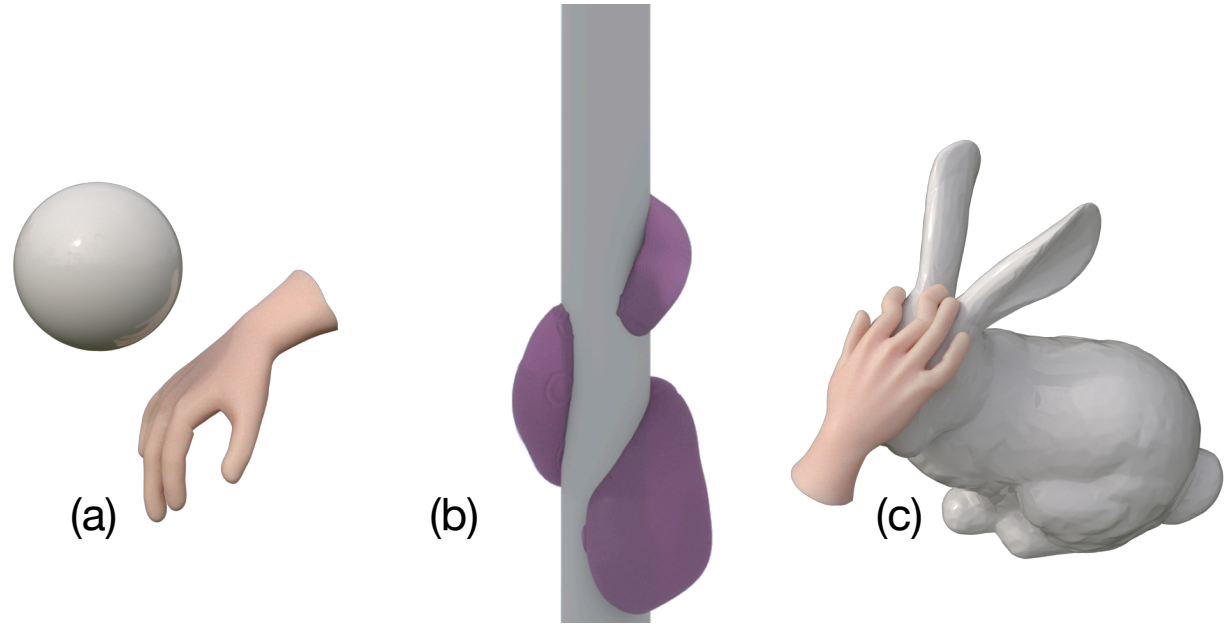

Figure 5.12: Catastrophic failures resulting from the removal of optimization terms, including (a) distance, (b) normal, and (c) prior penalties.

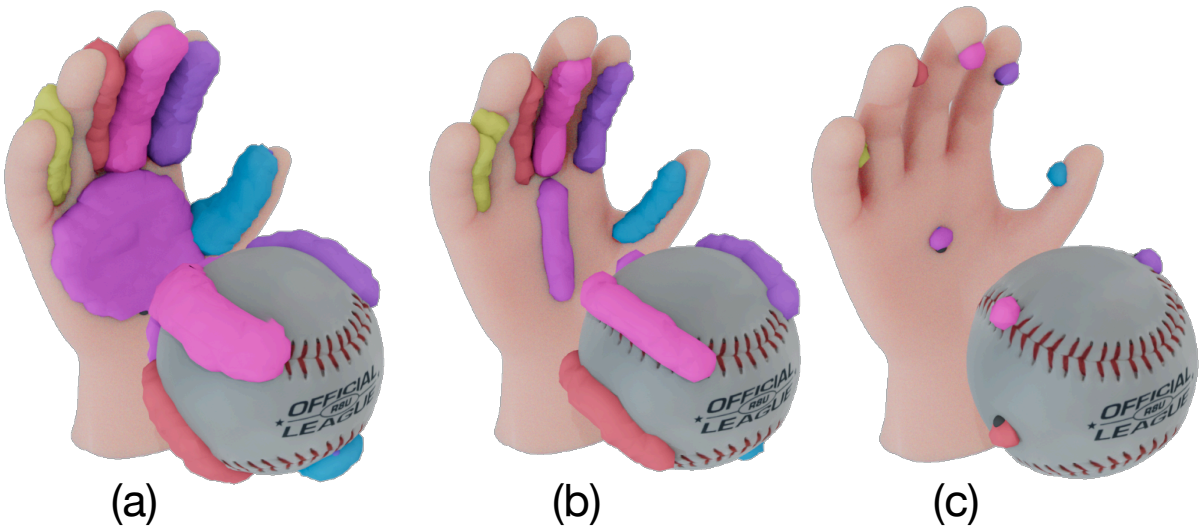

Figure 5.13: Contact sets used as the basis for pose solution comparisons using (a) our axis based area model (b) a curve-only model and (c) a single point model.

5.4.5 Ablation Studies

To demonstrate the necessity of each term in Eq. 5.1, we performed an ablation study in which each optimization term is individually removed. Figure 5.12 illustrates the results of the study. As anticipated, removal of any terms results in catastrophic failure.

Removal of Γ_D removes all incentive to progress towards the contacts, resulting in failure to even reach the desired object. Removal of Γ_N permits negligence of the geometry at the contact point locations, which creates ambiguity in distinguishing the object interior and exterior. This omission caused the tentacle to find a solution which conformed to the inside of the pole during part of the wrap rather than the outside. Finally, removal of Γ_P commonly results in highly undesirable pose artifacts such as contorted fingers and large regions of self-intersection. We note that this omission is particularly problematic in the context of unconstrained problems where there is otherwise no quantitative regularization to prevent unnatural behaviors from emerging.

5.4.6 Comparison with Single Point Contact Model

We also perform a comparison of the solutions generated by our optimization framework utilizing contact areas against both traditional single point IK and a “curve-only" variation inspired by the Line of Action [Guay et al. 2013]. Figure 5.13 illustrates the contact sets used as the basis for each solution. Full contact areas were approximated as single points and curves respectively on a best-effort basis. We found that responses were fairly consistent regardless of curve or point placement in ambiguous cases such as the large palm contact. Corresponding object points were selected individually in the single point formulation and were transferred using our axis-based solution in the case of the curve and area formulation, where the curve itself was used as the axis in the former case. All solutions were computed from same start pose using the same solver

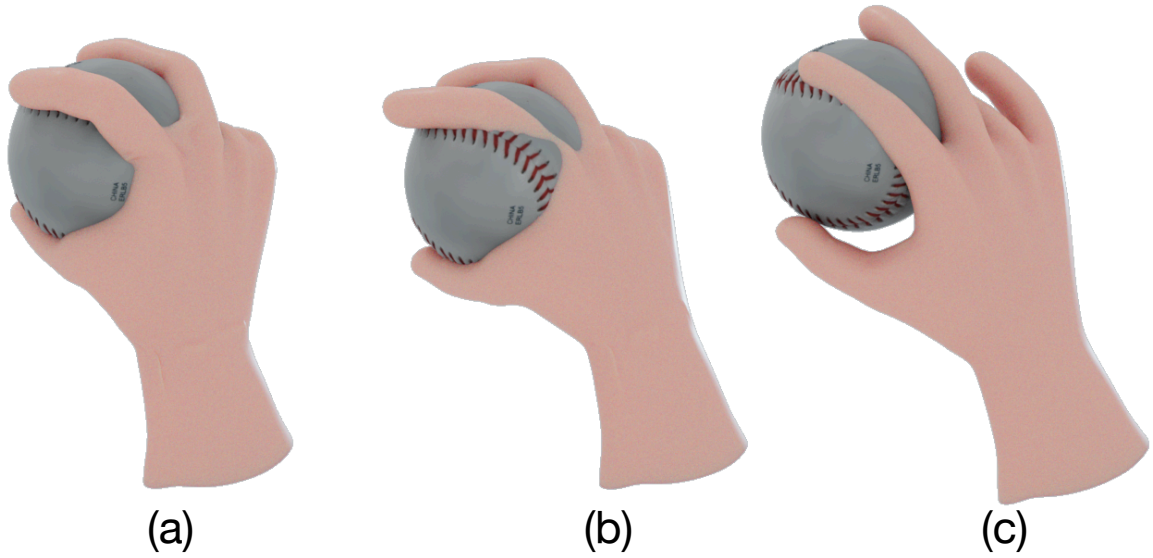

Figure 5.14: Poses computed using (a) Our axis-based area model (b) A curve-only model and (c) A single point model

and number of iterations, but note that weighting coefficients λ_d, λ_n, and λ_p had to be altered to account for differences in point density. Specifically, the values used are:

Table 5.1: λ weighting values used to generate results in Figure 5.14.

λ Weighting Values			
	λ_d	λ_n	λ_p
Single Point	3.5	6.1	10.0
Curve Only	1.0	1.2	10.0
Full Area	1.0	1.0	10.0

Figure 5.14 illustrates the poses computed under each formulation, while Table 5.1 provides the weighting coefficients used the generate the results. We note that both the single point and curve-only formulations resulted in solutions which produced highly unnatural finger configurations or significant penetration between the palm and the object respectively. The single point formulation also required extensive testing of contact point placement and hyperparameter tuning even to produce a somewhat similar result and was highly sensitive to changes, while curve and area formulations were less sensitive to changes. Importantly, we note that the choice of λ-values for the area based contacts not only produced the result in Figure 5.14, but also **every single grasp result shown so far**. The curve formulation also notably converged to similar solutions regardless of how the palm curve was oriented, assuming the number of points comprising the curve stayed the same.

We hypothesized and verified that the curve and area formulations produce similar solutions if contacts are confined to regions with an implicitly defined orientation (e.g. fingers only). The

failure of the curve formulation to generalize to regions such as the palm, however, underscores a general limitation: given two curves of the same length, it is not possible to non-uniformly weight the importance of one curve over another without either artificially making one curve sparser than the other or introducing more hyperparameters to adjust the weighting. In contrast, an area formulation implicitly provides non-uniform weighting capability by design. We note this observation to be particularly valuable in the context of artist tools since users already make this designation during creation of the contact areas themselves, which improves overall intuitive understanding of tool usage in comparison to tweaking weighting hyperparameters. Additionally, the results demonstrate that, more generally, an abundance of correspondences is valuable in addressing several well-known drawbacks in IK solvers.

5.5 User Studies

We conducted a small user study to gage the effectiveness of our plugin in the hands of real animators.

5.5.1 Subject Recruitment and Distribution

Five subjects over age 18 were recruited from a combination of channels including internal university mailing lists, in-class announcements, and referrals. Subjects were expected to have basic working proficiency in Maya, including the ability to control rigs and change camera views. Three out of five subjects came exclusively from an art background, while two had some technical background.

5.5.2 Setup

Participants were asked to partake in three one hour sessions and were compensated with a $20 Amazon gift card at the end of each session for a total of $60 per subject. Participants were asked to watch a 9 minute tutorial video prior to attending the first session but were not informed of session details in advance. All studies were conducted in-person with a proctor present during each session. Each session was conducted on a single workstation running an Intel i7 3.1 GHz processor with 8 GB of RAM. Sessions were screen-recorded without audio to limit exposure of personally identifiable information (PII). Participants were additionally provided a Maya scene with a skinned and rigged hand asset, a 3D digital scan of a physical object, and the physical object itself as reference material. 3D scans including geometry and textures were collected using an Artec Space Spider[1]. Participants were permitted full internet use during each session and were permitted to ask unlimited questions. Figure 5.15 includes all objects used during the studies. For simplicity, and to reduce the learning burden, we provided subjects access only to the default vertex selection paintbrush already included in Maya due to the high polygon counts of the 3D scanned meshes.

In the first session, participants were given an object by the proctor and asked to create as many good quality, natural looking poses as possible using the provided hand rig and object

[1]https://www.artec3d.com/portable-3d-scanners/artec-spider



Figure 5.15: All user study items

mesh. Participants were then asked to recreate the same poses on a best effort basis in the second session using view-only replicas of their original poses. Participants were tasked with performing the first and second sessions either using our plugin or existing Maya tooling only. The order of tool usage was randomized to reduce ordering bias. In the third session, participants repeated the first session procedure but were allowed to select any object in Figure 5.15 and use our tool as much or as little as desired. In an effort to simulate a production workflow, participants were not allowed to modify the provided hand rig, skinning weights, object or hand geometry, or asset scales; however, subjects were allowed to perform rigid transformations to the object or rotate any of the hand joints. Each subject was played an audio recording of directions at the start of each session and was sent a survey at the end.

Subjects were given access to the tutorial video and a textual reference during each session in which our plugin was used. Although participants were told to create as many high quality poses as possible or replicate as many poses as possible in the previous session using the plugin, our primary intention in the first plugin session was to provide subjects with hands-on experience with our tool, and in particular the processes of creating, manipulating, and clearing contacts, using and observing the optimization process, and navigating the general UI layout. To test the intuitiveness of plugin usage, we further randomly subdivided the first plugin session into two pools: participants in the first pool were provided guided suggestions by the proctor for exploiting plugin features, while those in the second pool only received answers to queries.

Rather than requiring users to designate an axis during contact creation time, we provided users with an automatically computed initial default axis to use with any newly created contact patch. We tried two methods for computing a default axis. In the first method, we simply connect the two furthest points on the patch with a geodesic. In the second, we use an approximation of

the patch's medial axis as the default axis: we use the heat method [**crane2017shm**] to compute a smoothed distance field u from the boundary of the patch, compute the gradient ∇u, and record points where ∇u changes direction by an angle of more than $\pi/2$ as points on the (smoothed) medial axis. These points were then greedily connected into a curve using Dijkstra's algorithm.

5.5.3 Study Results

Our findings were exciting and in some cases surprising. Although the subject pool was small, the response was generally positive. We elaborate upon detailed observations, positive feedback, and criticisms in the proceeding paragraphs.

One subject opted to use our plugin as a "last-mile" solver by painting contacts after creating an initial pose manually, while the remainder, as expected, used our plugin as a "coarse" solver by instead painting contacts in the beginning, solving for the pose automatically, and making minor adjustments manually at the end, if at all. 100% of subjects opted to paint contacts on the hand first and subsequently transferred them to the object. Three subjects opted to change the auto-generated default axis to an orientation consistent across all contacts. The remaining two, who opted to keep the default axis in all cases, opted to bring the object close to the hand before transferring contacts. All subjects could easily tell when transfer results were undesired and were able to quickly alter orientations via contact rotation. Subjects who heavily used contact rotation and translation did so with fine-grained movements and expressed satisfaction with the result. Local deformation was used sparingly. Three subjects opted to start from scratch for each new pose, while two chose to return to a template contact distribution for re-use across multiple poses and expressed interest in having an existing library for common regions such as fingertips or upper palm regions. All subjects that made fine-grained adjustments post optimization did so through manual rig adjustment rather than creating or modifying contacts. With the exception of one subject who created very sparse contact sets, users never modified any weighting hyperparameters. All subjects opted to visualize the optimization progress to understand how the final solution was generated and, more interestingly, how they could modify variables such as object or contact placement to produce even more desirable results.

Despite the large variance in usage across even a limited subject pool, the response from all subjects in the third session was positive. One subject opted to use the plugin exclusively during the third session, and the remainder chose to use both toolsets together. 100% of subjects claimed the plugin would be useful to other animators and found the tutorial video alone sufficient for using the tool regardless of whether guided suggestions were provided in the introductory session. The response to optimizer-generated poses was generally positive – most subjects were very satisfied, while some opted to make minor changes after completion.

Dissatisfaction and criticisms were consistently associated with one of three issues: solutions to difficult optimization problems such as those which involved very large wrist movement, contact transfers to object locations with vastly different curvature than the source region, or generic UI layout and the lack of hotkeys. Subjects who understood optimization limitations deliberately tried to push the limits of the solver, while those who encountered some unwanted artifacts usually just modified the object transform slightly and re-solved. We also found that neither of our default axis generation methods were really successful – regardless of the default axis provided, subjects either preferred to create their own axis or opted to move forward in

spite of, rather than aided by, the generated default axis.

While the subject pool was small, these results do suggest that our tool is intuitive and predictable, and more importantly that those results held true across users with different existing skill, users with limited technical background, and in a variety of usage patterns. We found it especially encouraging that many subjects approached us directly after the final session and asked when the plugin would be available publicly, and even moreso that the inquires came from heavy Maya users. Figure 5.16 illustrates a number of results fully generated by subjects.

5.6 Qualitative Assessments

To assess the quality of user study results, we additionally conducted a round of qualitative assessments.

5.6.1 Subject Recruitment and Distribution

We published and distributed a survey online to a total of 65 individuals with a background or interest in animation via a combination of internal mailing lists, individual outreach, and class announcements. We also solicited the individual response of an expert animator with ten years of experience. Subjects were required to be over age 18. All subject information, with the exception of the expert animator, was kept anonymous.

5.6.2 Setup

We collected 12 poses generated by users from the previous study which used our plugin in a substantial capacity and 12 poses created using fully manual FK tooling. Poses were selected based on overall quality by an internal review process.

Each pose generated using the plugin was paired with another using only manual tooling primarily by user study subject ID, secondarily by item, and lastly at random with results from other subject IDs. Importantly, we note that the decision to primarily group by subject ID rather than object was to calibrate for individual skill level, both with respect to existing experience and the level of familiarity with our plugin vs. existing tooling. Since poses using the plugin were largely sourced from the third session (e.g. where subjects were allowed to choose their object), nearly all pairs present non-corresponding items. Our results therefore assess the general ability of an individual subject with and without access to our plugin, which we consider more meaningful than comparing results between subjects.

Survey subjects were presented with 12 pairs of images in A/B format spanning each of the following categories:

- Overall visual appeal
- Realism / physical plausibility
- Complexity
- Natural appearance



Figure 5.16: Subject results generated using our plugin in a substantial capacity.

Answer choice and category order was fully randomized to reduce ordering bias. Subjects were required to provide a response to all questions in order to receive credit for completion.

5.6.3 Assessment Results

Table 5.2 tabulates the results of the study, where each answer choice is awarded a score of +1 for a win, 0 for a loss, and 0.5 for a tie. Selections were required to have a vote differential of 4 or more to be considered the majority choice.

Table 5.2: Qualitative study results

Qualitative Study Results		
	Plugin Score	Manual Score
Overvall Visual Appeal	8	4
Physical Plausibility	6	6
Complexity	4	8
Natural Appearance	6.5	5.5

The expert opinion was relatively consistent with aggregate results. Specifically, 9/12 plugin poses were preferred for visual appeal, 8/12 for physical plausibility, 4/12 for complexity, and 10/12 for natural appearance. Overall, poses generated by the plugin performed as well, if not better, than *fully manual* posing in a majority of categories. Importantly, we note that the loss in complexity is expected since users were exposed to the plugin for the first time during the previous sessions. We note that poses such as those in Figure 5.7 are possible to generate, but require more experience. However, performance in the remaining categories clearly indicate that our tools have a substantial impact on pose quality.

5.7 Drawbacks and Limitations

Although our tools enabled the creation of numerous high quality solutions, they do contain a number of limitations. For example, patch transfer can produce unexpected results if there exist points p on the patch and q on the axis such that $d_M(p, q) \neq d_{M'}(p', q')$, i.e. the geodesic distance between p and q on M does not equal the distance between their corresponding points p', q' on M'; Figure 5.17a illustrates such a failure. However, the user can draw an alternate axis that yields the expected results (Figure 5.17b).

Our contact model is currently not well-suited for grasps that do not have sizeable areas of contact with the target object as well as grasps that span concavities or holes, such as the rim of a glass or the holes of a button. While possible to model the contact as a set of many small patches, patch transfer and manipulation will be more susceptible to error. Another possibility is to convex-ify the surface, but this approach sidesteps the problem. We also note that our

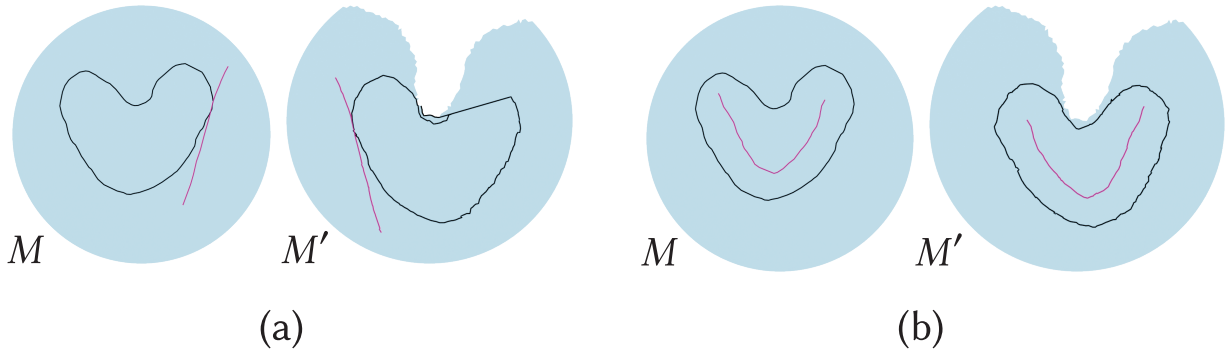

Figure 5.17: (a) Our transfer method can produce large unwanted distortion if the axis (magenta) is drawn such that the patch parameterization depends on geodesics expected to travel outside of a non-convex domain. (b) However, desired results can still be obtained by simply choosing the axis to lie entirely within the patch region.

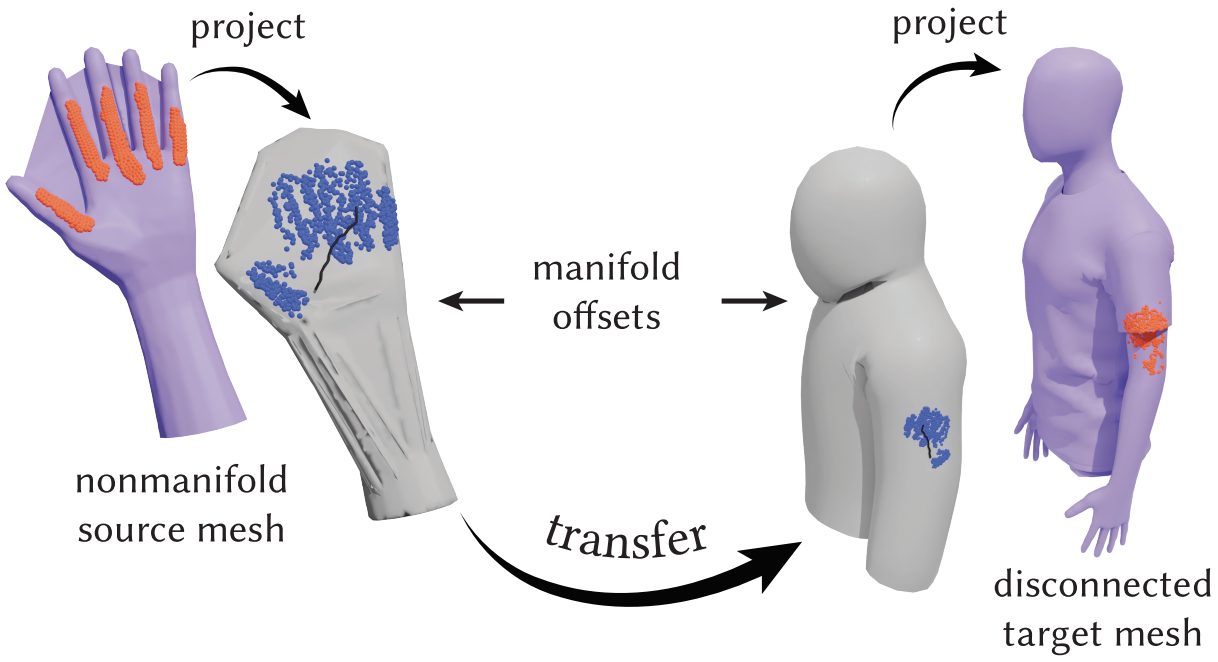

Figure 5.18: Although the geodesic tracing step of our transfer process is not directly applicable to nonmanifold and disconnected surfaces, contact transfer can still be achieved between such surfaces via the use of a manifold offset mesh. Parameterization and transfer occurs between the offset surfaces, and patches are transferred between the original and offset surfaces via projection. Offset surfaces in this figure were constructed using the "shrinkwrap" modifier in the open-source 3D software Blender.

optimization method, as illustrated by Figure 5.14(c), struggles when contact information is sparse, particularly in disambiguation of multiple solutions or resolution of uncanny artifacts.

Our method of contact parameterization and transfer is not designed for contact regions that belong to multiple disjoint surfaces, such as a figurine with separate layers of clothing. However, transferring such contacts is still possible under our framework. One option is to separately parameterize and transfer each portion of the contact region that belongs to a different surface. Alternatively, one can construct a single offset surface that encloses all components of either the source or target surface, project the contact region to/from the offset surface, and perform contact parameterization and transfer between the offset surfaces. On the other hand, transfer via a proxy surface often induces additional inaccuracy (see Figure 5.18). Robustly handling multi-component surfaces remains a challenge for future work.

Our solver produces locally optimal solutions only and struggles with finding solutions which require large amounts of wrist movement. User study subjects expressed little concern with this limitation in practice since modifying the object transform or moving the wrist or object into a more favorable start position required minimal effort; however, increased sensitivity to initialization renders our approach more problematic in automation-focused settings.

Other shortcomings involve the workflow of working with contacts. This issue was especially prevalent in non-human hands since the animator was expected to internally visualize the contact set without being able to replicate the pose themselves in the physical world. Maintaining finger spacing during transfer proved challenging as well, although hierarchical transfer partially combats the problem.

Two areas of future work are to reduce the time taken to draft contacts and extend our pose drafting capabilities to both full animations and VR / AR. Libraries of contact sets could assist with the former; however, full prediction is a notoriously challenging problem which requires a more extensive exploration into learning-based methods. The latter would likely require formulation of new interfaces, as our posing tools currently do not account for items such as interpolation, motion smoothness, or mid-air contact painting. We are excited by applications such extensions will enable.

5.8 Summary

We have presented EAD artist tools which enable intuitive modeling of high complexity, contact-rich manipulator-object interactions through the direct manipulation of contact areas. Our tools make extensive use of the AE contact model, which we demonstrate to be scalable to large contact regions, highly robust to the choice of axis and surface sampling variations, and capable of providing real-time, robust, and predictable updates in responses to edits including dragging, rotation, hierarchical composition, and transferring between surfaces. We have also introduced an optimization framework that is scalable to high DOF manipulators through two modifications of an existing formulation. Combined, our framework provides the first direct and responsive means of working with contact regions, which we demonstrate can lead to the synthesis of high quality results while allowing animators to retain full artistic control.



Chapter 6

Kinematic Motion Retargeting

Up until this point, we have only discussed contact areas in the context of grasping. Although we briefly touched on extending grasp generation to the drafting of simple manipulations, we have yet to examine motion in a substantial capacity. In this chapter, we examine using contact areas in conjunction with motion capture (mocap).

6.1 Motivation

Advancements in mocap technology have made it possible to collect motion data at high levels of detail, including large scale body movements and fine-grained hand movements together in a single capture [Romero et al. 2017; Taheri et al. 2020]; however, *using* this data is still difficult because the targeted embodiment for the data is almost never the same as the demonstrator. For hands in particular, retargeting – transforming the motion of one embodiment into another – is often required because of the large variety of different hand models used in practice and the precision required to make grasps, especially those with many points of contact, look compelling (e.g. Figure 6.1).

There are primarily two ways of performing the retargeting: proactively or retroactively. Proactive retargeting, also commonly known as tele-operation, involves mapping the motion — typically modeled as rotations of joints in a kinematic chain — of the source embodiment (typically a human) to the desired target embodiment before any data is collected. While proactive retargeting provides the benefit of automatically generating motion for the target embodiment, it requires significant upfront investment and is not reusable — if motion needs to be collected for a new embodiment, a brand new capture must be performed. Retroactive retargeting addresses these drawbacks by instead synthesizing the motion of the target embodiment after the data has been collected. By doing so, retroactive retargeting provides scalable data re-usability; however, it trades off collection overhead with downstream processing complications.

In the context of hand-object interactions, common post-processing artifacts include motion misalignment or lack of contact – both of which are difficult to repair even for a highly trained animator. Complications are further compounded when the target hand differs substantially from the demonstrator, whether that includes different finger proportions, degress of freedom

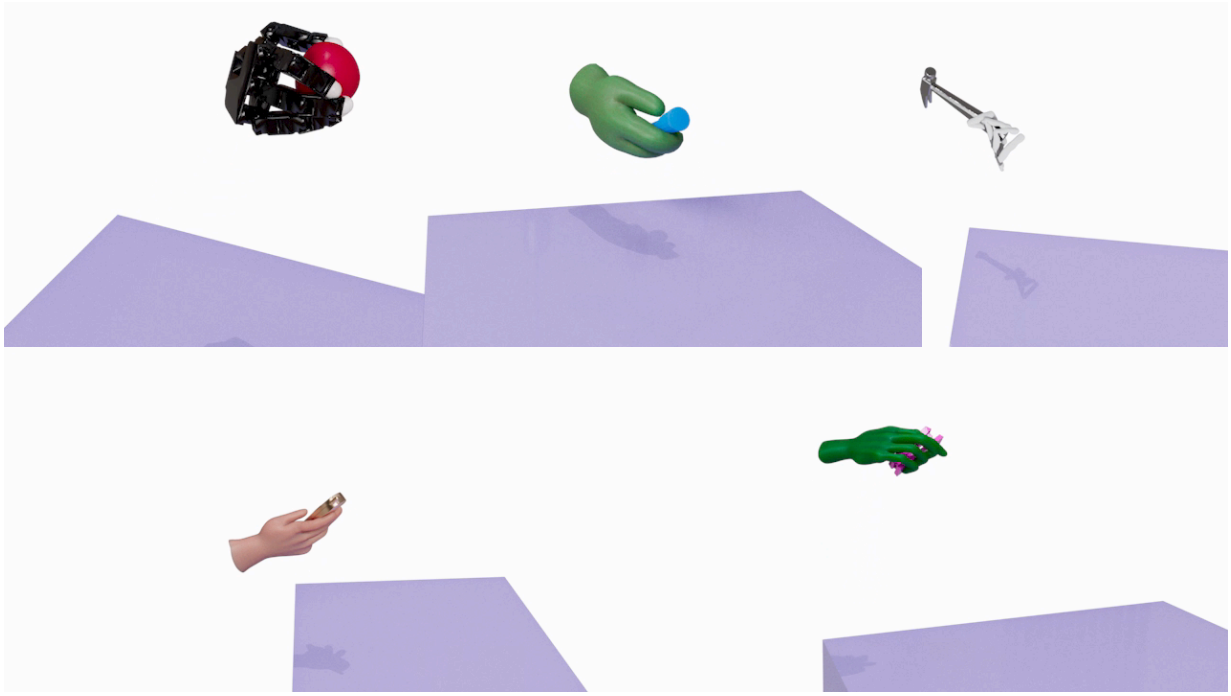

Figure 6.1: A collection of diverse hand models and some interesting motions synthesized for them.

(DOFs), or number of fingers.

The lack of reliable retargeting methods for retroactively retargeting contact-rich hand motion data remains a bottleneck for multiple communities. In film and game production, considerable animator time is spent customizing common manipulations to different character rigs. In robot learning, the difficulty in adapting existing manipulation data to different hands contributes to the data sparsity problem [Sivakumar et al. 2022], which in turn limits the feasibility of supervised and semi-supervised machine learning algorithms. A standardized approach for re-purposing existing data, and in particular data easily collectible from humans, would provide a much-needed solution for a broad range of applications. We propose such an approach through the exploitation contact areas.

6.2 Approach

Our retargeting pipeline is shown in Figure 6.2 and can be divided into the following steps:

1. Extract a dense corresponding set of contact areas between the object and source hand per frame
2. Procedurally transfer all contacts from the source to target hand across all frames
3. Estimate an initial trajectory for the target hand using fixed markers and the transferred contacts
4. Refine the estimates to improve temporal consistency

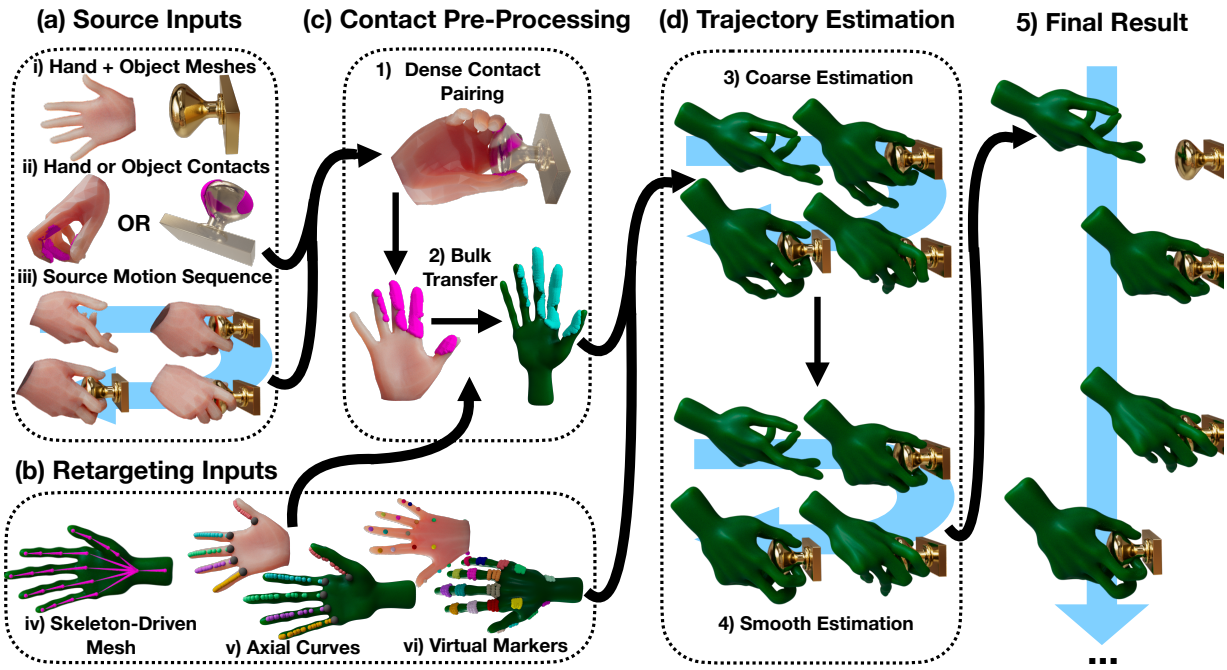

Figure 6.2: Overview of our retargeting framework. (a) Our approach requires inputs of accurate meshes of the original object and source hand, per-frame contacts on either the object or source hand, and a complete motion sequence of the source hand. (b) To perform the retarget, we require a skeleton-driven target hand mesh as well as a set of artist-annotated corresponding virtual markers and axial curves. (c) After recovering a dense set of contacts between the object and source hand, we transfer contacts across the entire time series and (d) use the virtual markers and transferred contacts to synthesize motion for the target hand from scratch.

5. Construct the final trajectory through spline fitting

Importantly, our pipeline assumes that the desired solution is one that *attempts to match the interaction mechanics of the source manipulation as exactly as possible*. For this reason, we assume contacts on the object are the same across all hands. We detail the expected inputs and processing steps in the following subsections.

6.2.1 Inputs

Our method requires existing hand-object motion data as input, which we expect to minimally include:

1. Accurate meshes of the original object and source hand
2. A set of dense per-frame contact annotations on either the object or hand mesh, which can be specified by any means (e.g. distance-based metrics)
3. A complete set of frames defining the motion sequence

For retargeting to a new hand, we require:

4. A skeleton-driven target hand mesh
5. A set of artist-annotated *corresponding virtual markers* on the source and target hand
6. A set of artist-annotated *corresponding axial curves* on the source and target hand

We expand on the required artist annotations and their usages in the following sections. Importantly, we do not assume any sort of morphological or geometric similarity between the source and target hand (e.g. identical finger counts, triangulations, finger or palm shape, finger length). We also do *not* require the source hand skeleton.

6.2.2 Dense Contact Pairing

Prior to retargeting, we assume a 1:1 correspondence between each hand-object contact point pair. Contact points are stored as barycentric coordinates to render our approach sampling agnostic, and thus robust, to variances in triangulations across different meshes. This approach ensures that data can be collected and transferred from a coarsely triangulated hand (e.g. the MANO hand [Romero et al. 2017]) to arbitrarily fine or irregularly sampled target hand meshes without risk of discretization error from vertex clamping.

In the event of only one dense set being available, we can generate a corresponding dense set through raycasting. We trace contacts out from source mesh locations along element normals until the opposite mesh is intersected, automatically generating a paired barycentric point. In the event of penetration, we invert the trace direction and retry. Points which do not intersect the opposite mesh or are further apart than an ϵ-metric can be considered errors and discarded.

6.2.3 Hand Shape Matching

In the previous chapter, we proposed a method of computing hand poses if we know the corresponding contact area distributions between the hand and object. Although we have obtained such correspondences by a procedural method in this context, we only know the distributions on the *source* hand. However, if the contact distributions on the object stay the same, then we can preserve the correspondences on the target hand by *transferring the contact distributions from the source hand to the target hand surface.*

But there are three complications when we consider full manipulation motions instead of only grasps. First, the motion can contain sub-sequences which may not contain any contact data whatsoever – for example, during phases when the hand is approaching the object or after the object has been released. Second, motions can be comprised of hundreds, or even thousands, of data frames. Third, depending on the method used, it is possible that the contact areas generated may not have labels at all. The second and third complications can therefore render the manual transfer of contact distributions between source and target hands by an artist impractical.

We propose addressing these problems by systematically shape matching the source and target hand shapes. Concretely, we require a one-time sparse set of artist annotations consisting of *virtual markers* and *axial curves.* Virtual markers help to define hand pose in situations where contact information is sparse or completely absent (problem 1), while axial curves enable scalable and customizable contact transfer across the entire time series without the need for labels (problems 2 and 3). We expand on both annotations below.

Virtual Marker Alignment We define virtual markers as an arbitrary collection of *fixed* corresponding sets of mesh points between the source and target hand. Markers can either be traditional single points or areas. In the former case, aligning virtual markers for pose computation reduces to traditional keypoint-based IK solving consistent with existing literature [Dasari et al. 2023; Qin et al. 2022; Sivakumar et al. 2022; Wei et al. 2023; Humberston & Pai 2015]. Area-based correspondences can be generated by applying our previously proposed contact transfer approach to produce an automatic mapping between such markers, which can be viewed as analogous to matching “contacts" in mid-air. Notably, however, an area-area configuration requires both sets to contain a 1:1 correspondence of discretized elements. Heterogenous mappings between single point and area based markers are also possible and trivial to designate as a one-to-many association under this modality. Figure 6.3 illustrates each configuration and instances where hetergeneous and area-based correspondences may be beneficial.

Contact Alignment Unlike markers, contact distributions are dynamic and can vary greatly between motion frames. We propose modeling bulk transfer as *non-isometric shape matching problem*, with the key insight to be formulating the correspondence as being governed by an atlas of multiple coordinate charts [Jost 2008]. Atlases can be comprised of one or many coordinate charts, provided the union of all charts generates a cover of the underlying manifold [Jost 2008].

Techniques targeting transfer of global media (e.g. textures), however, typically consider only single chart correspondences and require careful landmark placement to use effectively. Multiple

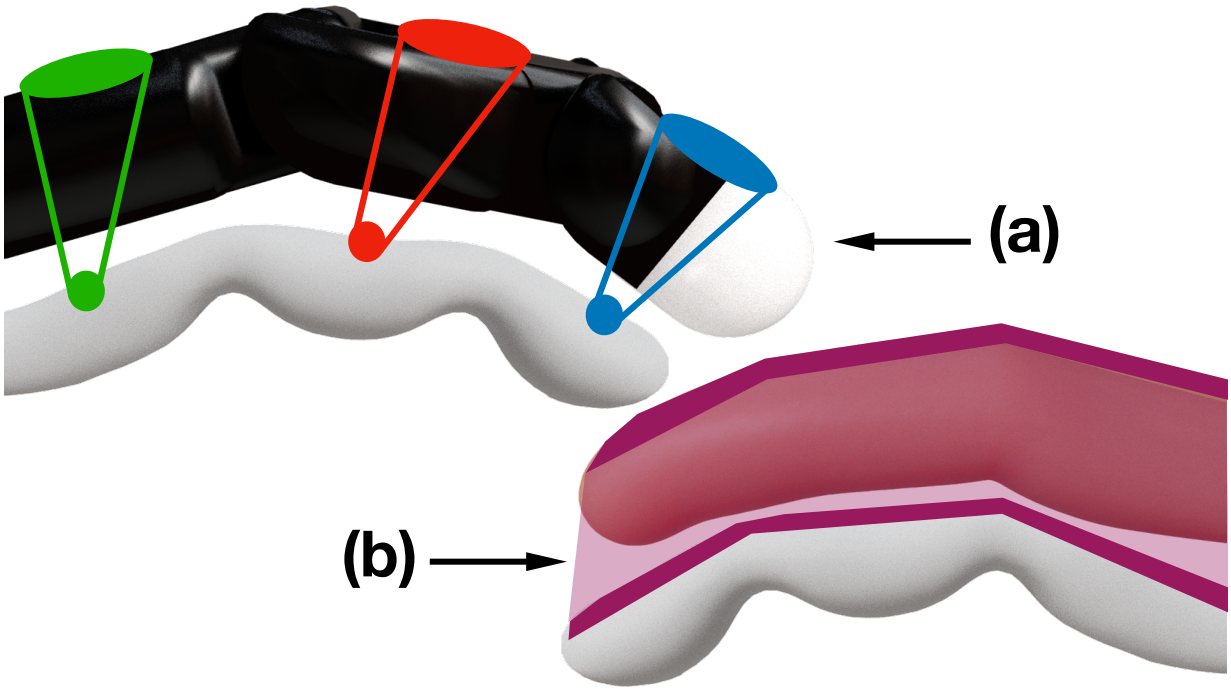

Figure 6.3: Virtual markers can be configured as traditional single-point one-to-one, or alternatively (a) heterogeneous many-to-one, or (b) dense, area-based configurations. Configuration (a) can be utilized to model uncertainty between virtual marker locations between differing source and target hands, which can be useful when finger link lengths are different sizes. Configuration (b) can be used to weight the importance of matching the deformed hand states over large regions, which can be useful when deformation behaviors diverge despite similar link lengths.

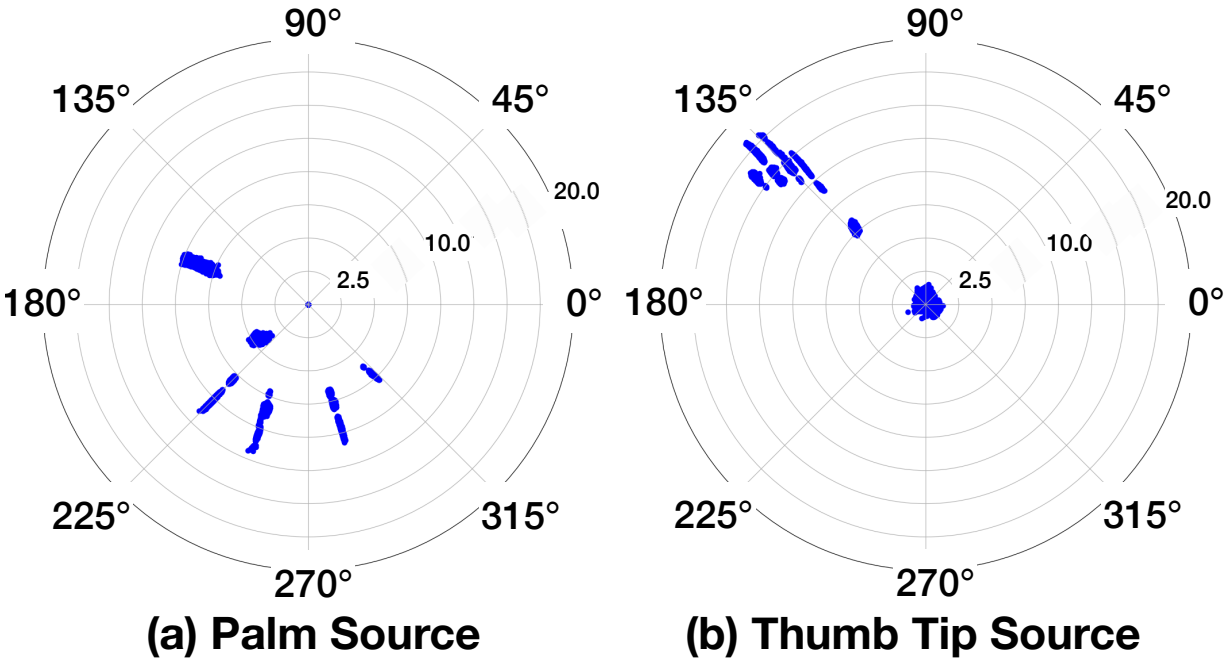

Figure 6.4: (a) When examining geodesic distance and outgoing tangent vector direction from a landmark point on the palm, clusters of contact points are clearly separable. (b) Viewing the same distribution from a landmark on the thumb tip cleanly isolates thumb contacts from those of remaining fingers, allowing us to easily introduce discontinuities to filter out the remaining contacts and subsequently perform arbitrary transformations exclusively on the isolated distribution.

chart correspondences are uncommon in such contexts because careful handling of interpolation across transition regions [Jost 2008] and hard chart boundaries is required to maintain global smoothness. Additionally, we found that the inherently high sensitivity to landmark placement makes it challenging for artists to accurately predict and intuitively understand incremental responses to such edits during annotation.

The highly local nature of contact areas, however, allows us to sidestep many of the drawbacks of mapping global features. For example, a chart used to parameterize contacts on the index finger need not have any influence on a chart used for the middle finger (qualitative semantic boundary), and a chart for contacts towards the tip of a finger need not influence a chart used for contacts towards the base (quantitative geodesic distance boundary). Quantitative boundaries are particularly useful because they can be reliably used even when semantic information is not provided. Figure 6.4 illustrates such an example, where clusters of contact areas between different fingers are clearly quantitatively distinguishable from each other when viewed from a sample reference point on the palm. It is therefore possible to isolate each finger region by introducing a hard discontinuity in the region of separation, allowing for subsequent fine-grained control over the parameters of each chart independent of the others. The atlas produced from the union of such disjoint charts is thus a segmentation of the manifold; however, unlike typical segmentations of unordered elements, contacts parameterized by any constituent chart can be fully reconstructed through chart inversion. We select the well-understood *logarithmic*

map (logmap) [Schmidt et al. 2006] as the template for each chart due to its low dimensional parameterization, easy inversion via the *exponential map* (expmap), and ability to be quickly and accurately computed via heat diffusion [Sharp et al. 2019]. The atlas of the source hand manifold Ω_S is thus formulated as:

$$\begin{aligned} \mathbb{A}_{\Omega_S} &:= \{(U_i, \psi_i) : i \in [1, M],\ \psi_i = log_q(c)\} \\ s.t. \quad & \bigcup_{i=1}^{M} U_i = \Omega_S \qquad U_i \cap U_j = \varnothing\ \forall\ i, j \in [1, M] \end{aligned} \tag{6.1}$$

where (U_i, ψ_i) represents each of the M constituent charts such that U_i is the subdomain of Ω_S governed by chart i and ψ_i is the logmap function that transforms contact c from barycentric U_i coordinates into logmap coordinates $(r_c, \theta_c)_q$ relative to origin q in the transformed space. We then postulate the atlas of target manifold Ω_T to be of the form:

$$\begin{aligned} \mathbb{A}_{\Omega_T} &:= \{(V_i, \psi_i^{-1}) : i \in [1, M]\} \\ s.t. \quad & \bigcup_{i=1}^{M} V_i = \Omega_T \end{aligned} \tag{6.2}$$

where $\psi_i^{-1} = exp_q c$, which can be computed by tracing a geodesic originating from q with parameters $(r_c, \theta_c)_q$. Including the disjoint condition is not necessary because contact-chart assignment is already determined on Ω_S. Note also that while segmentations of some contact distributions may not always be clear (i.e. full hand power-grasps), any partition of Ω_S is guaranteed to produce a contact point reconstruction on Ω_T. Figure 6.5 illustrates the aforementioned terms and proposed formulation.

Our formulation requires only the ability to compute Laplacians and geodesics, which notably are both intrinsic quantities. Unlike extrinsic methods that exploit properties related to a geometry's embedding in space (e.g. vertex locations, normals) [Li et al. 2007], intrinsic methods instead only consider the connectivity of the structure (e.g. triangle angles, edge lengths). The intrinsic perspective is highly advantageous in the case of hands because such metrics are isometry-invariant, which can allow us to move data between hand geometries at any point in time without knowledge of either the source or target hand pose.

We start by determining each region $U_i \in \Omega_S$ using a set of M landmarks extracted from one-time artist annotation. For convenience, we designate each landmark as the origin (q) of each logmap. We next determine the closest landmark to each contact point (q^*) using the Vector Heat Method [Sharp et al. 2019] and MMP [Mitchell et al. 1987] to extract the logmap coordinates of each contact point $(r_c, \theta_c)_{q^*}$. Importantly, automatically determining q^* means that we *do not require* contact data to be annotated or strictly associated with a pre-determined individual or grouped set of landmarks as mandated in Chapter 5 while also ensuring that each source chart's set of contact points is unique. Notably, while any partition is suitable under our formulation, our approach of taking the closest landmark to each contact under such an atlas is special in that it generates an equivalent point partition to that of a geodesic Voronoi segmentation [Herholz et al. 2017] with the added benefit of providing an exact method of reconstruction.

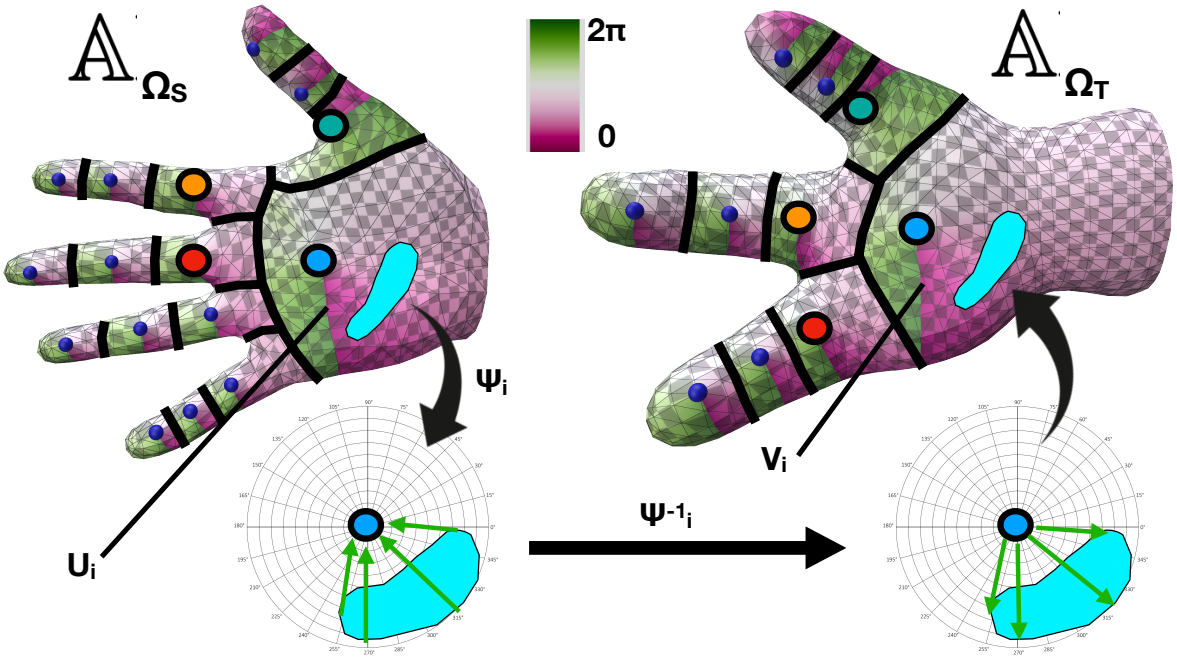

Figure 6.5: Cover generated from a set of landmarks (dark blue), chart boundaries (black), and example corresponding landmarks on each surface (red, orange, teal, light blue) obtained from annotation. A sample contact distribution (cyan) within the boundary of embedded chart U_i is parameterized against the chart's affiliated landmark using logmap transform ψ_i. Inverting ψ_i allows the distribution to be reconstructed from the corresponding landmark on $\mathbb{A}_{\Omega_T}$, while V_i determines the location of the contact's new embedding.

However, rather than requiring landmarks to be provided as individual points, we instead adopt a curve-based input approach [Gehre et al. 2018]. We select axial curves as the annotation implementation. Recall that zxial curves contain:

1. a finite set of points $\{a_1, \ldots, a_n\} \in \Omega$ which in our case serve as landmarks, with a *shortest geodesic* g_i connecting each pair of adjacent points (a_i, a_{i+1}) for $i = 1, \ldots, n-1$;

2. *turning angles* $\{\phi_i\}_{i=2}^{n-1}$, where each ϕ_i is the angle of rotation from the ending direction of g_{i-1} to the initial direction of g_i, expressed in the tangent space of a_i.

Axial curves also substantially reduce the need for careful landmark placement, automatically resolve the annotation overhead of consistently orienting the logmap zero angle of all constituent axis points, simplify the process of designating corresponding consistently oriented landmarks on Ω_T, and provides a representation that enables straightforward modification of expmap reconstruction parameters (discussed shortly).

In most typical cases, corresponding axial curves can be easily drawn by an artist; however, there are several interesting alternate cases we can discuss. In the case of Ω_T having fewer fingers than Ω_S, it is possible to safely "discard" unwanted contact groups by setting any $V_i = \varnothing$. Similarly, hands with additional fingers do not necessarily require all fingers to be used — the chart nearest the unused finger simply extends to the region, ensuring a full cover of Ω_T is still maintained.

To address variations in finger or palm shape or girth, we introduce an expmap scaling metric $\lambda_s(\theta)$ and reformulate ψ_i^{-1} as instead tracing geodesics of the form $(\lambda_s(\theta)r, \theta)$. This modification effectively permits "deforming" contact distributions parameterized on Ω_S into *any* shape on Ω_T, although in practice we found that uniform scaling was largely sufficient. Although such a parameter breaks the assurance of the expmap providing an isometry-preserving-as-possible reconstruction of contacts embedded by (U_i, ψ_i) on Ω_T, as illustrated in Figure 6.6, we note that such distortion is often desirable to meaningfully capture *semantic* similarity between geometric variations.

Finally, we address finger length variations by altering geodesic lengths of the axial curve representation to shift relative landmark distances, and by extension the desired locations of V_i. We therefore introduce one more parameter λ_a that can be used to extend or contract the length of the geodesics connecting the axial curve points on Ω_T. As also illustrated in Figure 6.6, the combination of λ_s and λ_a enables fine-grained, predictable adaptation of contact distributions across even widely varying geometries. Importantly, the flexibility to perform such alterations is made possible by the atlas model.

We can obtain a corresponding set of landmarks on both Ω_S and Ω_T, as well as hyperparameters λ_a and λ_s from one-time artist annotation, and propose building on our previously proposed artist tools to facilitate the process. Contacts across entire manipulation time series can then be subsequently procedurally transferred between the source and target hand.

6.2.4 Motion Retargeting

We now have all information necessary to synthesize the retargeted motion. We perform the reconstruction in three steps: (1) estimation of an initial trajectory for the target hand, (2)

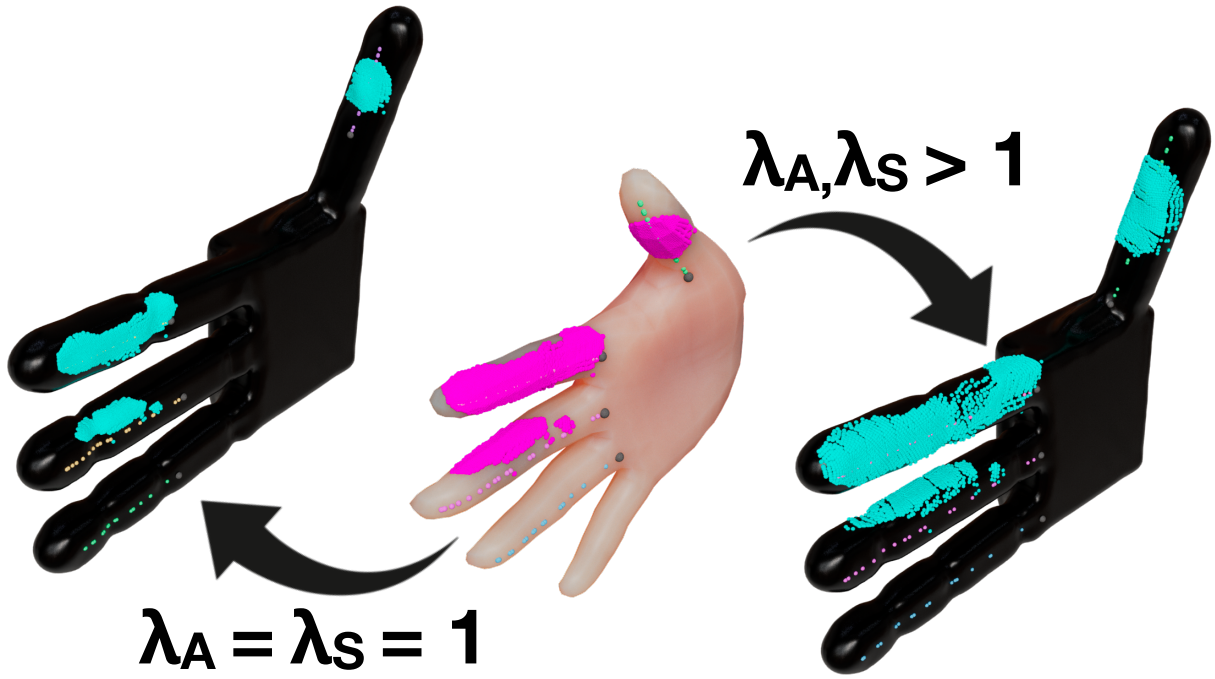

Figure 6.6: Illustration of isometry-preserving-as-possible contact transfer between two widely varying hand shapes. While unnatural squishing of contacts on the target domain initially fails to model semantic equivalence between fingers, these artifacts can be mitigated by adjusting λ_s and λ_a

refinement of the initial estimate, and (3) construction of the final trajectory through spline fitting. We use a simple and common objective formulation in all steps for computing solutions per frame:

$$\begin{aligned} \boldsymbol{\theta_f}^* = \arg\min_{\boldsymbol{\theta}} \quad & \lambda_m \Gamma_M + \lambda_c \Gamma_C + + \lambda_t \Gamma_T + \lambda_j \Gamma_J \\ \text{s.t.} \quad & \boldsymbol{\theta_L} \leq \boldsymbol{\theta} \leq \boldsymbol{\theta_U} \end{aligned} \tag{6.3}$$

where Γ_M, Γ_C, Γ_J, and Γ_T are the penalty terms, $\boldsymbol{\theta}$ is the DOF vector, $\boldsymbol{\theta_L}$ and $\boldsymbol{\theta_U}$ define the lower and upper bounds of $\boldsymbol{\theta}$ respectively, and λ_c, λ_m, λ_j, and λ_t are weighting hyperparameters.

We next describe each penalty term. We start with the marker penalty (Γ_M), which encourages aligning source hand and target hand virtual markers at frame f. Assuming M total virtual marker points, the marker penalty is defined as:

$$\Gamma_M = \sum_{m=0}^{M} \Gamma_{MD,m} \tag{6.4}$$

where $\Gamma_{MD,m}$ represents the L_2 distances between corresponding marker points. Next is the contact penalty (Γ_C), which encourages aligning target hand and object contacts at frame f. Assuming C total contact points, the contact penalty is defined as:

$$\Gamma_C = \sum_{c=0}^{C} (\lambda_{cd}\Gamma_{CD,c} + \lambda_{cn}\Gamma_{CN,c}) \tag{6.5}$$

where $\Gamma_{CD,c}$ represents the L_2 distances between corresponding contact points, $\Gamma_{CN,c}$ penalizes deviation from surface normal inversion at the contact points, and λ_{cd} and λ_{cn} are weighting hyperparameters. We also include a table penalty (Γ_T) that discourages hand-table intersection if a table is present. Assuming S sampling points on the target hand, the table penalty is defined as:

$$\Gamma_T = \sum_{s=0}^{S} max(0, -\Gamma_{SD,s}) \tag{6.6}$$

where $\Gamma_{SD,s}$ represents the signed distance function (SDF) of the table evaluated at the location of point s. For simplicity, we use the vertex set of the target hand or its affiliated manifold wrapper as S. We also assume a box geometry for the table, which allows us to compute the SDF analytically[1]. Lastly, we introduce the "prior" penalty (Γ_J), which serves as a regularizer against either the default rest pose or the previously existing keyed value at frame f. Assuming J DOFs, we obtain:

$$\Gamma_J = \sum_{j}^{J} \Gamma_{P,j} \tag{6.7}$$

where $\Gamma_{P,j}$ represents the deviation between DOF j and its existing value.

We propose performing the motion reconstruction in three stages: initial trajectory estimation, smooth refinement of the initial estimated trajectory, and fitting B-Splines to the smoothed trajectory estimates to recover the final motion. In the following subsections, "solving" and "re-solving" both refer to computing the optimal solution to Eq. 9.1.

Initial Trajectory Estimation Since no baseline data is available for the target hand motion, and we cannot easily obtain a preliminary trajectory through direct joint retargeting, we must synthesize it from scratch. We do so by first estimating a per-frame optimal initial trajectory estimate over F total motion frames, which we define as:

$$\Theta^* = \{\theta_0{}^*, \theta_1{}^*, ..., \theta_F{}^*\} \tag{6.8}$$

Importantly, the above formulation entails that each frame of the optimal trajectory *is independent of the estimates of its neighbors,* which runs counter to many existing works that add explicit conditioning terms on the estimate of the previous pose [Handa et al. 2019; Qin et al. 2021; Sivakumar et al. 2022]. This crucial distinction is important for mitigating error buildup over long sequences, which can otherwise cause the estimates of frames later in the motion to converge to highly undesirable local minima. Independent estimation also allows us to reliably prune poor locally optimal estimates during later processing stages.

[1] https://iquilezles.org/articles/distfunctions/

We propose performing the trajectory estimation in two passes. First, we solve for only root joint position per frame while keeping the remaining joints fixed in their default positions. We can seed the current frame estimate from the optimal solution of the previous frame exclusively at this stage for convenience, and then solve for the full pose per frame in the second pass using the root estimate as the seed.

Trajectory Refinement We next propose refining the estimated trajectory to improve temporal consistency using finite acceleration as the smoothing metric. To do so, we can impose a threshold $\mathcal{E}_{acc}$ and remove all frames that violate the threshold. We then replace each violated frame with a linear interpolation between its nearest valid left and right neighbors and re-solve. Frames can then be removed on a per-DOF basis and in a single pass to ensure independence of computation when re-solved.

However, updating violations can introduce new violations in the resulting trajectory. But we can perform the refinement procedure iteratively until either until no further violations are found or all iterations are exhausted. All unresolved violations at the end of iteration exhaustion can then be ignored during fitting.

Spline Fitting Finally, we can use the refined frame solutions as sample points and fit a cubic B-spline to each DOF across the computed time series, ensuring our final solution guarantees C^2 continuity. We can select a fixed number of control points and solve simultaneously for both values and locations in time using least squares pseudo-inverse approximation [Eberly 2005]. This representation also permits artist control over the fitted spline, and better approximations, at the cost of possible overfitting, can be easily obtained either by increasing the number of control points or hierarchically compositing splines to reduce error residuals [Lee & Shin 1999].

6.3 Experiments and Results

We validate our framework through demonstrations across five different hand shapes and six motions of different objects. We select the GRAB [Taheri et al. 2020] dataset as the data source for our experiments, which notably contains all input data required by our method.

Figure 6.7 illustrates all hands used in our experiments, which were sourced as-is from in-house artist rigs intended for media production and robot manipulators intended for simulation. The hands span a wide range of kinematic, morphological, and geometric diversity.

Figure 6.8 illustrates the heterogeneous virtual marker sets used for all results, in which the source hand was allocated a set of single-point markers and the target hand an area-based set. The annotation procedure is manual but was straightforward for an artist and only has to be performed once per source and target hand pair. We use all received annotations as-is.

We also obtain a corresponding set of landmarks on both the MANO source hand and all target hands, as well as hyperparameters λ_a and λ_s, again from one-time artist annotation. We use the artist tools outlined in Chapter 5 to facilitate the process. Contacts across entire manipulation time series are subsequently procedurally transferred between the source and target hand. Figure 6.9 illustrates the received annotations and a representative sample transfer, while Figure 6.10 illustrates a sample generated result of both the retargeted contacts and

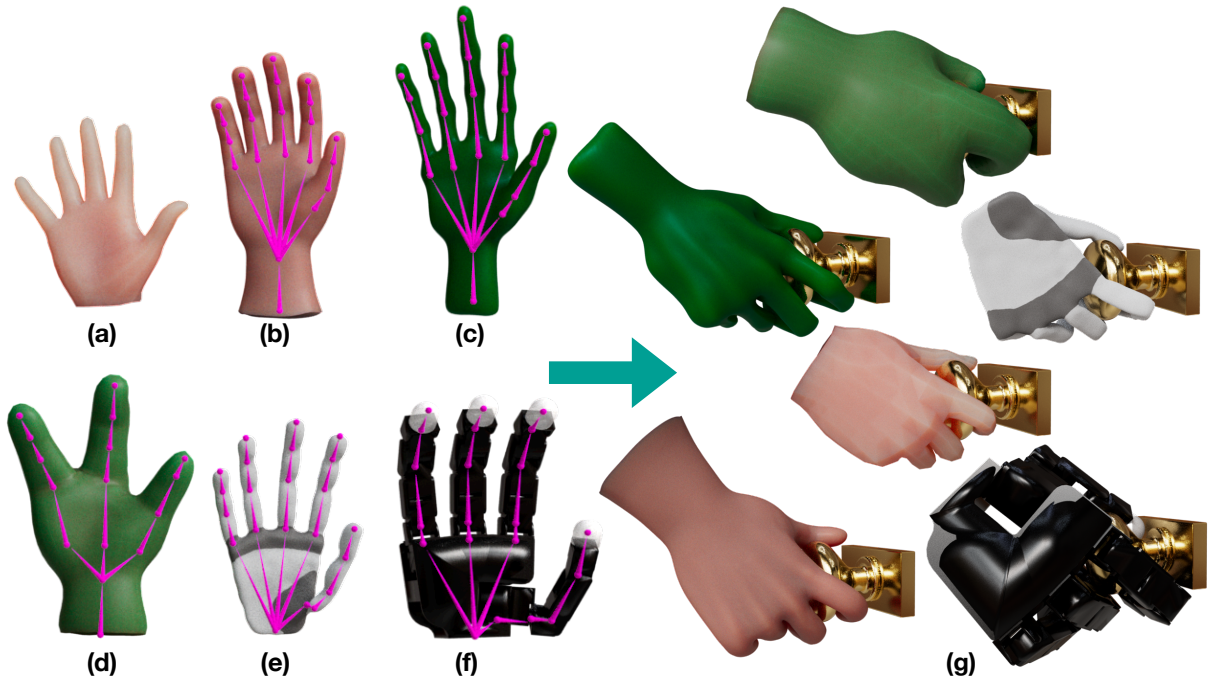

Figure 6.7: All hands used in our experiments, including the (a) source MANO hand, (b) an alternate human hand, (c) a witch hand, (d) an alien hand, (e) a custom prosthetic hand, and (f) the Allegro Hand. (g) We retarget demonstrations performed by the source hand to all these hands by procedurally transferring contact areas over the entire time series via shape matching.

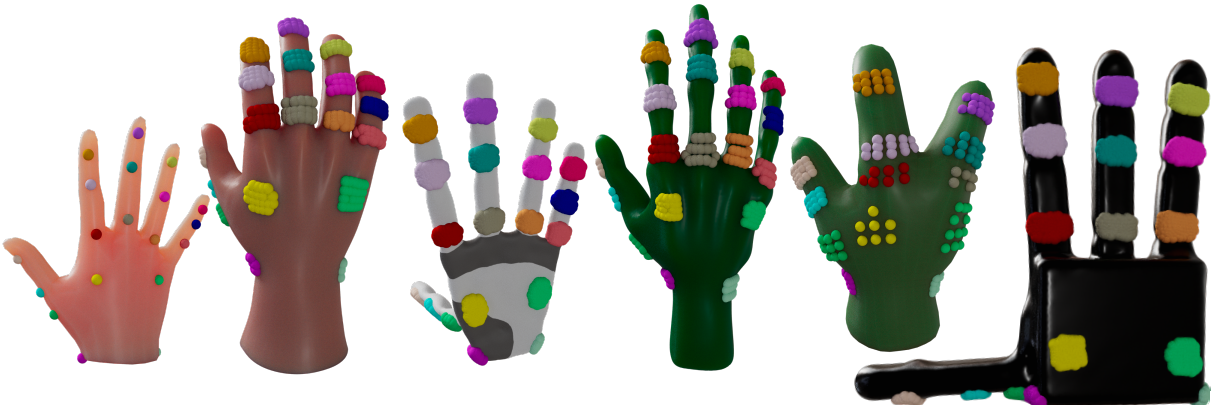

Figure 6.8: Single point virtual marker configuration on the source MANO hand and area based corresponding marker sets on all other hands. We use a manifold wrapper of the Allegro Hand for contact processing.

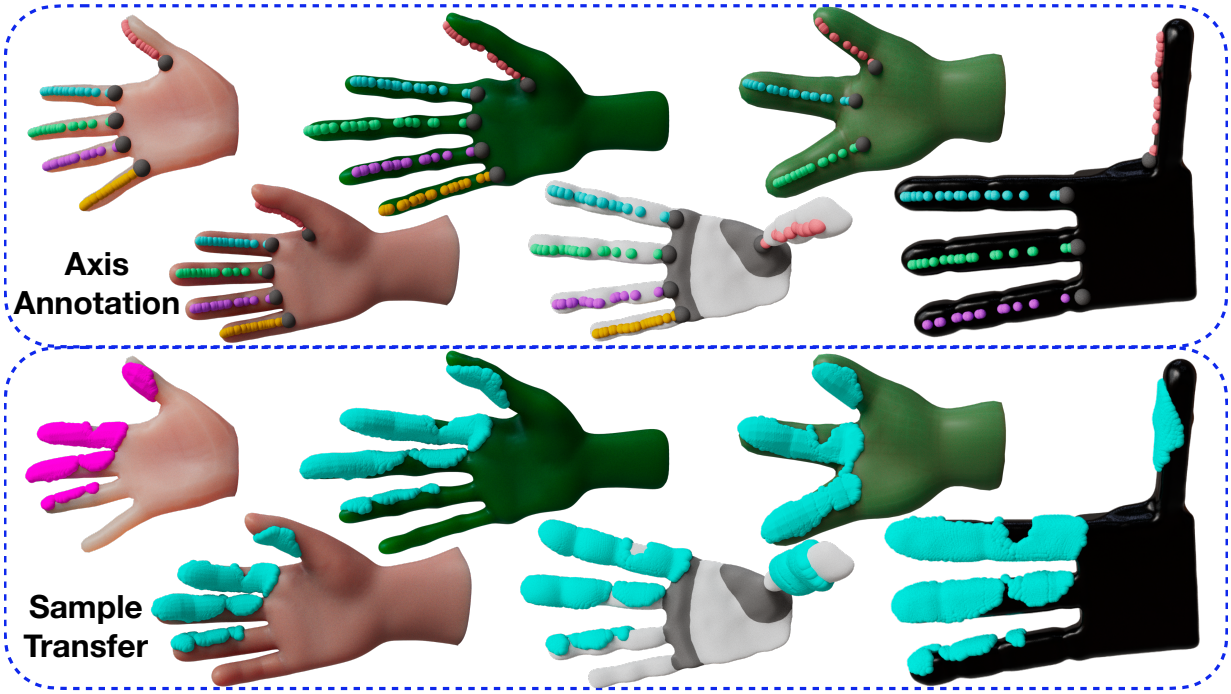

Figure 6.9: One-time axis annotation (top) enables scalable retargeting of original contacts (magenta) to customized configurations per target hand (cyan). Axis colors indicate matching annotations.

corresponding motion.

We use a uniform set of optimization hyperparameters across all hands and motions, the same transfer coefficients for each hand across all motions, and an identical number of control points per motion across all hands and DOFs. No parameter required careful tuning. We also did not modify any of the generated DOF spline motion curves in any of our examples.

Figure 6.11 quantifies the total amount of hand-object, self, and table intersection generated by our method across all results. To compute these quantities, we first determine all penetrating hand vertices via raycasting as detailed in Section 6.2.2. The resulting points are then clustered via depth-first search on the hand mesh. Finally, we compute the convex hull of each cluster and extract its volume. Total intersection is defined as the sum of all such hull volumes relative to the total volume of the hand, expressed as a percentage. All reported percentages are overestimates due to convex hull approximation. Despite not performing any hand-object or self-intersection resolution, we observe that intersection volumes are nonetheless low. Our results indicate that contact areas, in part due to their implicit encoding of natural grasp states are viable as a cheap approximation of both physical motion and intersection minimization in the absence of a full physics simulation.

Overall, we found the retargeted trajectories to be of generally high quality despite blanket standardization. Such generalization is particularly notable because results confirm that parameterization of contacts across arbitrary motions is possible *without knowledge of distributions in advance* and without customization of landmarks to individual distributions.



Figure 6.10: (Top Row) Object contacts, (Second Row) source hand contacts, (Third Row) computed target hand contacts, and (Bottom Row) source and retargeted hand motion for four different stages of a complex phone manipulation: (First Column) table pickup, (Second Column) in-hand dialing, (Third Column) holding for use, and (Last Column) movement back towards the table for release. Although poses and contact distributions vary dramatically during the manipulation, our method can successfully produce target hand motion by using source hand contact areas as a foundational retargeting medium.

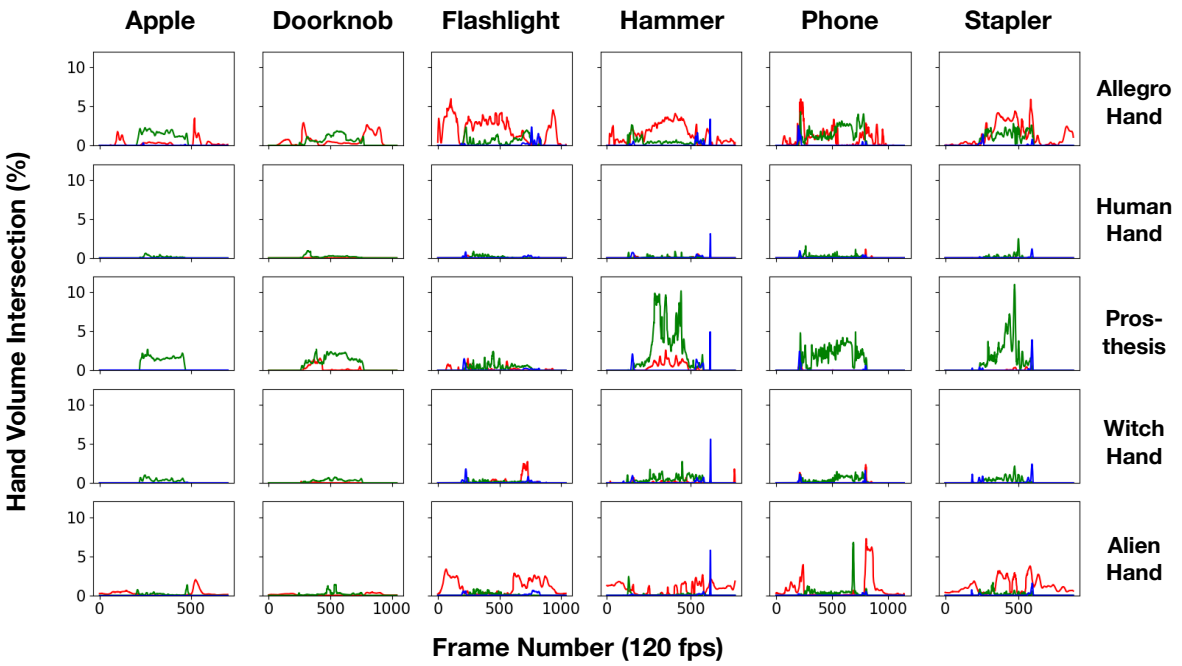

Figure 6.11: Plots of all hand-object (green), self (red), and table (blue) intersection.

6.3.1 Extension: Object Substitution

Our method also enables retargeting generic manipulations to different objects. Contact sets can be generated on new objects through any means of correspondance, including by reversing the raytrace process from the source hand, tracing from the original object, through diffusion models [Wei et al. 2023], or, perhaps most interestingly, by using the same atlas-based approach used for blanket retargeting of hand contacts. The remainder of the pipeline is subsequently applied normally to produce the final result. We illustrate an example in both Figure 6.12 and Figure 6.13 and in the supplementary video of re-purposing an existing apple “handoff" manipulation to a potato, where contacts on the potato are generated by tracing the original contacts outward from the apple, and a more geometrically complex gargoyle sourced from the Thingi10K dataset [Zhou & Jacobson 2016], where contacts are instead retargeted to slightly different configurations using an atlas between the apple and the gargoyle.

6.3.2 Comparisons

We next validate the importance of contact information by comparing our approach against two existing contact-free methods used in tele-operation: fingertip keypoint tracking [Qin et al. 2022; Dasari et al. 2023] and functional pose equivalence [Handa et al. 2019]. We use the Allegro Hand for consistency with the baselines. To isolate the objective formulation, we beneficially augment both methods with information about the known full source trajectory. Specifically, we provide access to ground truth root joint estimation, independent pose computation per frame (Section 6.2.4), and retroactive smoothing of the whole sequence (Sections 6.2.4-6.2.4).

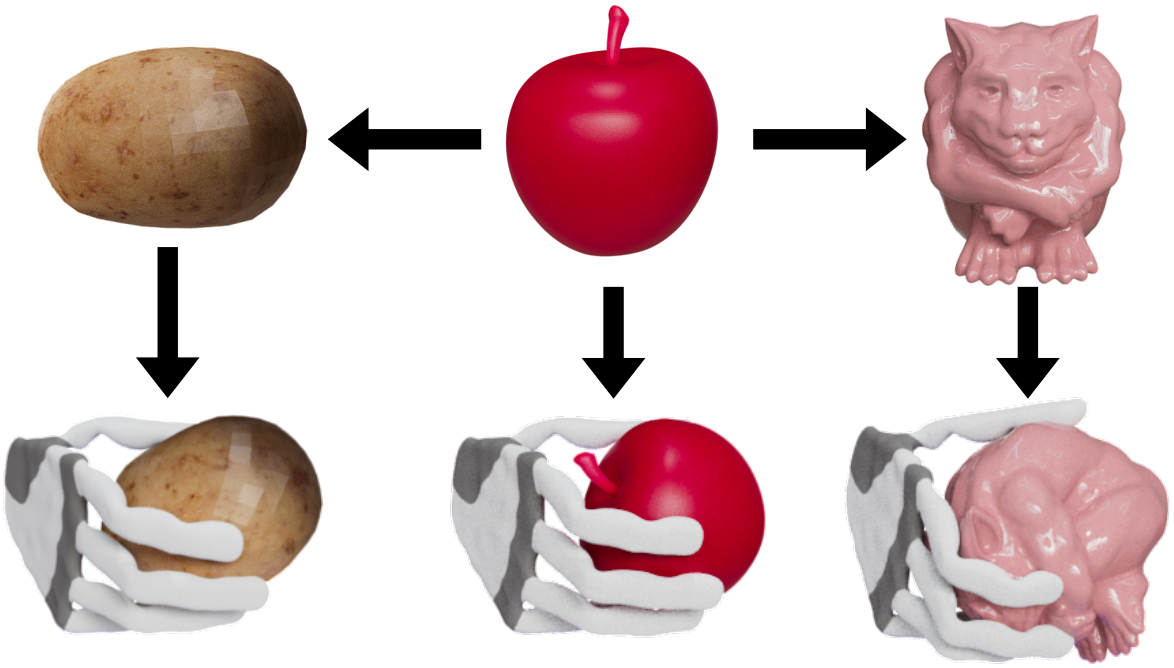

Figure 6.12: Our method can be extended to accommodate retargeting generic motions between different objects. Hand grasps successfully make subtle, but important adjustments to adapt to simple shapes (left) as well as more dramatic adjustments for more complex features (right).

Fingertip Keypoint Tracking We first consider the popular strategy of fingertip keypoint tracking. We select four keypoints at the tips of each Allegro finger and assign them to track the human thumb, index, middle, and ring fingertip positions. The corresponding Allegro hand pose is computed using an existing optimization formulation [Qin et al. 2022], which is equivalent to Eq. 9.1 and 6.8 with $\lambda_C = 0$.

Functional Pose Equivalence We next consider the strategy of generating an equivalent functional pose on the target hand, which instead considers relative distances between pairs of keypoints on each hand and measures equivalence in terms of task space keyvectors [Handa et al. 2019; Sivakumar et al. 2022]. Specifically, we use the objective formulation proposed by DexPilot [Handa et al. 2019] as the basis for pose computation. Exact keyvectors and detailed term explanations are available in the cited work.

As illustrated Figure 6.14, we observe that both methods, while capable of producing smooth trajectory estimates, generally struggle to make contact with the object and produce non-trivial motion misalignment artifacts. Because both of these techniques have been used for retargeting in teleoperation scenarios, a person in the loop can interactively correct for errors such as those shown in Figure 6.14. In contrast, the approach presented in this paper provides results with good contact without the need for such intervention.

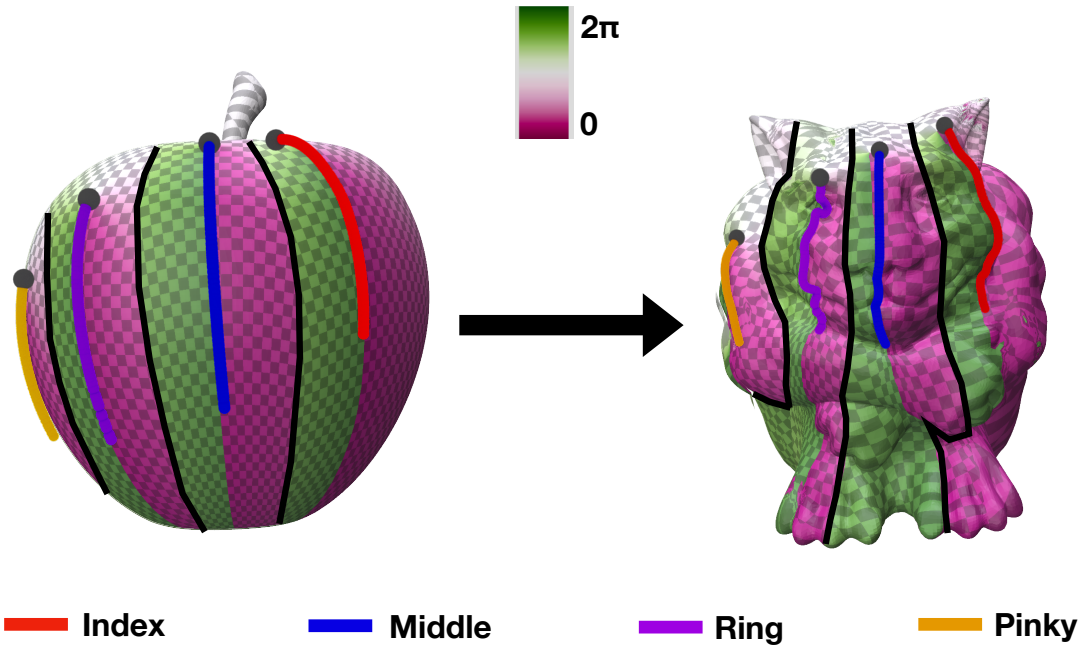

Figure 6.13: Approximate atlases used to retarget contacts of a handoff manipulation between a simple apple and a more geometrically complex gargoyle figurine. Axial curves corresponding to each finger are denoted by different curve colors, while black spheres denote the start point of each curve. Curve placements on the target surface can be arbitrary and be used to implicitly modify grips on the target surface to any extent desired.

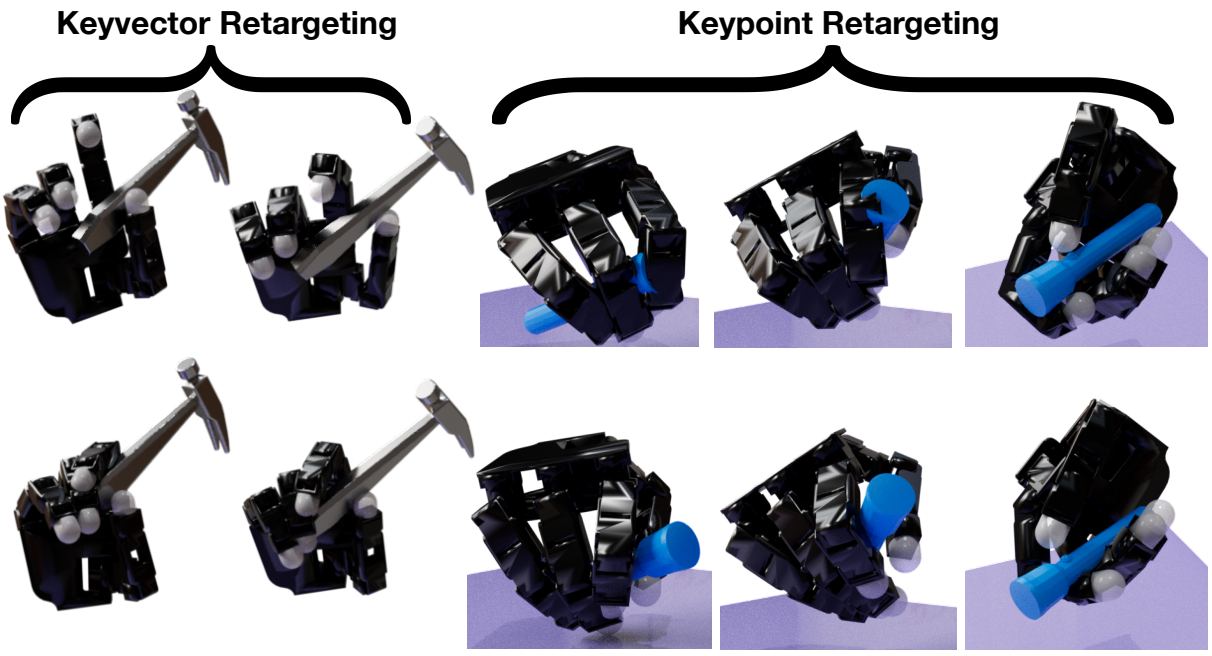

Figure 6.14: (Top row) Functional pose equivalence in terms of keypoints or keyvectors visibly struggle encouraging hand-object contact and motion alignment especially when source and target hand geometries greatly diverge. (Bottom row) Using contact areas greatly reduces such artifacts.

6.4 Discussion

We were pleasantly surprised to find that high quality retargets were possible even with a uniform set of parameters across all hands, although certain hands were comparatively easier to retarget than others. Surprisingly, the three fingered alien hand was the easiest to work with despite its unusual morphology. Its comparatively thick size was highly beneficial for limiting pose estimate uncertainty with respect to both markers and contacts. The only noticeable drawback was higher relative self-intersection compared to other hands; however, this behavior was unsurprising since the large majority of these intersections were caused by overlap between the second and third phalange, which were mapped to to the index and middle finger of the human hand — two fingers which commonly moved together in close proximity within the task suite we examined. In contrast, despite its more anthropomorphic appearance, estimations of the prosthesis wrist position were comparatively far noisier due to its thin profile. The prosthesis also exhibited relatively high hand-object intersection for the hammer and stapler tasks, which we found could be attributed to its small size and limited DOFs. In the case of the stapler, the prosthesis had difficulty curling fingers cleanly around the entire object. In the case of the hammer, as illustrated in Figure 6.15, the lack of a torsional MCP DOF caused the index finger to penetrate through the hammer rather than wrapping around the handle. The witch and Allegro hand were the most difficult to retarget due to their unusually long finger lengths, which we argue is one of the most significant factors in determining retargeting complexity between hands.

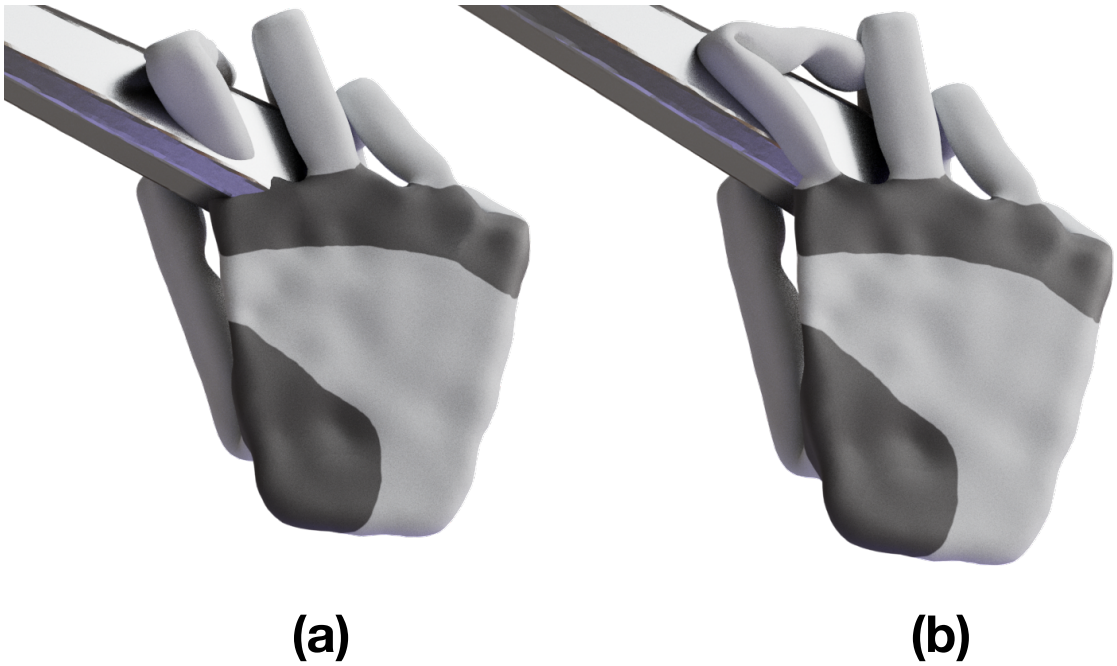

Figure 6.15: When optimized from the same start position with the same marker and contact distribution targets, (a) a 2-DOF MCP joint converges to a solution which penetrates through the hammer while (b) a 3-DOF MCP joint is able to conform to the hammer's handle with much less intersection. We found that the torsional DOF was utilized in this particular grip across all hands that included a 3-DOF MCP joint.

Additionally, although we focus on hands in this paper, we showed that our shape matching approach is general and can be used to precisely define transfer of arbitrary local data between shapes. Additional applications include defining complex interactions between surfaces (e.g. between two deformables), adaptable manipulation planning, or movement of entire assets positioned on one surface to another for surface animation retargeting. We are interested in exploring additional extensions of our approach to broader contexts in the future.

Finally, we note that we did not receive any hand pairs following an area-area virtual marker configuration; however, because our goal in this work was to examine a wide variety of dissimilar hands, we suspect that the annotator's decision to use only many-to-one configurations is largely a result of experiment specifics. Even the retargeted human hand we used had different finger length sizes than the source. Retargeting between human hands of similar proportion or highly anthropomorphic robot hands (e.g. the Shadow Hand[2]) may provide a more compelling use case for area-area configurations.

6.5 Drawbacks and Limitations

While effective in retargeting motions across a wide range of hands, objects, and manipulations, our method contains several limitations.

The atlas generated by our shape matching approach is notably discontinuous and non-differentiable, making it unsuitable for gradient computations or transfer of global media such as smooth functions or textures. Although these properties were not critical to our problem scope, formulating a C^∞ atlas would potentially enable a broader range of useful applications.

Our method does not guarantee a retargeted solution free of artifacts. The most common artifacts were wobbles resulting from leftover trajectory estimation noise or under / over fitting splines. However, both of these artifacts can generally be resolved by tweaking the acceleration cutoff for trajectory refinement or adjusting the number of control points. We found that a fixed set of parameters was suitable for most motions, but note that further refinement is possible with per-hand-per-motion customization if desired.

Our method sometimes has difficulty resolving table intersections in a manner that appears natural, which can result in uncanny finger contortions during object pickup and release. This issue arose in the flashlight and stapler manipulations by the witch hand, and the problem is generally more common in unconstrained long-fingered hands. These observations suggest that long-fingered may sometimes exhibit considerably different interaction mechanics than standard human hands during table pickup tasks, and that additional studies of such interactions may be beneficial.

Our motion synthesis pipeline currently treats contact distributions per timestep independently and therefore limits the ways in which we can utilize contact information. For example, if temporal changes between specific contact distributions (e.g. an index finger contact area) were available, we could use that information to edit the manipulation mid-trajectory in ways that the original distributions would not permit. This would be useful for correcting artifacts generated by ambiguous shape mappings such as between the human and alien hand. Alternatively, knowing how distributions change could allow us to analyze and determine what mode [Cheng

[2]https://www.shadowrobot.com/dexterous-hand-series/

et al. 2022] a particular distribution is operating in (e.g. sliding, rolling, etc.) and translate that information into a control policy for a robot.

Finally, our formulation of matching object contacts from the source hand is generally less effective for hands with large morphological or kinematic divergence from the source hand. The Allegro hand is a prime example of a challenging retargeting problem because of its size and limited DOFs. The main difficulty caused by the large size of this manipulator was greater self-intersection, the likelihood of which is generally higher in larger hands. In such instances, and in others where the target hand diverges enough to substantially alter the general expected interaction mechanics (e.g. a human vs. a Barrett Hand[3]) or with respect to a particular manipulation (e.g. a pinch grasp on a larger hand vs. a human power grasp), matching original human contacts exactly may not be the best problem formulation. While we do not assume kinematic or morphological similatirty between source and target hands for our retargeting process and demonstrated that our method can still handle considerable divergences, assuming identical object contacts between hands can be viewed as an implicit limiting factor of our approach. Addressing such divergences would be an interesting area for future work.

6.6 Summary

We have presented a simple, reliable, and standardized framework capable of kinematically retargeting contact-rich manipulations to a wide variety of target hands. Central to our method is the utilization of contact areas, for which we have presented both a novel, atlas-based shape matching algorithm capable of transferring localized point data procedurally with high control and precision, and an optimization pipeline capable of utilizing said information to create high quality retargeted results. We have shown that our method is capable of enabling unique extensions, including object substitution and visualizing the impact of design parameters. Finally, we have shown the value of contact information and key aspects of our processing pipeline through baseline comparisons. A natural next step would be to extend our currently purely kinematic retargeting framework to incorporate contact dynamics and physical constraints, which we will explore in the next chapter.



[3] https://robots.ros.org/barrett-hand/

Chapter 7

Online Adaptive Control

So far, we have proposed ideas for the generation of grasps and high quality manipulation retargets only considering kinematics. While kinematics is a natural starting point, there is absolutely no promise that any synthesized result will actually work when deployed on a physically simulated, let alone real, system. In this chapter, we explore techniques for addressing this drawback with the intention of building contact-driven controllers for dexterous robot hands.

7.1 Motivation

Many controllers for dexterous manipulation tasks recently have been constructed using imitation or reinforcement learning. Although the demonstrated results are impressive, many of these works require extensive offline computation, exploitation of task-specific assumptions (e.g. up-facing palm, primarily power grasps, constant number of maintained contacts, primitive objects, etc.), or significant upfront engineering costs (e.g. tele-operation, large scale data collection).

But what if we could build controllers cheaply from a single arbitrarily sourced manipulation demonstration conducted with a possibly different hand? In particular, demonstrations for human hand manipulations are already publicly available in high quality motion capture datasets [Taheri et al. 2020; Fan et al. 2023] and lower quality video datasets [Grauman et al. 2022; Damen et al. 2022]; however, adapting such demonstrations to robot hands is a well-known challenge due to the retargeting gap. A common strategy is to first kinematically retarget the human trajectory to the robot hand and then solve for the control policy with the retargeted trajectory as a baseline using imitation [Qin et al. 2022; 2021] or reinforcement [Rajeswaran et al. 2018; Dasari et al. 2023] learning. Generating a controller using these strategies, however, requires considerable compute time and resources. The resulting motion can also sometimes yield unexpected behaviors when rolled out online (e.g. object throwing), which can be challenging to foresee and triage due to the slow development cycle and black-box nature of the learned policy. We instead propose a sampling-based *online* model-predictive control (MPC) approach, which we title ContactMPC, that explicitly formulates the control policy as the solution to an auto-regressive optimal control problem. In particular, our MPC approach takes advantage of the kinematic retargeting pipeline proposed in the previous chapter as well as the retargeted contact area distributions during online evaluation.

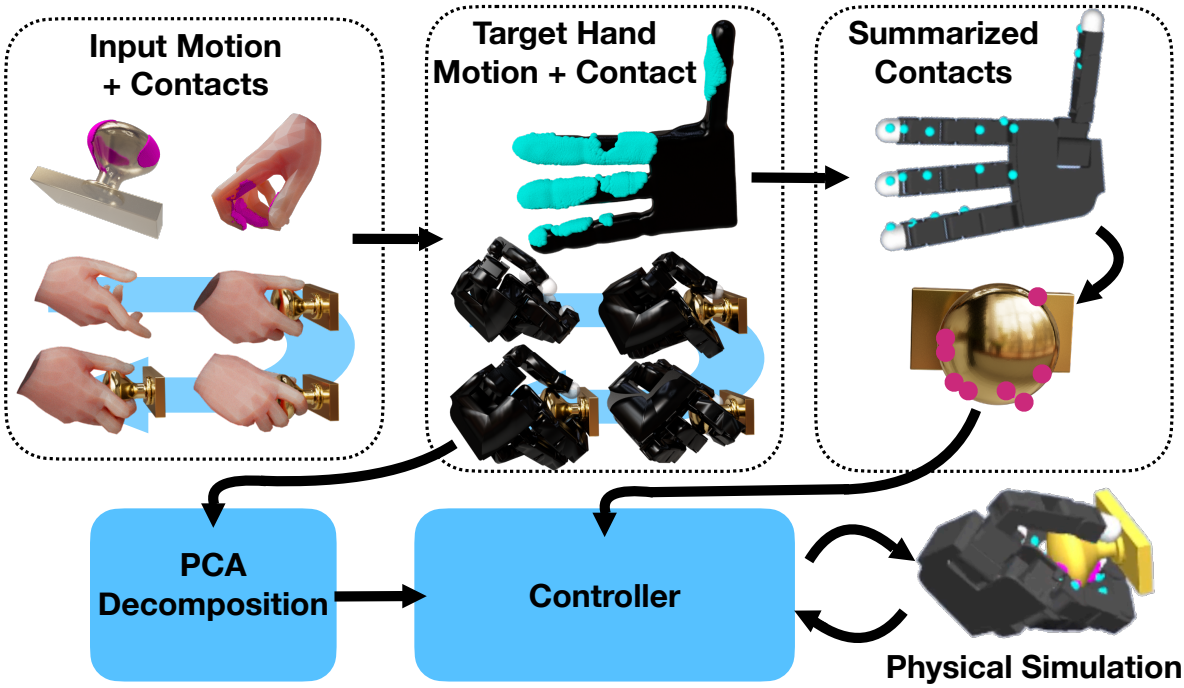

Figure 7.1: Overview of our proposed framework. Starting with a single motion capture sequence with corresponding dense contact areas, we retarget both the motion and contact disbributions to obtain a baseline kinematic trajectory. Contact regions are then downsampled to maintain realtime speeds, while the retargeted motion sequence is decomposed into PCs. The motion baseline seeds the controller, while corresponding contact distributions are used to evaluate sampling costs.

7.2 Proposed Approach

Figure 7.1 illustrates our processing pipeline, while Figure 7.2 provides a more detailed look at our proposed controller. Our goal is to generate an online control policy for a target hand which can be used to execute a task sourced from a single human motion capture sequence. Our framework can be roughly divided into three stages:

- extract a set of *dense corresponding contact regions* between the human hand and object over the motion sequence
- retarget the contacts and generate a C^2 differentiable kinematic trajectory for the target hand
- solve an online optimal control problem to execute the task in physical simulation using the kinematic trajectory as an initial guess

In particular, we address the first two steps using our kinematic motion retargeting framework proposed in Chapter 6. Recall that the output of this framework is a set of B-Spline trajectories for each DOF of the target hand. We propose computing the principal components of the resulting retargeted hand trajectory and decompose the original motion curves into a new set of

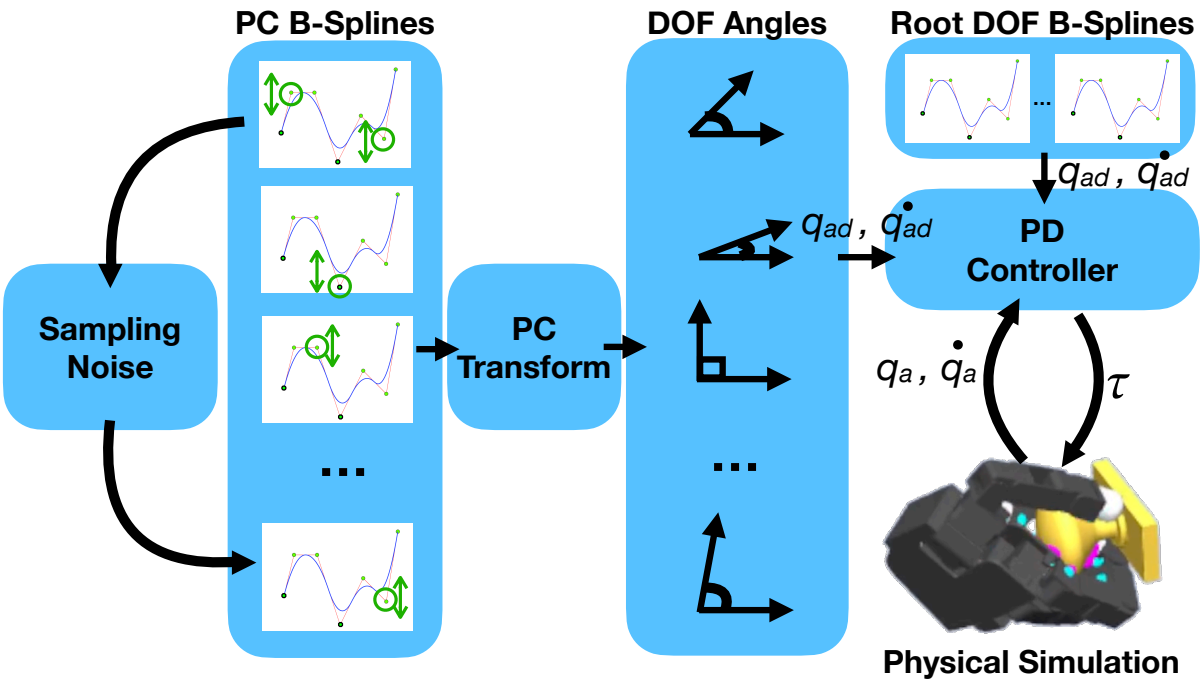

Figure 7.2: Detailed overview of our sampling-based controller overview. In our case, samples are used to adjust control points of the B-Splines representing PC-curves. Final joint torques τ are computed by re-assembling the PC values back into DOF space and executing the desired joint angles via PD control.

time-indexed "PCA-curves", which can then be fit with a new set of B-Splines using the same number of control points as the original curves. The resulting PCA-curves can then serve as the initial guess to our online control problem.

7.2.1 Problem Formulation

We assume a standard policy function π which, assuming system state x, agent controls u, and system dynamics f, is defined as $\pi(x_t, u_{t-1}) \rightarrow u_t \; s.t. \; x_{t+1} = f(x_t, u_t)$. In our case, x_t is comprised of the agent configuration $q_a(t)$, agent spatial velocity $\dot{q}_a(t)$, object configuration $q_o(t)$, and object spatial velocity $\dot{q}_o(t)$. The general online control problem can thus be defined as:

$$\begin{aligned} &\underset{x_{1:T}, u_{1:T}}{\operatorname{argmin}} \sum_{t=0}^{T} l(x_t, u_t) \\ &\text{s.t.} \;\; x_{t+1} = f(x_t, u_t), \;\; \text{given } x_0 \end{aligned} \tag{7.1}$$

where l is the loss function and T is a constant finite time horizon. π is thus the running solution to Eq. 7.1 such that evaluating π from $t...T$ results in minimal cumulative loss.

We define the loss function as:

$$l(x_t, u_t) = \lambda_o ||\hat{q_o}(t) - q_o(t)||^2 + \lambda_c ||c(q_a(t)) - c(q_o(t))||^2 + \lambda_v ||\dot{q_a}||^2 \tag{7.2}$$

where $\hat{q_o}(t)$ is the expected object configuration at time t, $c(q_a(t))$ and $c(q_o(t))$ are corresponding time-indexed expected contact distributions on the agent and object respectively, and $\lambda_o, \lambda_c, \lambda_v$ are weighting coefficients.

One notable difference in our formulation is the control signal u. Typically, u directly controls the agent actuators (e.g. DC motors, servos, etc.); however, we instead define u as the function:

$$u_t = \boldsymbol{P}^T \begin{bmatrix} B_M(PC_0)(t) \\ B_M(PC_1)(t) \\ \dots \\ B_M(PC_k)(t) \end{bmatrix} + \boldsymbol{p} \tag{7.3}$$

where $B_M(PC_i)(t)$ represents ith B-Spline PCA-curve evaluated at time t, $\boldsymbol{P}$ is the PC-projection matrix, $\boldsymbol{p}$ is the mean-centering offset vector, and k is the total number of principal components used, which by default is set to the total number of agent DOFs. All B-Splines are cubic, uniform, and represented using a fixed number of control points M pre-determined during the kinematic retargeting process.

7.2.2 Online Evaluation

We propose using max-selection predictive sampling to solve Eq. 7.1, which is both trivial to implement and allows f to be non-differentiable. For a given sample s taken at time t, we first determine the control points impacted by domain $[t, t+T]$ for each PCA-curve. We can then perturb all impacted control point with Gaussian noise and evaluate the perturbed trajectories across all samples S. The sample of lowest cumulative cost on domain $[t, t+T]$ is selected and the corresponding perturbed control points are committed to the nominal policy π. The process is repeated during every agent planning step throughout the manipulation sequence.

Because B-Splines are entirely determined by control point positions, we can run our simulation at arbitrarily small time steps without escalating problem dimensionality; instead, dimensionality scales with spline degree and time horizon length. For a cubic spline, the entire curve shape of a knot segment can be determined by 4 control points; therefore, if T is sufficiently small, the typical problem dimensionality is $4k$. Our method is especially favorable in comparison to the standard dimensionality of $k \cdot \frac{(T-t)}{dt}$, where dt must be carefully selected with respect to the simulation time step.

Finally, the PC-curves from the nominal policy are converted back to u_t using Eq. 7.3 to compute the desired trajectory in DOF space. We then use PD control to compute the final signals ultimately deployed to the agent.

7.3 Experiments and Results

We select 4 tasks for evaluation: doorknob twisting and pulling, apple pickup and hand off, water bottle lifting and pouring, and stapler pickup and wielding. We select the Allegro Hand

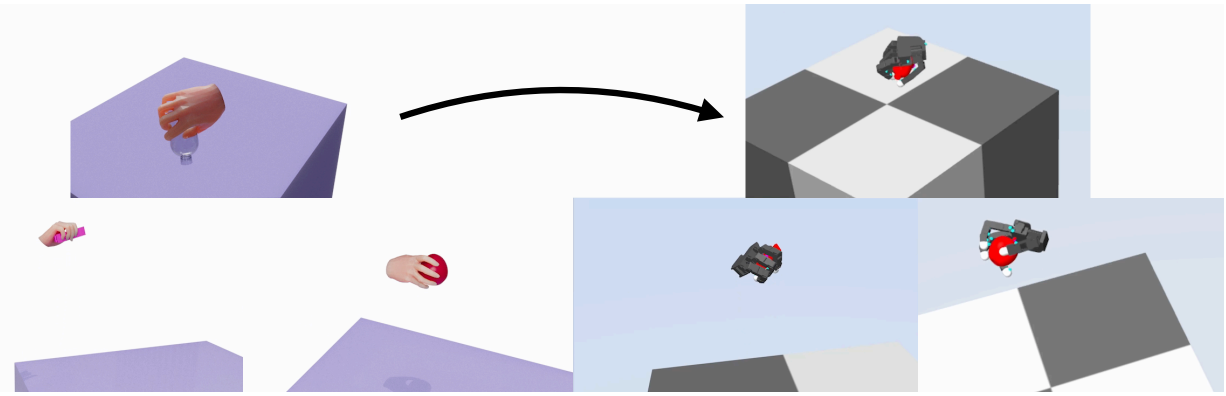

Figure 7.3: Illustrations of our evaluated physically simulated control tasks.

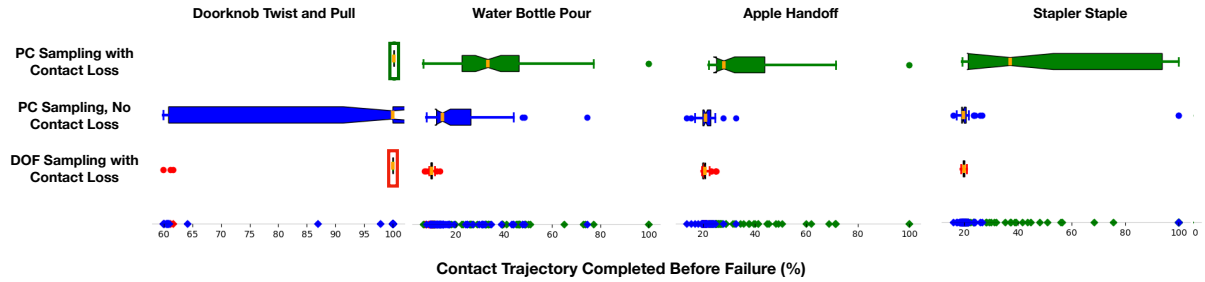

Figure 7.4: Plots of time-to-failure metrics across all 4 tasks expressed as a percentage of time between the first and last expected frames of contact per task. Tasks are ordered from left to right by relative complexity in relation to system dynamics, with the doorknob being the simplest and the stapler being the most complex.

as the target agent. To avoid workspace discrepencies between human and robot arms, we allow the wrist to float with three linear actuators controlling the position and three motors controlling the orientation synchronized using attitude control. We use 128 predictive samples, a 2.5 second time horizon, and 1 and 10 millisecond timesteps for the simulation and agent planning respectively. We slow down the original human demonstration by a factor between 5 and 10x. We use a standard Gaussian noise kernel $\mathcal{N}(0, \sigma^2)$ with $\sigma^2 = 0.2$ for all PCs and $\sigma^2 = 0$ for root DOFs, effectively limiting sampling only to hand poses. All experiments were run on a single Apple M1 chip using 10 parallel threads, of which 7 are used for predictive sampling. All simulations are run using the Mujoco Engine [Todorov et al. 2012] and MujocoMPC framework [Howell et al. 2022]. Figure 7.3 illustrates the evaluated task suite alongside the original human demonstrations.

We consider the time-to-failure metric (TTF) [Qi et al. 2022] for evaluating policy performance, with longer TTFs indicating better task performance. In our case, a task is considered failed when the hand fails to maintain any contact with the object for 300 milliseconds despite more than six down-sampled expected contacts existing during those timesteps. The assumption then is that the hand has completely lost control of the object and will be unable to recover. Each task was repeated 50 times.

Green boxplots in Figure 7.4 illustrate the results of our method across all tasks. We also ablate our approach by considering an alternate cost formulation which does not include a contact correspondence term, as well as a more typical sampling strategy over the original joint configuration space with each DOF $\sigma^2 = 60^\circ$. Blue and red boxplots in Figure 7.4 illustrate results for each ablation respectively. Although we observed that ablations were competitive on "simpler" tasks such as the environmentally constrained doorknob, performance rapidly deteriorated with increasing task complexity. However, the inclusion of our contributions consistently resulted in longer TTFs across the entire task suite.

7.4 Discussion

We were pleasantly surprised to find that, despite its simplicity, our proposed formulation was capable of executing non-trivial tasks. The massive performance gain in comparison to ablated baseline policies was especially encouraging to observe and strongly validated the importance of our contributions in relation to getting online MPC from a single demonstration, and particularly from data sources that are already publicly available, to actually work.

We also observed interesting emergent behaviors mid-manipulation, such as spidering, sliding, and other non-trivial finger movements that collectively adjusted objects into more favorable configurations. We argue that expected corresponding hand and object contact distributions are largely responsible for encouraging such behavior because said distributions implicitly inform agents *how* to achieve a desired configuration from a deviant intermediate state. Additionally, we saw that sampling over the PC space provides *better quality samples* of meaningful synergies than the original DOF space, which, combined with our B-Spline formulation, helps combat the high dimensionality problem. However, it is currently unclear *why* that is the case outside empirical observation.

But our method is not yet reliable. Although we observed nearly perfect completion on the doorknob task, completion rates for the remaining tasks were less compelling. These results were unsurprising given the substantially higher difficulty of manipulating free-floating objects compared to those which are environmentally constrained. The highly stochastic nature and poor knowledge retention of rollouts from max-selection predictive sampling over long sequences also severely impacts the reliability of online evaluations.

Our method also has yet to be deployed in the real world. There are two main hurdles to address for real-world deployment: online hand-object state estimation, and the fidelity of future state predictions based on real-world dynamics. Fortunately, both problems are long-standing, widely studied, and have rich existing literatures. Although mutual occlusion problems are unavoidable in vision-based tracking approaches for dexterous manipulation, one particularly exciting body of work that has the ability to jointly assist in both of these challenges is tactile sensing, which has already lead to impressive results in online state estimation [Suresh et al. 2024]. Our method is especially well poised to take advantage of such advancements especially because our proposed cost function is fairly simple and does not assume smooth dynamics, require known object mass-inertial properties, or depend on higher order differential terms outside hand velocity. Because our controller also operates fully online, it is feasible to deploy on device immediately without the need for pre-training in simulation and subsequently dealing with Sim2Real headaches. Finally, our framework's design of dealing directly with previously

collected data as input has the potential to completely circumvent the need for tele-operation.

7.5 Summary

We have presented an online adaptive control framework for complex, contact-rich dexterous manipulation tasks in physics-based simulation from a single human hand motion. To do so, we introduced two simple, but powerful contributions within the context of a classical MPC framework: time-indexed corresponding hand-object contact area distributions as an explicit cost term, and a search strategy over the B-Spline control points of hand pose principal components.



Chapter 8

Applications

The contact models proposed in this thesis have also been utilized in several compelling applications as part of collaborative efforts. We will touch upon two such applications in this chapter: rapid prototyping of hand designs, and dataset annotation at scale.

8.1 Rapid Prototyping of Hand Designs

Drafting and refining prototypes of hand designs is a challenging and time-consuming endeavor. One particular complication is that the functionality of the hand must typically be evaluated independently of the task(s) in which it is expected to operate. For example, while it is relatively straightforward to evaluate a CAD model candidate's kinematic range of motion in isolation, it is often unclear how that range of motion will be utilized when grasping a specific object (e.g. a box or flashlight). This issue is further compounded when the hand exhibits further complexities such as underactuation or finger configurations that don't exactly match the human hand. Under such constraints, it can be challenging for a design to *quickly visualize how* a particular design candidate will grasp an object and, furthermore, how changes to the candidate will impact the result. Evaluating such candidates prior to fabrication offers an opportunity to considerably improve prototyping turnaround times since fabrication and evaluation in the real world is a highly time-consuming endeavor [Mannam et al. 2023].

Our proposed contact area models provide a means of partially automating the design candidate evaluation process. Suppose a designer wishes to design an anthropomorphic hand that can mimic the way a human grasps an object. Although the designer may already have a design candidate template in mind, optimizing certain critical parameters such as the placement and starting configuration of the thumb can be challenging. Rather than having the designer perform this manually, we can instead supply a concrete task description by means of a human contact map captured from the real world (e.g. Chapter 9). These contact regions can then be parameterized on the object and transferred to the hand design candidate, which will then allow our grasp computation framework (5.3.4) to compute the optimal corresponding hand pose to match the object contact distribution. If the design candidate's connectivity does not change, then the designer can freely update the geometry and kinematics of the design and subsequently solve for the optimal pose under the proposed mutation.

Figure 8.1 illustrates the proposed iterative design framework. The framework was evaluated

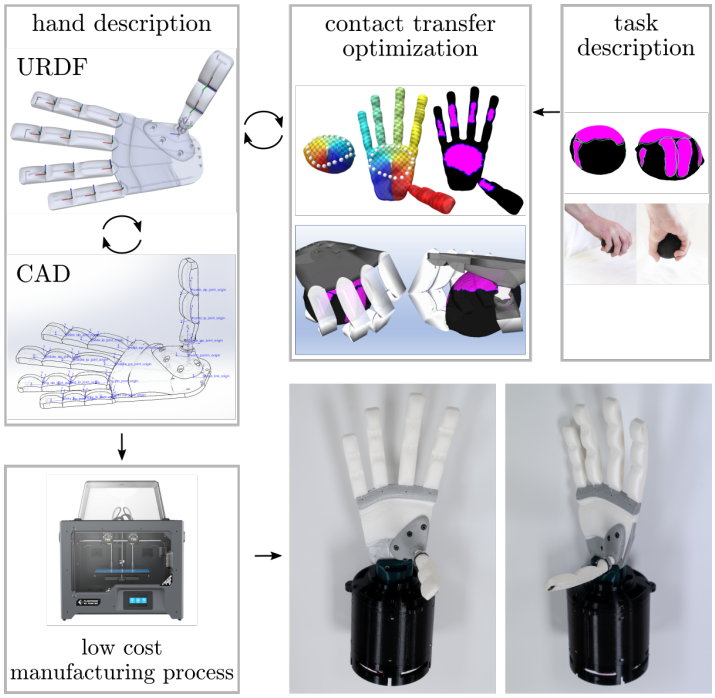

Figure 8.1: Iterative hand design prototyping framework enabled by contact area transfer and grasp computation

via a design study of thumb placement with respect to performance across 4 tasks illustrated in Figure 8.2.

Based on both the computed solutions as well as the hand designer's familiarity with the fabrication process and general intuition, candidate (d) was selected for fabrication. Finally, as illustrated in Figure 8.3, the fabricated candidate was then assembled and successfully managed to replicate the tasks in the real world via controlled via open-loop poses.

8.2 Scalable Dataset Annotation

Although we demonstrated that our contact-driven artist tools (Chapter 5) were capable of enabling animators to create high quality results, it is unclear whether such tools are intuitive enough for a completely untrained user with no existing background in character posing, or even exposure to 3D modeling software. Previous work has demonstrated that even minimally trained users are capable of providing high quality annotations of human contact when given a simple interface [Tripathi et al. 2023]. We were interested to see if a similar interface would also enable everyday people to utilize our contact-area-based artist tools, which, if successful, would enable high quality dataset annotation at scale via internet crowd-sourcing.

We tested this conjecture in the context of full-body human pose estimation — a long standing and high value problem in computer vision. The goal in this work specifically was to jointly reconstruct image-aligned human-object interactions from a single in-the-wild image, which is an extremely difficult task given the lack of assumptions which can be made about an image (e.g. lighting and background conditions, mutual occlusions, baseline human pose, etc.). A second

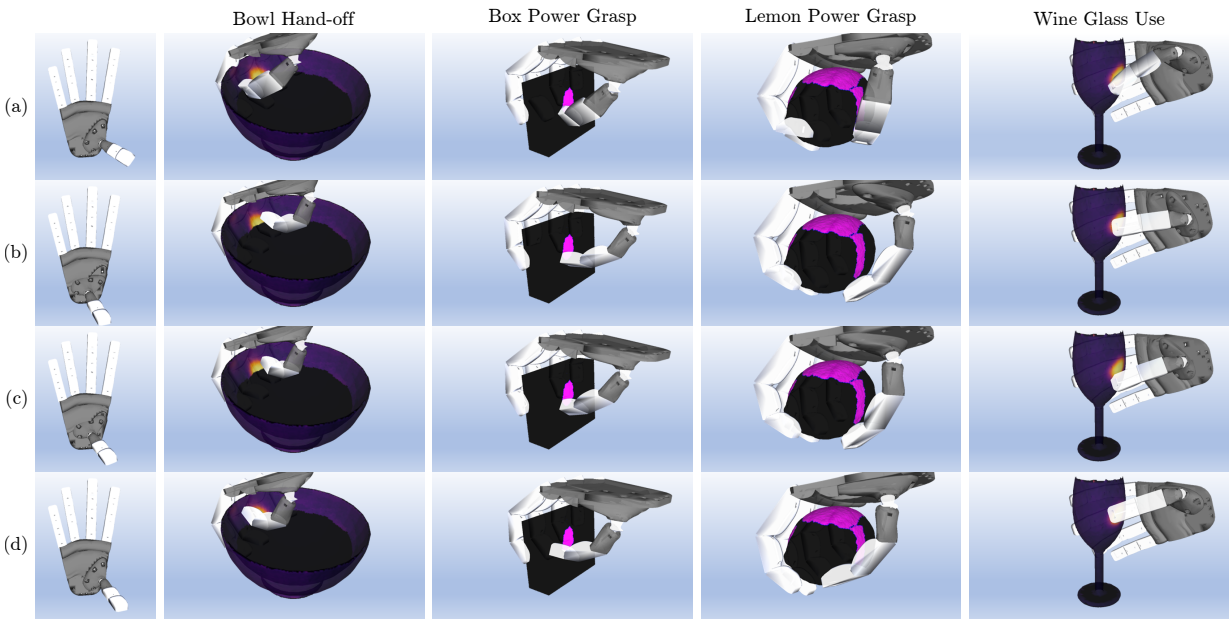

Figure 8.2: Anthropomorphic hand designs featuring a variety of different thumb placements (*column 1*) are evaluated for their ability to grasp a bowl (*column 2*), a box (*column 3*), a lemon (*column 4*), and a wine glass (*column 5*).



Figure 8.3: Based on the results in the design study shown in Figure 8.2, candidate (d) is selected for fabrication. All grasps are successfully completed in a pick and place setup using key-framed open-loop poses that match the results obtained in our simulation.



Figure 8.4: Sample of generated human pose estimation results grounded via corresponding contact regions.

and equally challenging issue is the wide variance of object meshes. Unlike the standardized SMPL human model [Loper et al. 2023], there is no standardization for object meshes (even within a single object category from widely used public datasets [Chang et al. 2015; Deitke et al. 2023]). Our tools must therefore also be robust enough to work across such variations.

To reduce annotator cognitive load, we limited user workloads exclusively to transfer operations using the axis contact model. SMPL human body contacts for all sourced in-the-wild images were first generated via DECO [Tripathi et al. 2023] and subsequently parameterized under a novel, PCA-based default axis generation scheme. Users were then tasked with transferring the resulting parameterized contacts to an automatically selected object mesh that reasonably approximated the object in the image. After the annotations were received, an optimization procedure was performed to both align the human and object with respect to the corresponding contact distributions as well as jointly against the image.

Figure 8.4 illustrates a small number of results generated using the proposed annotation pipeline, with the contact regions rendered in each cutaway. We have so far managed to successfully crowd source over 3,000 annotations via Amazon Mechanical Turk in relation to a target goal of 5,000 results, thereby (at least so far) successfully validating our conjecture. Furthermore, results in comparison to baseline methods [Zhang et al. 2020] so far have also highlighted a secondary observation: exact pairwise correspondences of individual points comprising corresponding contact regions, which we obtain by design from our transfer process, enables significantly stronger alignment of humans and objects in comparison to general correspondences of all points between regions. Our artist tools therefore do have the potential to provide significant value in dataset collection pipelines at scale for contact-rich tasks, and by extension significantly advance downstream contact inferencing for a wide range of applications.

Chapter 9

Real World Capture

We will now turn our discussion from purely digital contexts to real world settings, starting with capture and reconstruction of contact areas from human demonstrations. We briefly touched on two methods of doing so in Chapter 1 and Chapter 2 — one being the approach of Kamakura et. al [Kamakura et al. 1980] which mixed ink and glue to capture contact areas on hands, and the other being the approach of Brahmbhatt et. al [Brahmbhatt et al. 2019a] of capturing heat signatures on objects post manipulation using thermal camera imaging. We expand this literature by introducing two new methods: low cost RGB imaging via thermochromic paint for human-object interactions (HOI), and tactile-augmented motion capture for human-human interactions (HHI).

9.1 Thermochromic Paint Imaging for HOI

9.1.1 Motivation

Ideally, we would like to simultaneously capture contact areas on the object and the hand during interaction; unfortunately, doing so is not physically possible due to the mutual occlusions generated by the contacting hand and object. Instead, we are realistically limited to capturing the contact signature only on one surface and reconstructing the other.

The ink and glue mixing technique of Kamakura et. al [Kamakura et al. 1980] provides us a means of capturing the hand signature. After the grasp is complete, the subject can simply re-pose the hand in a new free-space configuration which can expose all the areas at once in a single image. This technique is thus easy and extremely cheap to perform; however, it is difficult to recover the object signature.

Thermal imaging provides a path forward for capturing object contact signatures. The technique is especially effective in the case of HOI because internal body temperatures are significantly higher than the surrounding environment; consequently, interaction with other objects typically causes heat to diffuse into the contacting object and leave clearly visible and high resolution signatures over short time scales. Brahmbhatt et. al [Brahmbhatt et al. 2019a] propose capturing this signature with thermal cameras, which image in the infrared rather than visible light spectrum.

Unfortunately, thermal cameras are notoriously expensive and often sport poor pixel reso-



Figure 9.1: Capture setup used for thermochromic paint imaging along with (upper left) pigment and applicator materials.

lutions when compared to standard RGB cameras. As a concrete example, the Flir A65 [1] costs roughly $10,000 and sports only a 640 x 512 pixel resolution. They also rely on specialized equipment to calibrate and register when used in stereo applications. These impediments make it difficult to justify investing in a thermal camera setup if only a small number of captures are necessary for a certain task. Consequently, we set out to find an alternative means of object contact capture which required significantly less upfront investment.

9.1.2 Process

We instead propose the use of thermochromic paint for contact imaging. Thermochromic pigments are commodity items readily available on Amazon or other retailers for tens of dollars. More importantly, these pigments expose signatures ordinarily visible only in the infrared spectrum to visible light. As such, signatures can be captured from by standard RGB cameras — including the widely used Intel RealSense line.

Our proposed imaging process works well with standard PLA 3D-printed objects. After the print is complete, the objects can be spray painted with acrylic clear gloss mixed with the pigment. We advocate using a high-contrast pigment (e.g. black to pink) with as low an activation temperature as possible.

After the grasp is completed, the objects can be placed on a standard turntable and scanned via an Intel RealSense. Importantly, the turntable and object should be placed past the depth

[1] https://www.flir.com/products/a65/

near-field limit to improve point cloud fidelity. The camera should be fixed to a tripod and angled to capture the principal surface(s) contacted during the manipulation. Figure 9.1 illustrates the suggested setup. No cameras need to be calibrated due to manufacturer provided estimates of both RGB and depth camera intrinsics and extrinsics.

Since thermal signatures are highly ephemeral, we found it useful to use hand warmers prior to data collection. The collection should last approximately 10 seconds for a strong signal to appear. Three assets are required per frame for reconstruction: a fixed scale depth image, a color image, and a point cloud. Anywhere from 10-12 representative frames from the video should be sufficient for textured mesh reconstruction.

The entire approximate cost of materials required to replicate our experiments, with plenty of materials left over, is $300, of which more than 90% stems solely from the cost of the camera and the motor used to drive the turntable.

9.1.3 Textured Mesh Reconstruction

We follow the methods of Brahmbhatt et. al. [Brahmbhatt et al. 2019a] for registering the imaged thermal signature to the desired object mesh. The final vertex-colored mesh can be generated using Open3D [Zhou et al. 2018].

First, we segment the point cloud 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) [Besl & McKay 1992] is used to register the segmented cloud to the original mesh used to print the object. Point cloud processing was repeated for the aforementioned 10-12 sampling frames. ICP alignment was then be used to extract 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 [Zhou & Koltun 2014].

9.1.4 Thermal Camera Comparison

Figure 9.2 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 (cost: >$10,000). Despite the thermal camera's stronger signature, the signature is still 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 may present some drawbacks including ambient temperature and illumination sensitivity, our proposed technique is offset by the large difference in upfront cost.

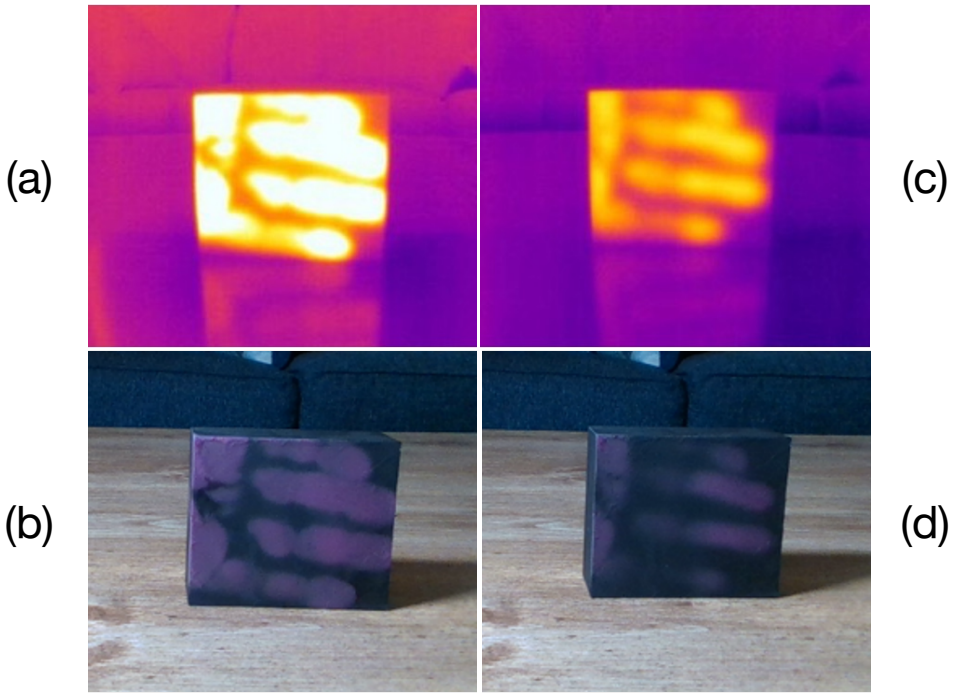

Figure 9.2: Comparison of our (bottom) thermochromic RGB imaging procedure with top a Flir A65 thermal camera at full signal strength (a, b) and after heat dissipation (c, d).

9.2 Tactile-Augmented Motion Capture for HHI

9.2.1 Motivation

We have so far only considered contact areas in terms of geometry, correspondence, and kinematic operations. However, these perspectives do not provide insight into how forces are distributed over the areas or how those forces evolve over time. To understand and analyze the physical dynamics of contact area evolutions, we must first be able to capture and reconstruct relevant data from realistic demonstrations.

There has been a considerable recent push in research focused on jointly capturing motion and force information from human demonstrations. A wide array of sensing modalities have been investigated to date, ranging from force-sensitive resistive [Sundaram et al. 2019; Luo et al. 2024] and capacitive arrays [Pressure Profile Systems 2026] to more specialized approaches such as embedded-camera-based [Fang et al. 2025], magnetic [Adeniji et al. 2025; Yin et al. 2025], acoustic [Mao et al. 2025], and Fiber Bragg Grating sensors [Xing et al. 2025]. But while there have been significant efforts centered on examining different hardware modalities, there have been considerably less efforts targeting reconstruction of contact areas and forces over time. We provide one such approach in this thesis, focusing less on the capture hardware and more so on creating high fidelity reconstructions. Our contributions are positioned within the subdomain of HHI, and in particular the focus task of assisted bathing.

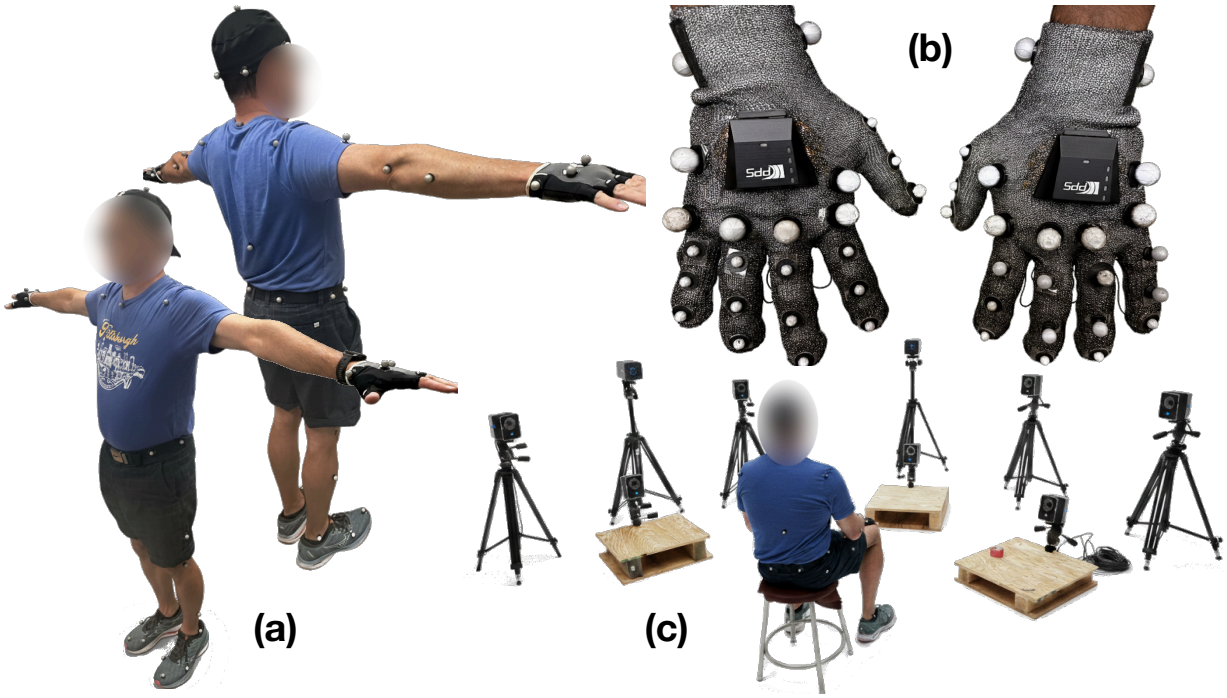

Figure 9.3: Optical marker configurations used on (a) human subjects and (b) tactile force gloves. (c) Cameras are floor mounted and positioned in a semi-circle configuration.

9.2.2 Capture

Assisted bathing demonstrations consist of a clinician who performs the bathing task and a subject on who the task is performed. Subjects were outfitted with 52 optical markers for motion tracking and encouraged, but not required, to wear dark summer clothing to minimize optical reflectance while exposing as many extremities as possible for direct skin contact. Clinicians were instructed to wear a pair of commercially available tactile sensing gloves [Pressure Profile Systems 2026] and perform bathing motions on the subject's skin, or alternatively clothing if skin exposure was not possible, using a hypoallergenic wet wipe. The dorsal side of each glove was outfitted with 23 optical markers, while the palmar side was outfitted with a layer of adhesive-backed water-resistant grip material [3M™ 2026] to protect the sensor electronics from incidental moisture damage from the wipe. The grip material was laser cut into a "tabbed" pattern designed to maximize surface coverage with minimal impact to range-of-motion (ROM). Notably, we found that the grip material provided the best compromise between water resistance, adhesion, and deformability in comparison to other material candidates (e.g. vinyl, silicone). Figures 9.3 illustrates our capture equipment.

The sensory array of each glove consists of 65 distinct "taxels", or pressure-sensitive pads, distributed throughout the palmar and digital surfaces. The sensors measure changes in pressure via differential capacitance [Baxter 1997] relative to a baseline taken at the start of each collection. We assume all forces are normal to each taxel.

Subjects were positioned upright on a stool surrounded by a set of 20 Vicon Vantage-V16 cameras [Vicon Motion Systems 2026]. Clinicians were asked to bathe subjects in the same way

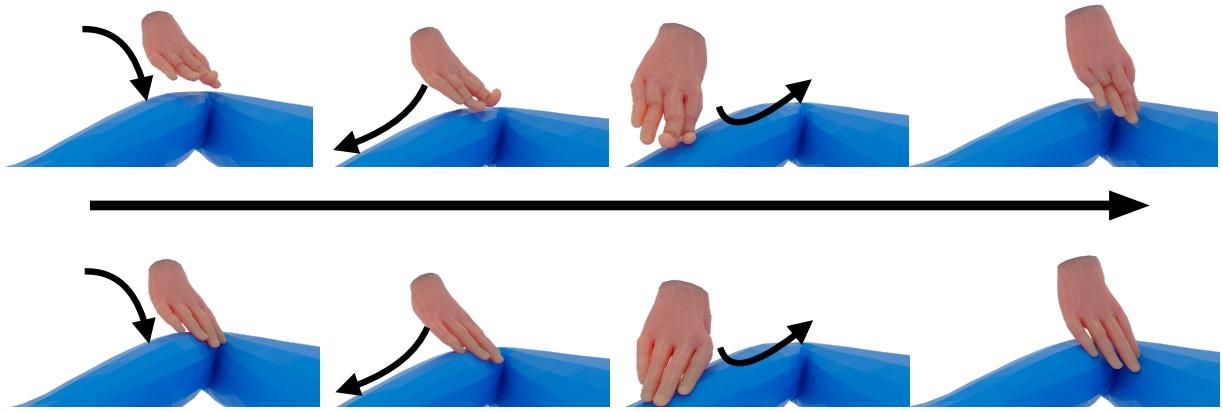

Figure 9.4: Comparison film strip of a leg bathing reconstruction using (top) MoSh++ [Mahmood et al. 2019] and (bottom) our reconstruction pipeline. Black arrows illustrate subsequent movement directions. Our method effectively resolves finger contortion and gap artifacts without any significant parameter tuning.

as they would a patient in a medical or assisted-care facility. Markers inhibiting bathing motions of specific body parts were temporarily removed during captures of those parts.

Each capture consisted of the tracked 3D positions of all subject markers and glove optical markers, as well as pressure readings from all 65 taxels per glove. Tactile and marker data were streamed at 10 and 120 Hz respectively and synchronized using Vicon timecodes. All subjects and clinicians were also asked to perform ROM tests prior to their first capture. We ensure all markers are labeled and have unbroken trajectories.

9.2.3 Shape and Motion Reconstruction

We convert tracked marker positions and force measurements into complete reconstructions of shape, motion, and contact using a multi-stage pipeline. We first utilize the SMPL-X [Pavlakos et al. 2019] and MANO [Romero et al. 2017] parametric models to estimate each subject's body and each clinician's hand shape parameters from their respective ROM tests using a random subset of 50 frames, as well as canonical marker position estimates relative to the baseline shape. We then estimate all pose and corrective blend shape parameters for each bathing capture trajectory using the MoSh++ solver [Mahmood et al. 2019]. This process effectively converts point-based marker trajectories into complete mesh trajectories of the body and hands.

However, as illustrated in Figure 9.4, the process routinely produces serious artifacts such as highly contorted finger poses and major gaps of separation between the hands and body, rendering it inadequate for high quality bathing reconstruction. These artifacts largely result from the necessary process of fitting bare-skin hand shape parameters to gloved captures. To mitigate these inherent discrepancies, we implement a series of straightforward, yet robust solutions.

First, to combat finger contortion artifacts, we replace the baseline fitted hand mesh with a constrained skeleton-driven reduced-DOF model that honors anatomical joint limits [Magee 2013]. We use the same baseline shape parameters and drive the constrained mesh using linear

blend skinning [Magnenat-Thalmann et al. 1989]. Joint angles are computed per frame using the following objective:

$$\begin{aligned} \boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \quad & \lambda_s \sum_{i=0}^{V} ||v_{bi} - v_{ci}||_2^2 + \lambda_j \Gamma_J \\ \text{s.t.} \quad & \boldsymbol{\theta_L} \leq \boldsymbol{\theta} \leq \boldsymbol{\theta_U} \end{aligned} \tag{9.1}$$

where v_{bi} and v_{ci} represent corresponding vertices of the baseline and constrained hand meshes respectively, V is the total number of MANO hand vertices, Γ_J is a regularizer against initial constrained hand pose deviation, $\boldsymbol{\theta}$ is the constrained hand DOF vector, $\boldsymbol{\theta_L}$ and $\boldsymbol{\theta_U}$ are the lower and upper DOF bounds, and λ_s and λ_j are weighting hyperparameters. We adopt an existing motion synthesis pipeline to solve for the full trajectory $\boldsymbol{\Theta}^*$ of each constrained hand based on the per-frame pose estimates [**lakshmipathy2025kinematic**], and refer to this fitting process as the *shape match* pass.

We next utilize contact data to combat the gap artifacts. Starting from one-time calibrated hand taxel position estimates, we estimate corresponding body contact regions per frame using closest-point queries [Sawhney 2021] and store the trajectories in barycentric coordinates. Next, utilizing tactile force data, we binarize the signal into activations and filter out all inactive contacts. We then perform a second *contact match* pass by adding a single additional objective term to Eq. 9.1:

$$\lambda_c \sum_{i=0}^{C} ||c_{hi} - c_{bi}||_2^2 \tag{9.2}$$

where c_{hi} and c_{bi} represent corresponding contact points on the constrained hand and body meshes respectively, C represents the total number of taxels per hand, and λ_c is a weighting hyperparameter. Final reconstructions are complete following the contact pass. We use hyperparameters $\lambda_s = 1, \lambda_j = 50$ and $\lambda_c = 40$ for all captures and 100 iterations per optimization.

9.2.4 Experiments and Results

We used our capture process to collect a dataset spanning six subjects and three clinicians. We summarize, visualize, and analyze our dataset in the proceeding subsections.

Dataset Summary We used our capture process to collect a dataset spanning six subjects and three clinicians. Table 9.1 tabulates the distribution of captures across all subjects and clinicians in our dataset. Subjects ranged in age from 22 to 31 years, with an equal distribution of males and females. Renders of all reconstructed subject body shape parameters are available in Figure 9.5. Clinicians were all female, aged 30 to 32 years. The average capture duration was 2007.34 frames ($\sigma = 536.615$).

Each capture is categorized by one of two assistance variables (limited or full) and one of two pressure intensities (mild or strong). Limited assistance captures assume subjects can independently lift and hold their limbs, while full assistance captures require clinicians to manually raise and lower them throughout the bathing process. Clinicians use their discretion

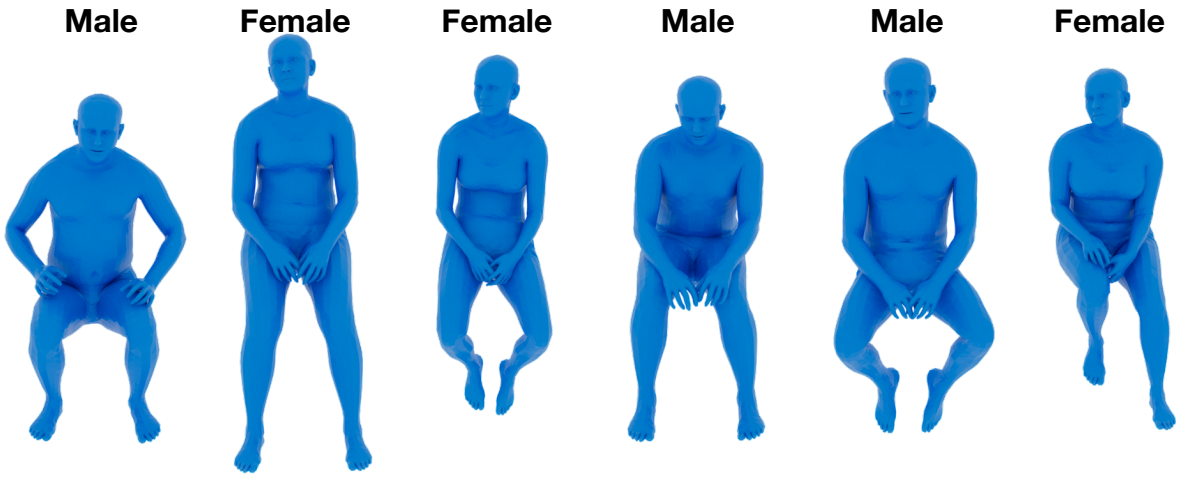

Figure 9.5: MoSh++ [Mahmood et al. 2019] shape reconstructions of all subjects.



Figure 9.6: Sample stills of captures generated using our joint tactile and motion reconstruction pipeline.

Table 9.1: Distribution of all captures in our dataset segmented by body part, assistance level, and pressure intensity.

	Limited, Mild	Limited, Strong	Full, Mild	Full, Strong
Right Arm	10	8	9	7
Right Leg	6	6	3	3
Right Thigh	3	3	N/A	N/A
Right Oblique	2	2	N/A	N/A
Left Arm	7	6	7	6
Left Leg	6	6	4	3
Left Thigh	3	3	N/A	N/A
Left Oblique	2	2	N/A	N/A
Back	6	6	N/A	N/A
Neck	7	1	N/A	N/A
Face	1	N/A	N/A	N/A

to differentiate between mild and strong pressure, as no explicit guidelines are provided. We also note that not all variable permutations are available per body part. For example, face bathes only consist of mild pressure demonstrations due to the region's sensitivity, areas such as the beck, neck, and obliques cannot possibly require full support, and multiple clinicians opted to not provide full assistance for legs because the entire limb could already be accessed from a seated position. We acknowledge that bed-bathing may provide some additional demonstrations not represented in our dataset. Finally, while we cover most parts of the body, we do not have captures of hand bathing due to tracking difficulties or of the chest and pelvic regions due to IRB restrictions. Figure 9.6 illustrates several stills of captures from the dataset.

Visualizations of Contact Coverage Figure 9.7 visualizes body coverage across all captures for two independent subjects, which we measure as all triangles that contain a contact point at any time during a bathing trajectory. Figure 9.8 visualizes the most commonly contacted taxels on the left and right hands across all captures, as well as the distribution of forces across taxels. Figure 9.9 illustrates a sample bathing "stroke", or surface path integral of a contact distribution over time, of the right hand up the back.

Quantitative Reconstruction Evaluation We quantitatively evaluate the quality of our reconstruction pipeline using two metrics: *L2* distances between estimated body contacts



Figure 9.7: Contact coverage (yellow) over all captures of a representative (a) male and (b) female subject.

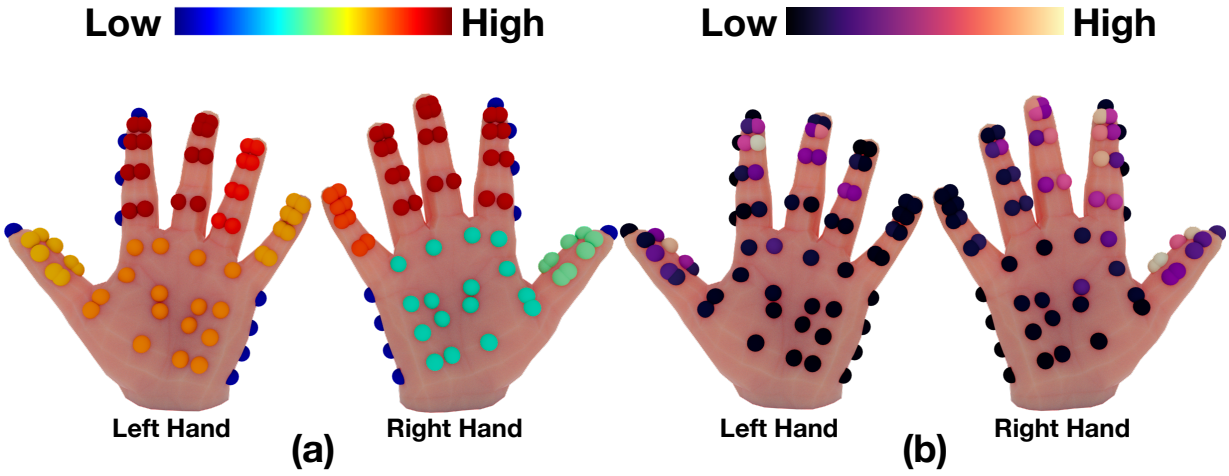

Figure 9.8: Visualizations of the most commonly (a) contacted taxels and (b) applied pressures across all captures.

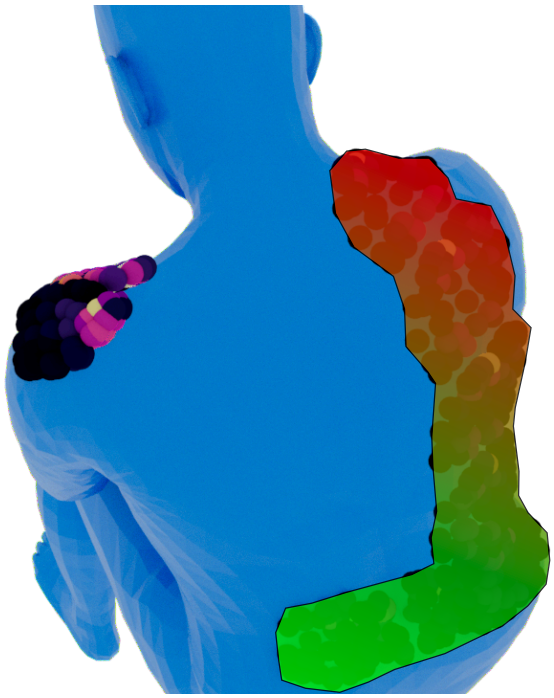

Figure 9.9: Visualization of a bathing “stroke" from the (green) bottom to the (red) top of the back. The subject is stabilized by a contact distribution at the left shoulder during the stroke.

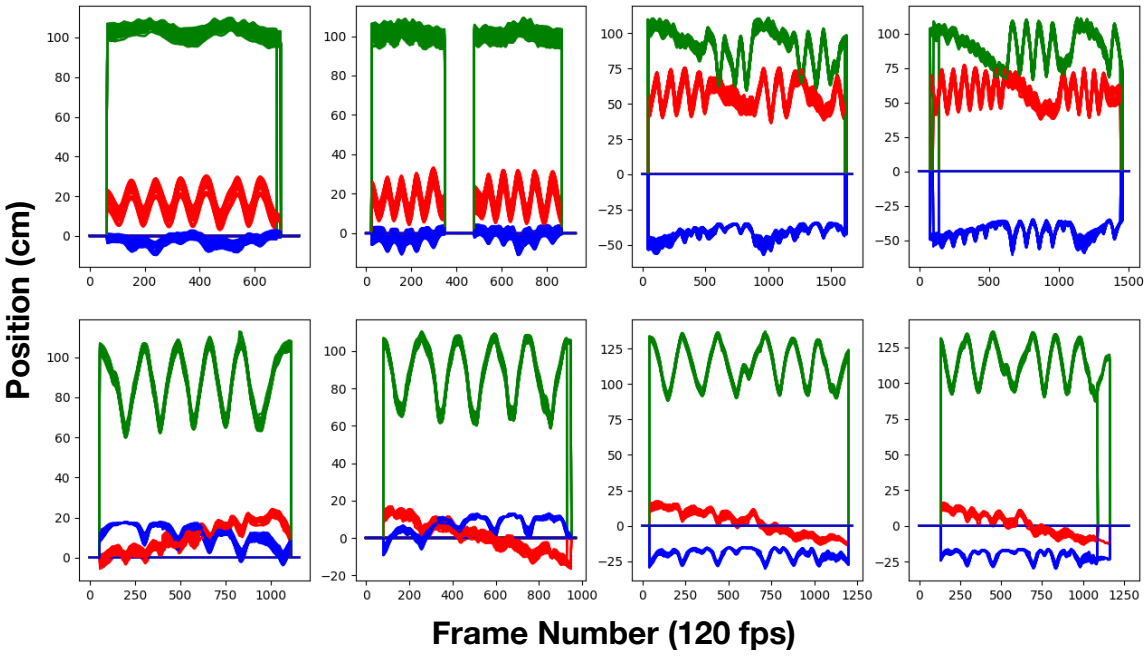

Figure 9.10: Illustrations of 8 representative contact trajectories, separated into (red) X, (green) Y, and (blue) Z channels. All positions are in world coordinates. Each band is comprised of 65 individual contact point trajectories. Inactive taxels are zeroed out.

Table 9.2: *L2* Contact distance metric comparison between MoSh++[Mahmood et al. 2019] and our reconstruction.

L2 Distance Comparisons (cm)			
	Median	Mean	St. Dev.
MoSh++	1.714	2.067	1.464
Ours	**0.536**	**0.837**	**0.842**

and glove taxels, and smoothness of body contact trajectories. Table 9.2 tabulates distance comparisons of our method against the original MoSh++ [Mahmood et al. 2019] reconstruction across 10 sample bathing demonstrations, while Figure 9.10 plots the trajectories of all active wiping hand body-projected taxels across multiple back bathing demonstrations. We observe high spatiotemporal correlation in the projected taxel trajectories and that oscillations coincide with multiple back-and-forth wiping passes over back segments.

9.2.5 Data Insights and Lessons Learned

As strongly evidenced by Figures 9.9 and 9.10, bathing contact trajectories are *highly* correlated, smooth, and dynamic. Modeling contacts as individual points, instantaneous static entities, or simple constraints, as is still commonly done by many long-standing analysis techniques [Mason 2001; Ferrari & Canny 1992; Lynch & Park 2017] and modern simulators [Todorov et al. 2012; Makoviychuk et al. 2021], is highly limiting for representing our captured data. Area- or volume-based [Elandt et al. 2019] contact models — when paired with well-defined velocity parameters — offer significantly more promise for capturing the sustained interactions necessary for bathing.

The recorded tactile signals for mild pressure demonstrations, regardless of clinician or body part, are quite low (in multiple cases below the minimum sensitivity of 0.04 N) and thus largely indistinguishable from noise. We therefore strongly advise prioritizing high contact sensitivity in hardware design for low-pressure data acquisition, or alternatively focusing efforts on strong pressure captures.

Unsurprisingly, we found that signals between adjacent taxels tend to be highly correlated regardless of the task. We argue there is little value gained from high resolution sensing of regions unaffected by pose changes (e.g. the proximal phalanx of an individual finger). In contrast, hand shapes induced by pose changes, as well as hand shape in relation to local body shape, have a much more significant impact on pressure distributions. Therefore, we argue that thinking of sensor coverage in terms of *regions between joints* is more valuable for bathing than intra-region resolution.

Interestingly, in captures where both hands were utilized, the majority of high pressure measurements came from the supporting hand rather than the wiping hand. This observation indicates that body stabilization requires substantially more force than wiping. We also observed a *substantial* amount of patient movement, both voluntary and in response to bathing forces.

Finally, while tactile information does help determine precise times and magnitude of contact, its utility is limited without context from motion and shape. Tactile data alone cannot determine *between which bodies* contact is happening or *why* forces change over time. Both issues made

identifying true contact events between the hand and body difficult to automate. Multiple contact events between the thumb and thenar region of the palm, for example, resulted from gripping the wipe rather than applying body pressure. Drops in pressure could be attributed to either breaking contact or simply reducing contact force. Low, but increasing pressure signals could be actual contact events or drift noise. We found that while motion and shape data are vital for clarifying such events, they are not a total solution; a holistic analysis of all three channels together is still required to accurately interpret results.

9.3 Summary

We have presented two new methods for capturing contact area signatures. Our first contribution presented a method for capturing HOI contact area signatures using thermochromic paint and a number of additional commodity items. Our setup costs are considerably lower than pipelines using thermal cameras, which renders our approach more practical for tasks requiring the collection of only a small number of signatures over infrequent sessions. Despite our significantly lower setup costs, we demonstrated that the quality of the recovered signature is comparable to that of a thermal camera and comes with the same drawbacks. Our second contribution presented a high fidelity capture and reconstruction pipeline for HHI that effectively combined motion and tactile data streams. This allows us to understand and analyze the force profiles of contact areas as they shift over time, and culminated in the first dataset of high quality assisted bathing demonstrations.



Chapter 10

From Contacts to Systems

Building upon our shift from digital to physical, we now turn our attention from computationally simulated embodiments to real robots. In this penultimate chapter, we will utilize contact areas captured and reconstructed in Section 9.2 to calibrate and control a robot bathing system. In doing so, we will offer a glimpse into what it takes to transfer human-human subject demonstrations to robots and show that embracing contact can address a number of non-trivial problems encountered during the process.

10.1 Motivation

Bathing is an essential activity of daily living and important component of long-term personal hygiene [Dunlop et al. 1997]. The Katz scale [Katz et al. 1963] identifies the ability to bathe as one of six major categories that quantify an individual's level of independence; consequently, losing this ability significantly impacts quality of life. As the number of Americans with bathing-related impairments rises and the pool of trained caregivers shrinks [Gill et al. 2006], the demand for bathing assistance is rapidly outstripping the available supply of care.

Robot-assisted bathing solutions have the potential to significantly expand access to care while reducing existing caregiver burden [Czuba et al. 2012; Darragh et al. 2015]. However, building systems that can safely and reliably perform the sustained, highly dynamic, and contact-rich human interactions inherent to bathing is technically challenging. While several impressive systems have emerged recently [Erickson et al. 2019; Liu et al. 2022; Madan et al. 2024], they generally rely on idealized definitions of bathing ranging from best practice guidelines [Madan et al. 2024] to coarse-grained performance metrics (e.g. debris removal percentages [Huang et al. 2022; Liu et al. 2022]) rather than real data from human demonstrations. Consequently, many important details such as realistic hand movements and shape adjustments across different body segments, reactive corrections in response to patient movement, the evolution of contact regions between the hand and body over time, the total body area covered by the strokes, and forces applied over those regions are not considered. The dataset collected in Section 9.2 offers the information necessary to address this limitation, and, as illustrated in Figure 10.1, we can utilize that data to transform human demonstrations into a fully functional robot bathing system.

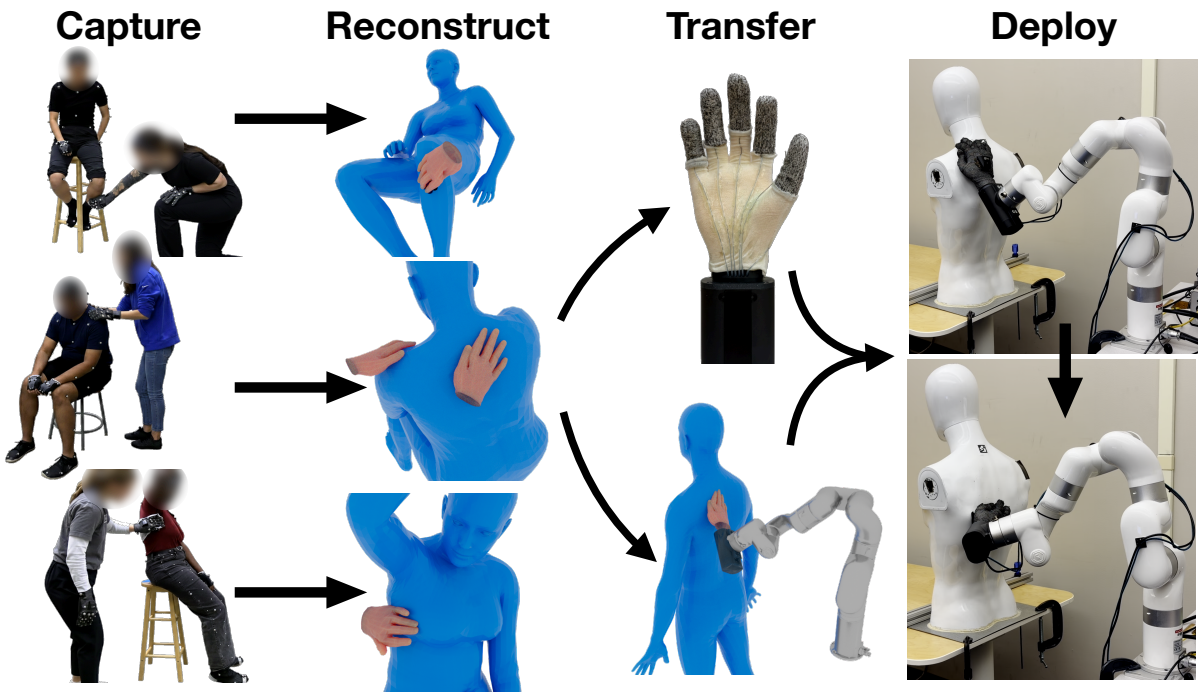

Figure 10.1: Our contributions in this chapter outline how to we can utilize contact areas across the entire development stack to translate human demonstration into fully operational robotic systems.

10.2 Approach

Figure 10.2 provides an overview of our capture, reconstruction, and transfer pipeline. We have already covered details of the capture and reconstruction portions in Section 9.2, and will therefore use this chapter to discuss system construction and demonstration transfer details.

10.2.1 Conceptual Design

Figure 10.3 illustrates our intended deployment cell. Because our system is an early proof of concept, we use a mannequin instead of real human subjects. The arm is positioned such that the workspace is able to cover the entire back of the mannequin. We also limit our experiments to transfers of back bathing demonstrations due to equipment constraints and setup overhead.

10.2.2 Design and Control of a Tendon-Driven Soft Hand

We select the anthropomorphic DexKit platform [King et al. 2025] — a foam cast hand fabricated directly from the artist-designed digital asset in Figure 6.7 — for demonstration transfer primarily due to its compliance and secondarily for its customizability. However, because the hand is fully soft with potentially infinite DOFs and unknown kinematics, three problems must be solved to use it effectively: (1) determining a tendon routing capable of achieving desired poses, (2) determining motor positions to actuate the tendons accordingly, and (3) building an effective control space. The combination of shape, motion, and contact offered by our reconstructions

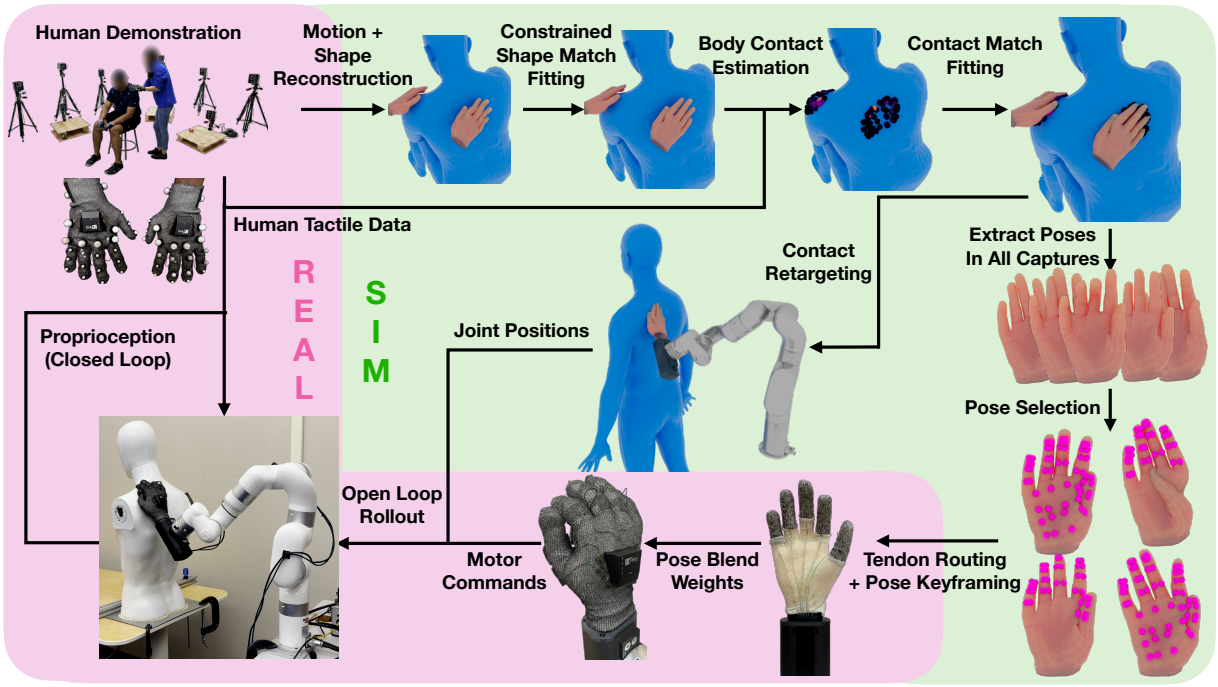

Figure 10.2: Overview of our pipeline. Starting with captures of motion and tactile forces from human bathing demonstrations, we digitally reconstruct time-synchronized data across shape, movement, and contact force with high accuracy. We then utilize these reconstructions to both design and control an anthropomorphic soft hand, as well as retarget results to a robot arm. We demonstrate the final system's efficacy by replicating bathing tasks on a mannequin in the real world.

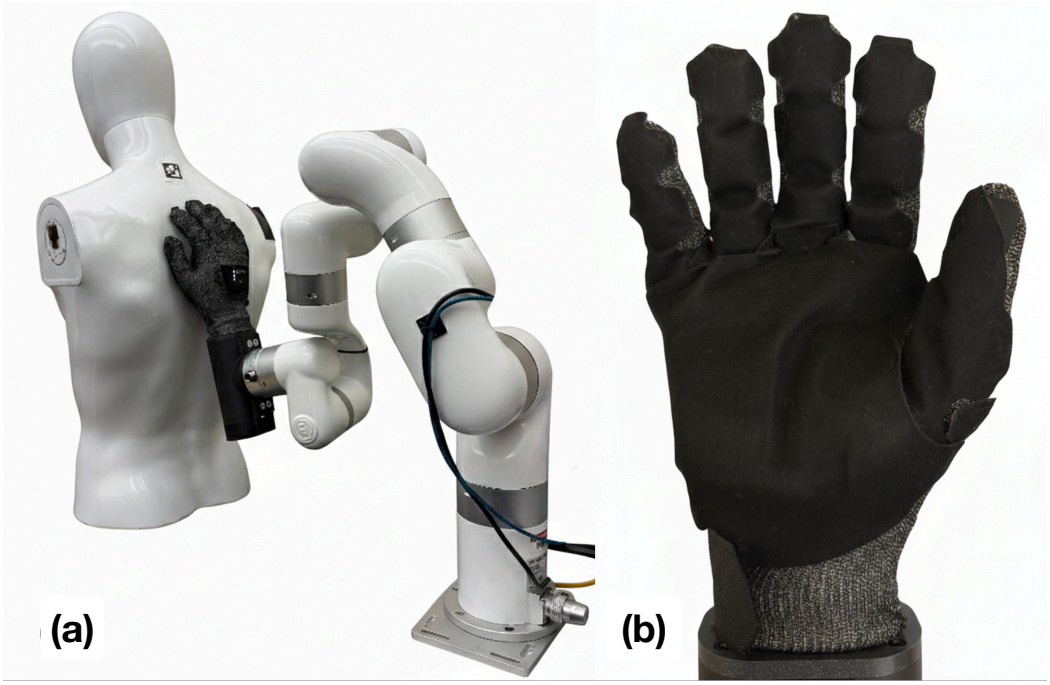

Figure 10.3: (a) The robot-assisted bathing setup, consisting of a robotic arm [UFACTORY 2026], DexKit robotic hand [King et al. 2025], and mannequin. (b) The DexKit hand is outfitted with a tactile sensing glove [Pressure Profile Systems 2026] with grip material [3M™ 2026] to increase friction.

helps solve each of these problems.

Starting from a dataset of human hand poses P, we first select a strictly smaller subset of candidates $S \subset P$ that sufficiently covers the full space. We do so by solving an NP-hard maximum diversity problem [Kuo et al. 1993] via greedy approximation [Ghosh 1996] and analyze the resulting loss curve to select the optimal number $|S|$. We then manually select a subset $S' \subset S$ that can be physically mimicked on DexKit under a given motor budget and commit to a tendon routing capable of generating the mimicked poses.

Next, we determine motor commands $\boldsymbol{\phi}$. As shown in Figure 10.4, calculating motor commands that adapt to contact-induced pose corrections depends on identifying hand contact regions and accurately simulating those interactions. We can directly obtain these contact regions from the dataset, and then retarget the tactile pad locations from the original human hand to the Dexkit hand using the atlas-based approach introduced in Chapter 6, and specifically the axial curve annotations in Figure 6.9, to figure out where to simulate contact.

We construct two matrices: $\boldsymbol{M}_{S'}$, where each column corresponds to a pose $s \in S'$, and $\boldsymbol{M}_{\boldsymbol{\phi}(S')}$, where each column consists of the manually determined motor positions $\boldsymbol{\phi}(s)$ required to generate that pose. For a given pose $p \in P$, we solve the following constrained least-squares QP to obtain normalized linear blend weights $\boldsymbol{\beta}^*$:

$$\begin{aligned} \boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta}} \quad & (\boldsymbol{M}_{S'}\boldsymbol{\beta} - p)^T(\boldsymbol{M}_{S'}\boldsymbol{\beta} - p) \\ \text{s.t.} \quad & 0 \leq \boldsymbol{\beta} \leq 1 \end{aligned} \tag{10.1}$$

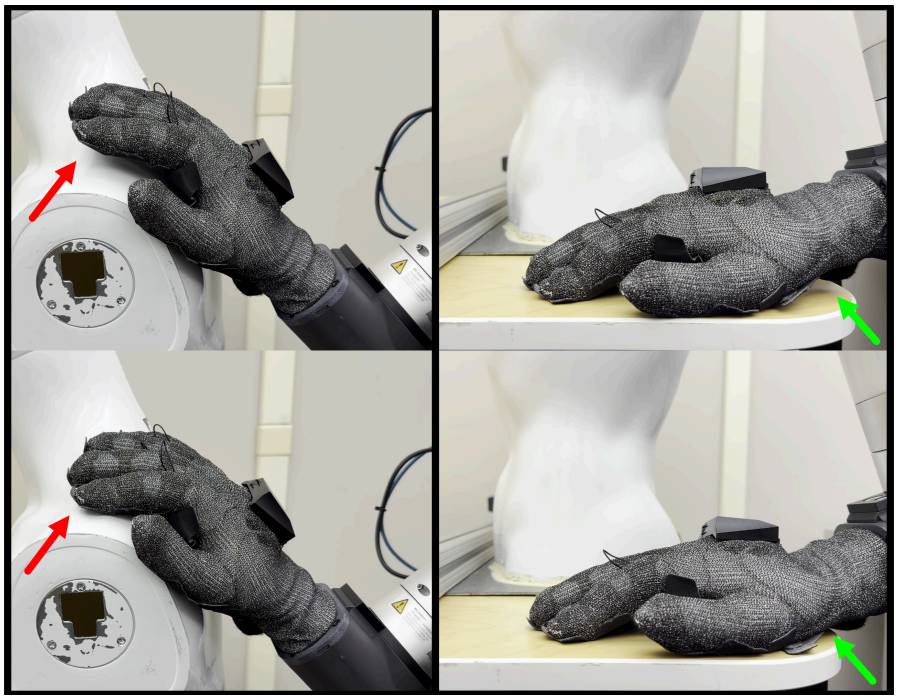

Figure 10.4: Visualizations of improved surface contact following hand pose correction, comparing the original (*top*) and corrected (*bottom*) states. In the first example (*left*), the red arrows highlight a significant increase in finger contact. In the second (*right*), the green arrows indicate the hand flattening against the table as the heel moves closer to the surface.

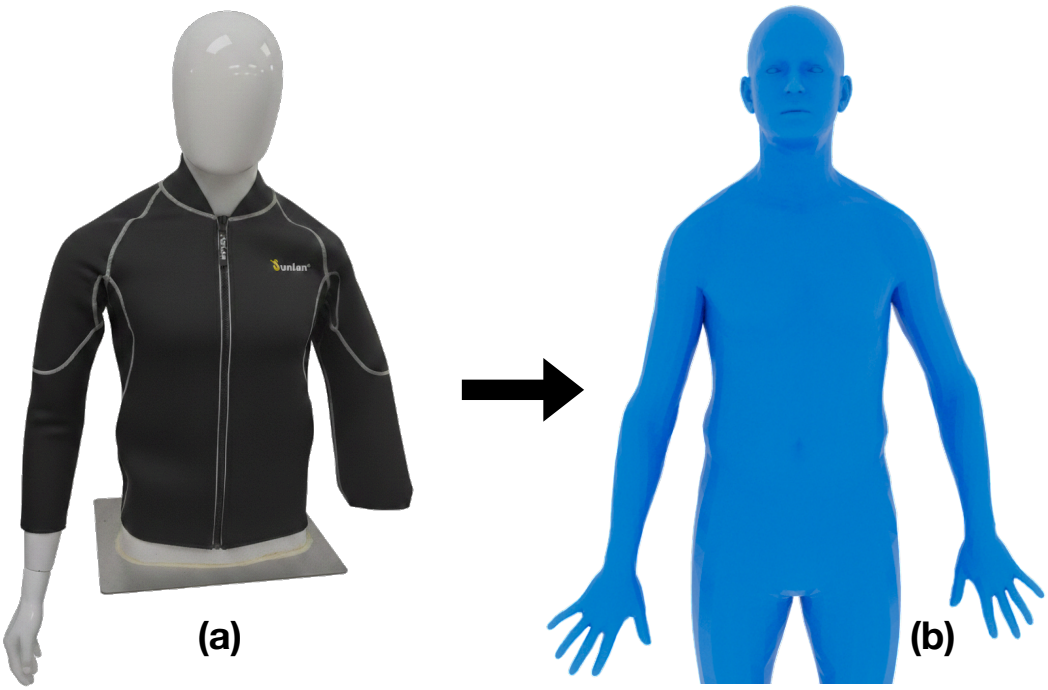

Figure 10.5: Illustrations of the (a) mannequin input image and (b) extracted SMPL-X proxy mesh from TRAM [Wang et al. 2024].

We then compute the corresponding motor positions from the optimal blend weights as $\boldsymbol{\phi} = \boldsymbol{M}_{\boldsymbol{\phi}(S')}\boldsymbol{\beta}^*$.

10.2.3 Arm Motion Retargeting and Control

We mount the DexKit hand onto an xArm 7 [UFACTORY 2026]. We deploy pre-computed motor trajectories $\boldsymbol{\phi}$ directly to the hand, but must retarget the human hand wrist trajectories to obtain arm joint angles $\boldsymbol{\psi}$. Unfortunately, we found that the commonly used strategy of using human wrist positions and orientations as end-effector targets [Sivakumar et al. 2022] did not work well due to the arm's restricted workspace and the fact that hand trajectories were highly dependent on each subject's body shape and reactive motion.

Instead, we compute $\boldsymbol{\psi}$ using retargeted body contacts. Specifically, we use the TRAM video pose estimator [Wang et al. 2024] to fit SMPL-X parameters to the mannequin. Figure 10.5 depicts the mannequin image and TRAM-reconstructed SMPL-X body shape, averaged across 10 frames of video. Notably, we found that attaching at least one arm and placing clothing on the mannequin was necessary because the model otherwise failed to identify the structure as a human. We only utilize the baseline body shape parameters, ignoring blend shape corrections and root pose estimates.

We then directly roll out contact trajectories computed on the source subject. Because contact trajectories are stored in barycentric coordinates, the retargeting operation is agnostic to the target body's shape or pose parameters. We can thus always perform the rollout regardless of any shape or pose divergences between the source and target domain. The complete arm-

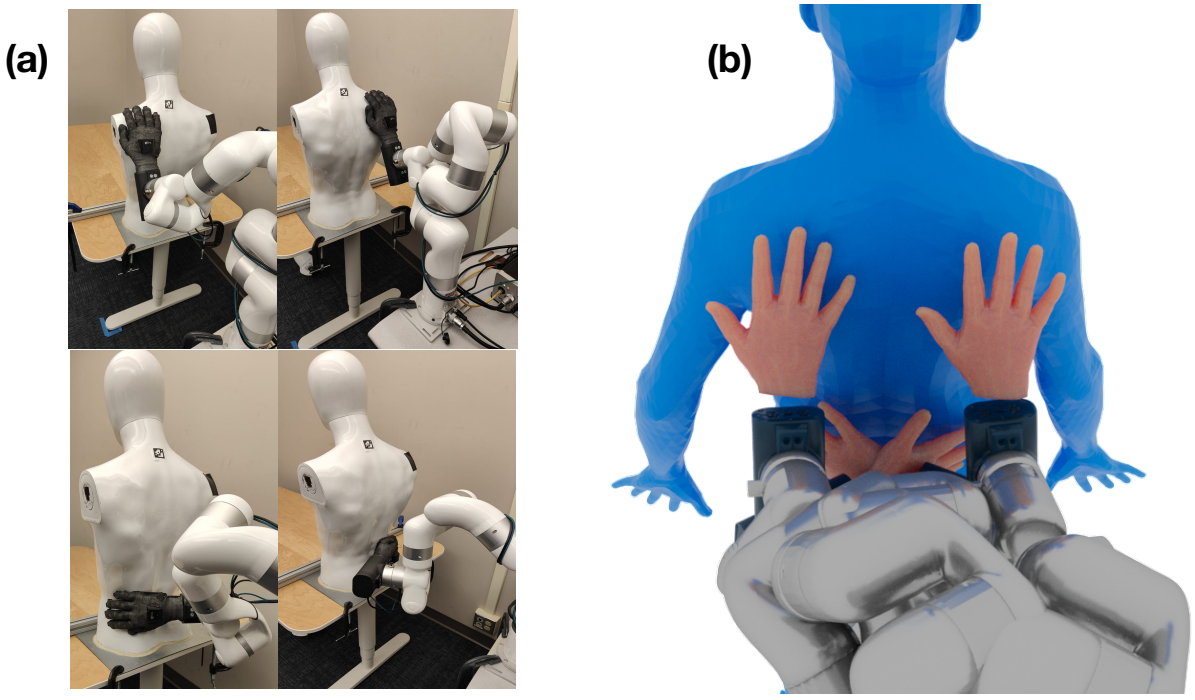

Figure 10.6: (a) xArm calibration poses collected in the real world are (b) superimposed for mannequin positioning in simulation.

hand trajectory can then be computed in simulation using the motion synthesis framework in Chapter 6 via retargeted body contact trajectories and Dexkit hand tactile pad estimates. However, in practice, we found that an alternate strategy could generate similar arm trajectories: rather than model the complete system, we could generate a cheap and effective proxy by replacing the Dexkit hand end effector with the source MANO hand, removing all wrist DOFs, copying Θ^* from the original motion, and freezing the hand into those poses. Because the hand poses are fixed, the problem reduces to a 7-DOF solve and is thus much faster to compute.

However, because ψ is computed in simulation, we must perform a real world alignment calibration. To do so, we manually adjust the xArm into four distinct poses to estimate the "boundaries" of the mannequins back, which include the left and right shoulders as well as the left and right hip segments. We then manually align the mannequin proxy in simulation relative to the corresponding xArm positions, and note that we intentionally offset the proxy mesh slightly further from the hands. This accounts for shape discrepancies and ensures the arm biases toward overreaching rather than falling short. Figure 10.6 illustrates the process. The resulting trajectory $[\{\phi_0, \psi_0\}, \{\phi_1, \psi_1\}, ..., \{\phi_T, \psi_T\}]$ can then be deployed open-loop on the real system post-calibration.

However, the open-loop controller frequently breaks, or alternatively applies too much, contact with the mannequin due to the gap between the real geometry and its fitted SMPL-X parameters in simulation. We create a closed-loop system by sensorizing the DexKit hand with the same tactile glove used in data capture, and utilize contact and force information from the capture to create a lower-level reactive controller. Specifically, at each time step, we derive a normalized corrective direction, N_C, for the end-effector by averaging the distance vectors from

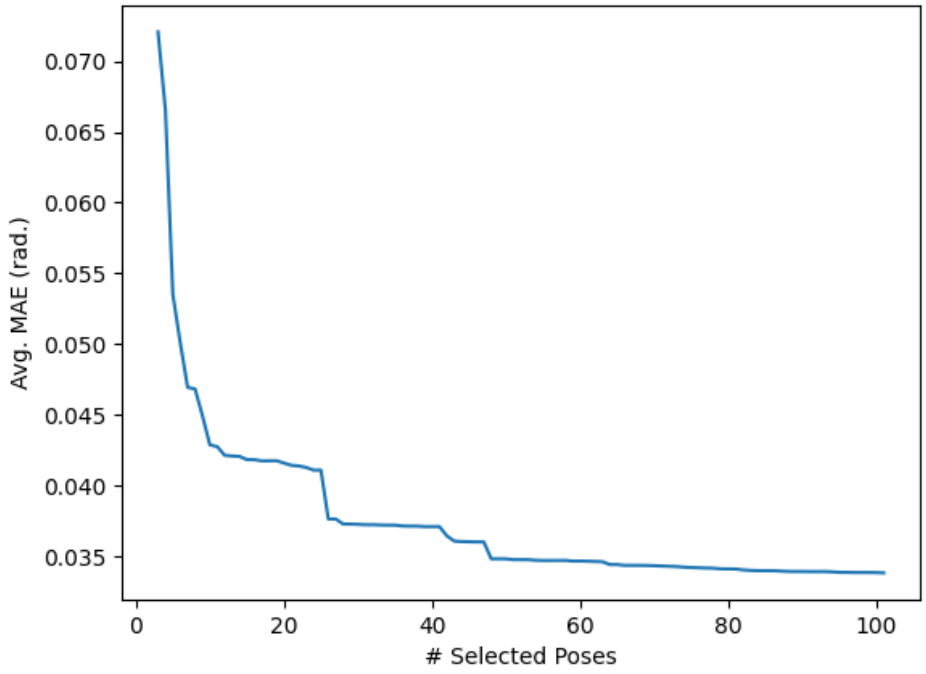

Figure 10.7: Complete MAE loss curve over the parameter sweep of $|S|$.

the active set of simulated hand and body contacts. We convert $\boldsymbol{\psi}$ to EE positions $FK(\boldsymbol{\psi})$ rolled out at a baseline frequency and apply corrective position updates at a fixed timestep Δt, creating inner loop EE position $FK(\boldsymbol{\psi}) + N_C \Delta t$. Corrective movements are applied toward the body until the online sum of tactile pressures matches the original measurement, then reversed if that value is exceeded. Note that while our decision variable is currently based on the integrated force across all taxels, more fine-grained localization across corresponding retargeted body and Dexkit hand contact distributions are available and can be utilized in alternate end-effector control strategies.

10.3 Experiments and Results

10.3.1 Soft Hand Design and Control

We perform and evaluate DexKit hand design using a subset of the dataset focusing on back bathing. We collect poses and contacts from the right hand in all subject captures to build P, and perform a sweep over the number of candidates in the diversity subset S to determine the optimal sample size $|S|$. $|S|$ candidates are evaluated via MAE differentials between the original poses and their respective projections in S obtained by solving Eq. 10.1. Figure 10.7 illustrates the complete loss curve over a sweep of $|S| = 1$ to $|S| = 100$. The size of all right hand back bathing poses $|P| = 9491$. While there are further dips in the curve at values $|S| > 100$, and it is possible to achieve zero error by choosing $|S| = |P|$, we found that many poses were qualitatively repetitive. Errors of re-projection MAE losses computed using the finally constructed $M_{S'}$ were normally distributed with parameters $(\mu, \sigma) = (0.04477, 0.01928)$. We select $|S| = 50$ based on

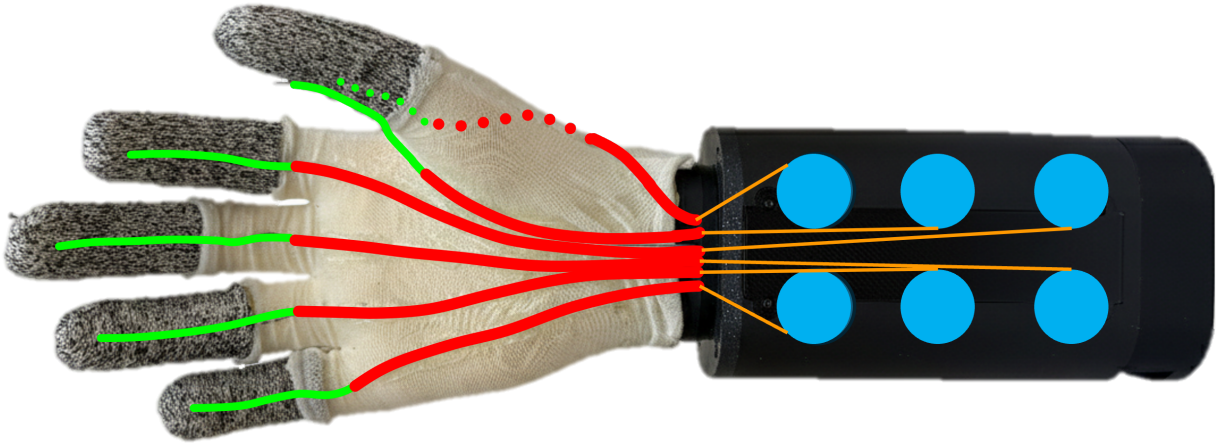

Figure 10.8: The DexKit hand is comprised of a cast foam hand with a textile "skin". Tendons are sewn into the skin (green), and routed along the palm through captive PTFE sheaths (red). The hand shown has six tendons, five flexors on the fingers and thumb, as well as an adductor on the thumb which is routed along the back (shown with dotted lines). The hand interfaces with a motor housing (black) which contains an array of six servo motors. Inside the motor housing the free tendons (orange) route to pulleys (blue) on the motors which reel in the tendons to actuate the hand.

observing diminishing returns in the loss curve for larger values of $|S|$, and then choose $|S'| = 20$ based on a budget of six motors. We only consider poses in contact as candidates for S and S' and store all active taxels for each selected pose. $\boldsymbol{\beta}^*$ weights for all trajectory poses are computed by solving Eq. 10.1 using the resulting $\boldsymbol{M}_{S'}$.

Figure 10.8 provides schematics of the finally selected Dexkit motor and tendon configuration. We select a flexor-only configuration for the index, middle, ring, and pinky fingers, and an adduction routing for the thumb to pull in closer to the palm thenar region. We found the adduction routing more useful than a flexure routing because the flexed thumb interfered with the palm making contact with the mannequin during wiping the relatively flat back regions. The total system mass is roughly 0.25 kg, which is well within the xArm payload limit.

We next build $\boldsymbol{M}_{\boldsymbol{\phi}(S')}$ and compute $\boldsymbol{\phi}$. We manually determine motor positions for each candidate pose without contact, and then adjust positions while pressing the hand at the regions of contact designated by the active taxels. Figure 10.9 illustrates several poses used during calibration.

10.3.2 Real World Deployment

Figure 10.10 illustrates the xArm positions computed following a retargeted contact trajectory rollout, as well as deployment of that trajectory on our real world platform. $\boldsymbol{\psi}$ and $\boldsymbol{\phi}$ commands are deployed together in the control loop, and the original human demonstration is slowed from 120 to 6 Hz (20X).

We also perform a pressure comparison between the recorded tactile data of the original human demonstration and the robot deployment. Figure 10.11 compares the recorded glove

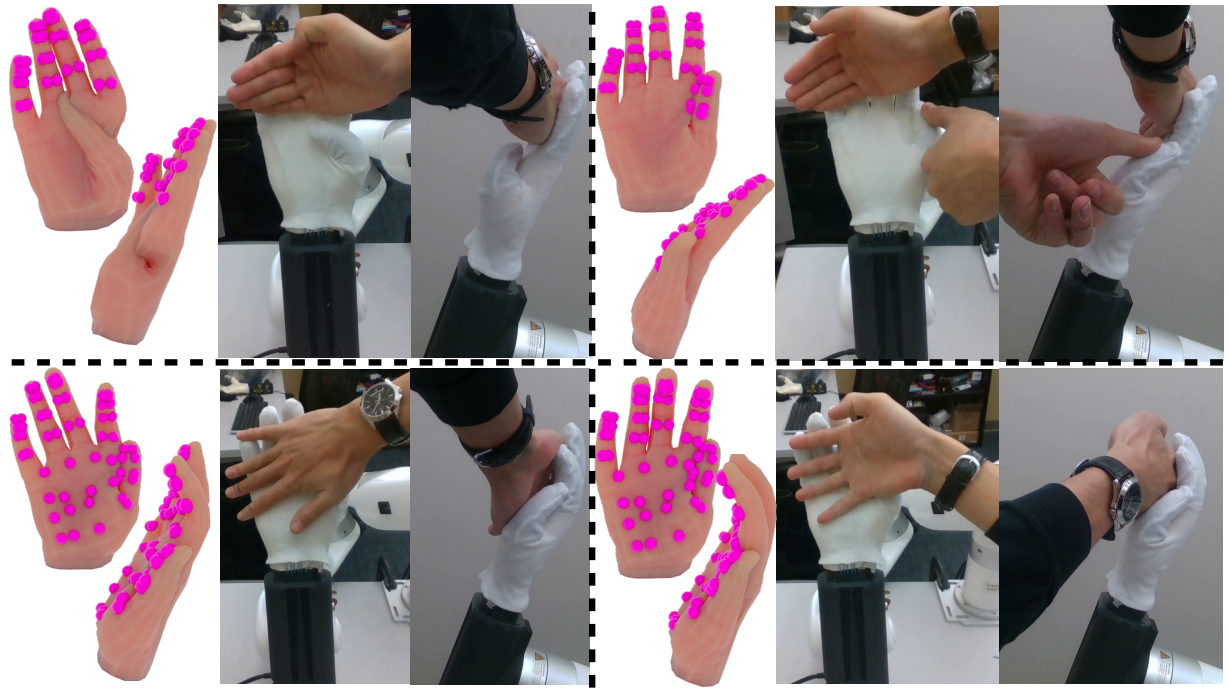

Figure 10.9: Sample poses used to build $M_{\phi(S')}$. Active contacts in each pose (pink) are simulated by pressing against the hand.

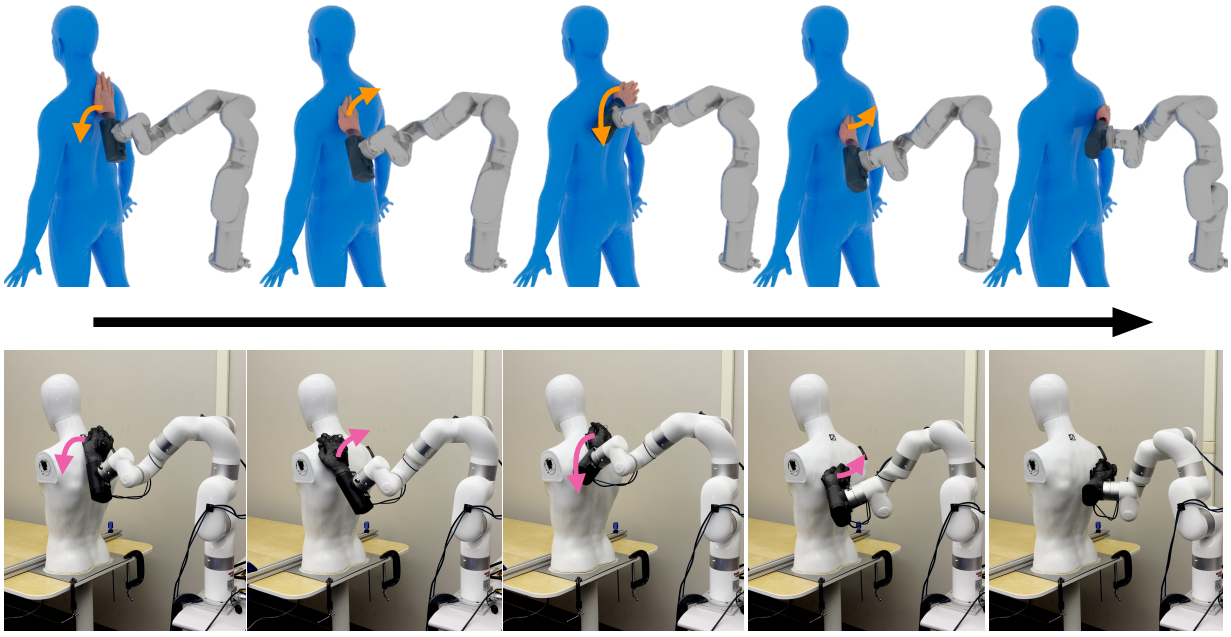

Figure 10.10: Film strip of back bathing demonstration retargeted to the xArm 7 (top) in simulation using the clinician's shape-fitted hand as the end effector. The computed joint positions and pre-computed hand motor commands are (bottom) deployed on the real world setup. Orange and pink arrows illustrate subsequent movement directions.



Figure 10.11: Overlaid plots of the (blue) original human, (orange) robot open loop, and (green) robot closed loop tactile glove pressures during a back bathing task. We provide (a) absolute sums, (b) normalized sums, and (c) median intensities calculated across the full taxel array per time step. Note that the spike in pressure at the start of closed loop data collection is due to initialization noise rather than a consequence of trajectory execution.

pressure signals across open-loop rollout, closed-loop rollout, and the original human back bathing demonstration. We make several observations in the data.

As expected, the integrated tactile pressure for the open-loop rollouts is significantly higher than both the human demonstration and closed-loop results for the majority of the trajectory. These elevated pressure levels indicate unsafe operating conditions, a conclusion further supported by visible mannequin vibrations in the supplementary video. Closed-loop control substantially curtails such issues, as quantitatively evidenced in the plots as well as qualitatively by minimal mannequin vibrations in the video. Closed-loop pressures also reasonably track the original human demonstration, particularly when viewed on a normalized scale.

However, the closed-loop controller deviates from the human demonstration in absolute pressure and especially spatial distribution, as indicated by the comparison of median values. The distributional misalignment is an inherent trade-off of the controller design, as optimizing for an aggregate pressure sum does not necessitate parity in the underlying taxel-wise distribution. Furthermore, the attenuated dynamic range is a byproduct of necessary safety constraints in real world deployment. Specifically, we found that targeting a strict absolute pressure induces significant arm oscillations and mitigated this behavior by implementing tolerance bounds around the target instead. We also identified a failure mode where friction-induced finger curling during wiping motions could lead to mechanical damage. If curling occurs while the controller commands an inward corrective force, the system may drive the arm further into the mannequin in a futile attempt to reach a pressure target that the compromised finger geometry cannot achieve. To mitigate this, we implemented a temporal safety timeout for unreached pressure setpoints. Interestingly, this failure mode was absent in open-loop trials; because the controller advances through the trajectory regardless of feedback, the hand's passive compliance allowed fingers to recover from compression during lateral movements.

While reasonably effective, we conjecture that our closed-loop strategy can be significantly improved by incorporating distributed, tendon-level differentiable force control and dedicated extensor tendons. Such additions would enable the hand itself to respond to local pressure misalignments and escape the identified failure mode more reliably. Both avenues are promising real-world-focused future research directions for this work.

10.4 Discussion

Several key insights emerged during the system construction and policy deployment process. We discuss several lessons learned and current limitations below.

10.4.1 Lessons Learned

Contact was and needed to be extensively utilized to arrive at our final system. We first had to capture contact areas on the hands of the clinician and reconstruct them on the shape-fitted bodies of human subjects as described in Section 9.2, transfer them to a different body, retarget distributions from the human hand to the robot hand to perform calibration, align the new body and compute the trajectory of a limited workspace robot arm, and finally utilize forces distributed across the contact areas to close the control loop. We relied heavily upon the constraints inherently provided by contact areas to reliably compute solutions to the hand

and arm as well as the forces distributed over them to serve as targets for the final impedance controller. We also discovered shortcuts, problems that could be directly addressed via our area-based representations, and insights into how to refine the system moving forward.

Maintaining consistency in mesh connectivity is a useful way to sidestep complex contact retargeting. Because the same exact vertices from SMPL-X are used for human subjects and mannequin, no intermediate representation necessary — we can simply replay the barycentric coordinates and expect the evolutions of the contacts to reasonably maintain the area structure of the hand taxels. But this is not always possible; for example, no such mapping exists between the human MANO and Dexkit geometries. The shape matching approach introduced in Chapter 6, and by extension the AE model introduced in Chapter 4, offered a practical solution to this retargeting problem.

We found that compliance was *of paramount importance* in arm-mounted bathing systems. Sustained, dense-contact interactions permit little room for trajectory mismatch errors, which are especially hard to avoid during sim2real rollouts. Although the Dexkit hand's material compliance mitigated these issues during our experiments, we contend that incorporating compliance into arm control is essential for safe and effective interaction with human subjects.

Despite its promise from a data-driven standpoint, we ultimately found that a flexor-only finger tendon configuration was insufficient to capture all necessary hand motions during rollouts. Passive compliance did not always allow the hand to escape friction-induced finger curling during wiping motions. Including dedicated extensor tendons would help mitigate such issues, but would be difficult to do under limited motor budgets.

10.4.2 Limitations

There are three fundamental issues limiting our current system: (1) capturing and reconstructing forces on the hands and body that accurately represent sensations felt by the subject, (2) enabling the system to perceive real human subjects during deployment rather than only a static mannequin, and (3) reproducing the sensation originally imparted by the human demonstrator. We discuss the first limitation in Section 9.2.5 and detail the remaining limitations below.

Perceiving the state of human subjects is a critical component of real world deployment; however, detailed real time tracking of human subjects is an unsolved computer vision problem. Specifically, no existing method achieves fully online human pose and shape estimation while maintaining the sub-frame latency required for live interaction. Our method to fit SMPL-X parameters to the mannequin [Wang et al. 2024], for example, is a state of the art human shape and pose estimator; however, it takes minutes to process a video sequence of several hundred frames and thus can only be run offline.

Additionally, our strategy of direct contact area rollouts between the human subject and mannequin proxy, and more generally between different subjects, implicitly depends on both domains exhibiting similar SMPL-X parameters. We anticipate that large divergences between parameters (e.g. extremely thin vs. wide body shapes) would distort the contact trajectories by causing them to excessively stretch or contract. One alternative possibility would be to use axis annotations to shape match bodies just as we did with the hands in Figure 6.9; however, the choice of where to place axes and how to determine the appropriate λ_A and λ_S parameters is less clear. Suboptimal annotations, for example, could generate unexpected jump discontinuities

in trajectory rollouts from large changes in the underlying geoedesic path traces. Additionally, the stroke patterns themselves could differ dramatically based on the subject's body proportions and may not be possible to transform purely geometrically to begin with.

Finally, our current closed-loop strategy is highly limited by its use of tactile pressure sums rather than the distribution over individual taxels. The primary barrier to distributed force control is the significant discrepancy between human and Dexkit hand responses when subjected to similar pressure stimuli. While existing methods mitigate this issue through careful placement of sensors, selection of sensing modality, and calibration to align readings between similar human and robot trajectories [Yin et al. 2025], such approaches tend to be brittle and require significant engineering overhead. There remains an opportunity to more directly model the inherent discrepancies in hand materials. *Retargeting force distributions* to different hand geometries and material properties, for example, is a promising direction of future research.

10.5 Summary

We have presented an end-to-end framework for conceptualizing, designing, building, and controlling a complete robot bathing system starting from human demonstrations. Importantly, we have shown that utilizing contact regions as a key processing primitive effectively addresses many pain points in the construction and transfer pipelines. We hope that the contributions will spur further research in physical human-robot-interaction systems and ultimately make progress towards bringing useful and reliable robots into high-value clinical settings.



Chapter 11

Conclusion

The research presented in this thesis has introduced several key advancements towards establishing area-based contact models as first-class primitives, which we hope will serve as a as a strong foundation for encouraging future research in this direction.

11.1 Summary of Contributions

Our contributions include:

- **New Representations:** We introduced three novel models for contact areas, which include the Boundary, Single-Point Embedded (SPE), and Axis-Embedded (AE) models. We also defined and introduced operators for these representations, which collectively enable fundamental transformations such as translation, rotation, bending, interpolation, and transfer. Furthermore, by utilizing ideas from discrete differential geometry, we have enabled all of these representations and operators to work on realistic shapes rather than only primitives.

- **Artist Tools for Grasp Drafting:** These tools enabled intuitive modeling of contact-rich grasps from scratch by using the AE contact area model as the core editing primitive. Combined, these tools enabled a workflow that was capable of creating not just high quality, art-directable results, but in many cases results that looked more visually appealing in comparison to those created with existing fully manual joint control techniques. We also showed that these tools could be used in new applications such as rapid prototyping of hand designs and ground-truth data annotation at internet scale.

- **Kinematic Motion Retargeting for Contact-Rich Anthropomorphic Manipulations:** We extended techniques from grasping to full manipulation trajectories by utilizing contact areas as the primary retargeting medium. To do so, we introduced a novel atlas-based shape matching method that proved simple, intuitive, and robust in practice. We then further highlight the value of contact areas by showing that the motion synthesis pipeline could be trivialized to basic grasp solving and stitching, and that the retargeted trajectory generated using this approach could be extended to full physics simulation through local online adaptive sampling. Finally, we showed that the techinque was generic and could easily be extended to arbitrary objects and bi-manual manipulations.

- **New Real-World Capture and Reconstruction Techniques:** We introduced two new techniques for capturing and reconstructing contact areas from the real world. Our first contribution built on the thermal imaging literature [Brahmbhatt et al. 2019a] by offering a low-cost RGB alternative via thermochromic paint. Our second contribution enabled joint reconstruction of contact area time series data consisting of both location and forces. We then demonstrated the mutual benefits of both data streams by showing how they could be combined to produce high fidelity reconstructions of sustained contact-rich human-human interactive tasks in the application context of assisted bathing.

- **Full Stack System Construction from Contact:** Finally, we showed how to construct a proof-of-concept robotic assisted bathing system from the ground up by utilizing contact areas as a core processing primitive across the full development stack of data, conceptualization, hardware, and software. To the best of our knowledge, this system is the first of its kind within the physically assistive human-robot interaction domain, capable of transferring actual bathing demonstrations performed by real clinicians on human subjects.

11.2 Open Problems and Future Directions

While we have made considerable progress towards making contact areas viable as manipulation primitives, our contributions are largely only first steps. We are still very far away from replicating the complete set of existing capabilities offered by point contact models. While there are far more problems to solve, we offer three open problems that we believe will significantly help bridge this divide:

- **Creating a True Continuous Area Contact Model:** While useful, all of the area contact we introduced in this thesis are discrete in that they still computationally treat areas as groups of points. But there are many useful applications which can be derived from modeling areas as true continuous distributions that jointly evolve on both contacting surfaces, as well as time-integrable variables in relation to the manipulator dynamics ODEs. The compact parameter space of continuous distributions, for example, would make *learning* such parameters from large-scale data tractable and *in a way that does not tie parameters to a specific geometry discretization* (e.g. mesh, point cloud, etc.).

- **Area-Based Contact Planners:** Planning is a notoriously challenging problem because problem complexity scales exponentially in the number of search variables. This would make planning manipulation trajectories with our discrete area models computationally intractable, especially at the high frequencies required to make planners viable on real robotic systems. A continuous area model would substantially reduce the search space dimensionality, and by doing so make it possible to expand traditional planners such as RRT-Connect [Kuffner & LaValle 2000] to area-based manipulation trajectories.

- **Standardizing Contact Perception:** Tactile sensing has rapidly become one of the most exciting frontiers in manipulation research. But working with tactile data currently is highly specific to each sensor, whether that requires building a simulated model, collecting

real world demonstrations and learning a latent space, or figuring out which tasks benefit the most from the information gain. But all tactile sensors ultimately are intended to help estimate contact state, and areas have the potential to standardize the representation across current and future sensor variations. It would be interesting to use that standardization to evaluate questions such as which sensors are most useful for a task, where they should be placed, and how we can build automated calibration techniques.

11.3 Final Remarks

Contact in the real world is messy, but critical to understand and exploit. While we have known for decades that real contact is area-based, there have been relatively few efforts to actually make area-based contact models viable as first class primitives. This thesis has taken concrete steps towards addressing this gap, and has demonstrated how useful area-based contact models can be across a range of applications ranging from digital grasp solvers to full stack robotic system development. While there is still a long way to go in enabling area-based models to become as widely utilized as point-, line-, or plane-based models are today, we hope that the ideas presented in this thesis will spur future research in this direction.



Bibliography

3M™ (2026). *Gripping Material TB641*. https://www.3m.com/3M/en_US/p/d/b40069682/. Accessed: 2026-01-26.

Adeniji, A., Chen, Z., Liu, V., Pattabiraman, V., Bhirangi, R., Haldar, S., Abbeel, P., and Pinto, L. (2025). "Feel the force: contact-driven learning from humans". *arXiv:2506.01944*.

Arunachalam, S. P., Güzey, I., Chintala, S., and Pinto, L. (2023). "Holo-dex: teaching dexterity with immersive mixed reality". *IEEE International Conference on Robotics and Automation*, pp. 5962–5969.

Bauer, D., Bauer, C., Lakshmipathy, A., Shu, R., and Pollard, N. S. (2022). "Towards very low-cost iterative prototyping for fully printable dexterous soft robotic hands". *IEEE International Conference on Soft Robotics*, pp. 490–497.

Baxter, L. K. (1997). "Capacitive sensors". *IEEE Design and Applications*.

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

Bishop, A. L., Zhang, J. Z., Gurumurthy, S., Tracy, K., and Manchester, Z. (2024). "Relu-qp: a gpu-accelerated quadratic programming solver for model-predictive control". *IEEE International Conference on Robotics and Automation*, pp. 13285–13292.

Brahmbhatt, S., Ham, C., Kemp, C. C., and Hays, J. (2019a). "Contactdb: analyzing and predicting grasp contact via thermal imaging". *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8709–8719.

Brahmbhatt, S., Handa, A., Hays, J., and Fox, D. (2019b). "Contactgrasp: functional multi-finger grasp synthesis from contact". *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2386–2393.

Brahmbhatt, S., Tang, C., Twigg, C. D., Kemp, C. C., and Hays, J. (2020). "ContactPose: a dataset of grasps with object contact and hand pose". *European Conference on Computer Vision*, pp. 361–378.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al. (2015). "ShapeNet: an information-rich 3d model repository". *arXiv:1512.03012*.

Chen, J., Chen, Y., Zhang, J., and Wang, H. (2024a). "Task-oriented dexterous grasp synthesis via differentiable grasp wrench boundary estimator". *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5281–5288.

Chen, S., Bohg, J., and Liu, C. K. (2024b). "Springgrasp: an optimization pipeline for robust and compliant dexterous pre-grasp synthesis". *Robotics: Science and Systems*.

Chen, T., Xu, J., and Agrawal, P. (2022). "A system for general in-hand object re-orientation". *Conference on robot learning*. PMLR, pp. 297–307.

Cheng, X., Huang, E., Hou, Y., and Mason, M. T. (2022). "Contact mode guided motion planning for quasidynamic dexterous manipulation in 3d". *IEEE International Conference on Robotics and Automation*, pp. 2730–2736.

Cheng, X., Patil, S., Temel, Z., Kroemer, O., and Mason, M. T. (2023). "Enhancing dexterity in robotic manipulation via hierarchical contact exploration". *IEEE Robotics and Automation Letters* 9.1, pp. 390–397.

Crane, K. (2018). "Discrete differential geometry: an applied introduction". *Notices of the AMS, Communication* 1153.2.

Cseke, A., Tripathi, S., Dwivedi, S. K., Lakshmipathy, A. S., Chatterjee, A., Black, M. J., and Tzionas, A. (2025). "Pico: reconstructing 3d people in contact with objects". *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1783–1794.

Czuba, L. R., Sommerich, C. M., and Lavender, S. A. (2012). "Ergonomic and safety risk factors in home health care: exploration and assessment of alternative interventions". *Work* 42.3, pp. 341–353.

Damen, D., Doughty, H., Farinella, G. M., Furnari, A., Kazakos, E., Ma, J., Moltisanti, D., Munro, J., Perrett, T., Price, W., et al. (2022). "Rescaling egocentric vision: collection, pipeline and challenges for epic-kitchens-100". *International Journal of Computer Vision* 130.1, pp. 33–55.

Darragh, A. R., Sommerich, C. M., Lavender, S. A., Tanner, K. J., Vogel, K., and Campo, M. (2015). "Musculoskeletal discomfort, physical demand, and caregiving activities in informal caregivers". *Journal of Applied Gerontology* 34.6, pp. 734–760.

Dasari, S., Gupta, A., and Kumar, V. (2023). "Learning dexterous manipulation from exemplar object trajectories and pre-grasps". *IEEE International Conference on Robotics and Automation*, pp. 3889–3896.

Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., and Farhadi, A. (2023). "Objaverse: A universe of annotated 3D objects". *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13142–13153.

Dijkstra, E. W. (1959). "A note on two problems in connexion with graphs". *Numerische mathematik* 1.1, pp. 269–271.

Ding, D., Lee, Y. H., and Wang, S. (2001). "Computation of 3-d form-closure grasps". *IEEE Transactions on Robotics and Automation* 17.4, pp. 515–522.

Dunlop, D. D., Hughes, S. L., and Manheim, L. M. (1997). "Disability in activities of daily living: patterns of change and a hierarchy of disability." *American journal of public health* 87.3, pp. 378–383.

Eberly, D. (2005). *Least squares fitting of data with b-spline curves*. URL: https://www.geometrictools.com/Documentation/BSplineCurveLeastSquaresFit.pdf.

Elandt, R., Drumwright, E., Sherman, M., and Ruina, A. (2019). "A pressure field model for fast, robust approximation of net contact force and moment between nominally rigid objects". *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 8238–8245.

Erickson, Z., Clever, H. M., Gangaram, V., Turk, G., Liu, C. K., and Kemp, C. C. (2019). "Multidimensional capacitive sensing for robot-assisted dressing and bathing". *IEEE International Conference on Rehabilitation Robotics*, pp. 224–231.

Fan, Z., Taheri, O., Tzionas, D., Kocabas, M., Kaufmann, M., Black, M. J., and Hilliges, O. (2023). "ARCTIC: a dataset for dexterous bimanual hand-object manipulation". *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12943–12954.

Fang, H. S., Romero, B., Xie, Y., Hu, A., Huang, B. R., Alvarez, J., Kim, M., Margolis, G., Anbarasu, K., Tomizuka, M., et al. (2025). "Dexop: a device for robotic transfer of dexterous human manipulation". *arXiv:2509.04441*.

Ferrari, C. and Canny, J. (1992). "Planning optimal grasps". *IEEE International Conference on Robotics and Automation*. Vol. 3, pp. 2290–2295.

Gehre, A., Bronstein, M., Kobbelt, L., and Solomon, J. (2018). "Interactive curve constrained functional maps". *Computer Graphics Forum*. Vol. 37. 5, pp. 1–12.

Ghosh, J. B. (1996). "Computational aspects of the maximum diversity problem". *Operations Research Letters* 19.4, pp. 175–181.

Gill, T. M., Guo, A., and Allore, H. G. (2006). "The epidemiology of bathing disability in older persons". *Journal of the American Geriatrics Society* 54.10, pp. 1524–1530.

Grady, P., Tang, C., Twigg, C. D., Vo, M., Brahmbhatt, S., and Kemp, C. C. (2021). "Contactopt: optimizing contact to improve grasps". *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1471–1481.

Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., et al. (2022). "Ego4d: around the World in 3,000 Hours of Egocentric Video". *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18995–19012.

Guay, M., Cani, M. P., and Ronfard, R. (2013). "The line of action: an intuitive interface for expressive character posing". *ACM Transactions on Graphics* 32.6, pp. 1–8.

Handa, A., Wyk, K. V., Yang, W., Liang, J., Chao, Y. W., Wan, Q., Birchfield, S., Ratliff, N. D., and Fox, D. (2019). "Dexpilot: vision-based teleoperation of dexterous robotic hand-arm system". *IEEE International Conference on Robotics and Automation*, pp. 9164–9170.

Hang, K., Stork, J. A., Pollard, N. S., and Kragic, D. (2017). "A framework for optimal grasp contact planning". *IEEE Robotics and Automation Letters* 2.2, pp. 704–711.

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

Herholz, P., Haase, F., and Alexa, M. (2017). "Diffusion diagrams: voronoi cells and centroids from diffusion". *Computer Graphics Forum*. Vol. 36. 2, pp. 163–175.

Hou, Y., Jia, Z., and Mason, M. T. (2020). "Manipulation with shared grasping". *Robotics: Science and Systems*.

Howell, T., Gileadi, N., Tunyasuvunakool, S., Zakka, K., Erez, T., and Tassa, Y. (2022). "Predictive sampling: real-time behaviour synthesis with mujoco". *arXiv*. DOI: 10.48550/arXiv.2212.00541. URL: https://arxiv.org/abs/2212.00541.

Huang, I., Chow, D., and Bajcsy, R. (2022). "Soft tactile contour following for robot-assisted wiping and bathing". *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 7797–7802.

Humberston, B. and Pai, D. K. (2015). "Hands on: interactive animation of precision manipulation and contact". *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 63–72.

Jacobs, J. and Fröhlich, B. (2011). "A soft hand model for physically-based manipulation of virtual objects". *IEEE Virtual Reality Conference*, pp. 11–18.

Jain, S. and Liu, C. K. (2011). "Controlling physics-based characters using soft contacts". *SIGGRAPH Asia Conference Proceedings*, pp. 1–10.

Jiang, Y., Yu, M., Zhu, X., Tomizuka, M., and Li, X. (2024). "Contact-implicit model predictive control for dexterous in-hand manipulation: a long-horizon and robust approach". *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5260–5266.

Johnson, S. (2017). *The NLopt nonlinear-optimization package*. URL: http://ab-initio.mit.edu/nlopt.

Jost, J. (2008). *Riemannian geometry and geometric analysis*. Vol. 42005. Springer.

Kamakura, N., Matsuo, M., Ishii, H., Mitsuboshi, F., and Miura, Y. (1980). "Patterns of static prehension in normal hands". *The American journal of occupational therapy* 34.7, pp. 437–445.

Katz, S., Ford, A. B., Moskowitz, R. W., Jackson, B. A., and Jaffe, M. W. (1963). "Studies of illness in the aged: the index of adl: a standardized measure of biological and psychosocial function". *Journal of the American Medical Association* 185.12, pp. 914–919.

Kim, B. H., Oh, S. R., Yi, B. J., and Suh, I. H. (2001). "Optimal grasping based on non-dimensionalized performance indices". *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vol. 2, pp. 949–956.

Kim, J., Iwamoto, K., Kuffner, J. J., Ota, Y., and Pollard, N. S. (2013). "Physically based grasp quality evaluation under pose uncertainty". *IEEE Transactions on Robotics* 29.6, pp. 1424–1439.

King, J. P., Ahluwalia, H., Heredia-Marin, I. B., McClary, K., Zuo, E., Baratz, M., Colgate, J. E., Clark, J., Orta Martinez, M., and Pollard, N. S. (2025). "Dexkit: a hardware platform for multi-finger dexterous robotics". *International Symposium on Experimental Robotics*.

Kry, P. G. and Pai, D. K. (2003). "Continuous contact simulation for smooth surfaces". *ACM Transactions on Graphics* 22.1, pp. 106–129.

Kuffner, J. J. and LaValle, S. M. (2000). "Rrt-connect: an efficient approach to single-query path planning". *IEEE International Conference on Robotics and Automation*. Vol. 2, pp. 995–1001.

Kuo, C. C., Glover, R., and Dhir, K. S. (1993). "Analyzing and modeling the maximum diversity problem by zero-one programming". *Decision Sciences* 24.6, pp. 1171–1185.

Lakshmipathy, A., Bauer, D., Bauer, C., and Pollard, N. S. (2022). "Contact transfer: a direct, user-driven method for human to robot transfer of grasps and manipulations". *IEEE International Conference on Robotics and Automation*, pp. 6195–6201.

Lakshmipathy, A., Bauer, D., and Pollard, N. S. (2021). "Contact tracing: a low cost reconstruction framework for surface contact interpolation". *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5165–5172.

Lakshmipathy, A. S., Feng, N., Lee, Y. X., Mahler, M., and Pollard, N. S. (2023). "Contact edit: artist tools for intuitive modeling of hand-object interactions". *ACM Transactions on Graphics* 42.4.

Lakshmipathy, A. S., Hodgins, J. K., and Pollard, N. S. (2025). "Kinematic motion retargeting for contact-rich anthropomorphic manipulations". *ACM Transactions on Graphics* 44.2.

Lakshmipathy, A. S., King, J. P., Zuo, E., Satishkumar, R., Chen, H., Ichnowski, J., Ding, D., Erickson, Z., and Pollard, N. S. (2026). "High fidelity capture, reconstruction, and transfer of human demonstrations for robot-assisted bathing". *Under Review*.

Lakshmipathy, A. S. and Pollard, N. S. (2024). "Contactmpc: towards online adaptive control for contact-rich dexterous manipulation". *Robotics: Science and Systems Workshops*.

Le Cleac'h, S., Howell, T. A., Yang, S., Lee, C. Y., Zhang, J., Bishop, A., Schwager, M., and Manchester, Z. (2024). "Fast contact-implicit model predictive control". *IEEE Transactions on Robotics* 40, pp. 1617–1629.

Lee, J. and Shin, S. Y. (1999). "A hierarchical approach to interactive motion editing for human-like figures". *Conference on Computer Graphics and Interactive Techniques*, pp. 39–48.

Li, M., Ferguson, Z., Schneider, T., Langlois, T., Zorin, D., Panozzo, D., Jiang, C., and Kaufman, D. M. (2020). "Incremental potential contact: intersection- and inversion-free large deformation dynamics". *ACM Transactions on Graphics*.

Li, Y., Fu, J. L., and Pollard, N. S. (2007). "Data-driven grasp synthesis using shape matching and task-based pruning". *IEEE Transactions on Visualization and Computer Graphics* 13.4, pp. 732–747.

Lin, T., Yin, Z. H., Qi, H., Abbeel, P., and Malik, J. (2024a). "Twisting lids off with two hands". *Conference on robot learning*. PMLR.

Lin, T., Zhang, Y., Li, Q., Qi, H., Yi, B., Levine, S., and Malik, J. (2024b). "Learning visuotactile skills with two multifingered hands". *arXiv:2404.16823*.

Liu, F., Patil, V., Erickson, Z., and Temel, Z. (2022). "Characterization of a meso-scale wearable robot for bathing assistance". *IEEE International Conference on Robotics and Biomimetics*. IEEE, pp. 2146–2152.

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., and Black, M. J. (2023). "Smpl: a skinned multi-person linear model". *Seminal Graphics Papers: Pushing the Boundaries*. Vol. 2, pp. 851–866.

Luo, Y., Liu, C., Lee, Y. J., DelPreto, J., Wu, K., Foshey, M., Rus, D., Palacios, T., Li, Y., Torralba, a., et al. (2024). "Adaptive tactile interaction transfer via digitally embroidered smart gloves". *Nature Communications* 15.1, p. 868.

Lynch, K. M. and Park, F. C. (2017). *Modern robotics*. Cambridge University Press.

Ma, Y. J., Liang, W., Wang, G., Huang, D. A., Bastani, O., Jayaraman, D., Zhu, Y., Fan, L., and Anandkumar, A. (2024). "Eureka: human-level reward design via coding large language models". *International Conference on Learning Representations*.

Madan, R., Valdez, S., Kim, D., Fang, S., Zhong, L., Virtue, D. T., and Bhattacharjee, T. (2024). "Rabbit: a robot-assisted bed bathing system with multimodal perception and integrated compliance". *Proceedings of the 2024 ACM/IEEE international conference on human-robot interaction*, pp. 472–481.

Magee, D. J. (2013). *Orthopedic Physical Assessment*. Elsevier Health Sciences.

Magnenat-Thalmann, N., Laperrière, R., and Thalmann, D. (1989). "Joint-dependent local deformations for hand animation and object grasping". *Graphics Interface*, pp. 26–33.

Mahmood, N., Ghorbani, N., Troje, N. F., Pons-Moll, G., and Black, M. J. (2019). "Amass: archive of motion capture as surface shapes". *IEEE International Conference on Computer Vision*, pp. 5442–5451.

Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., Hoeller, D., Rudin, N., Allshire, A., Handa, A., et al. (2021). "Isaac gym: high performance gpu-based physics simulation for robot learning". *arXiv:2108.10470*.

Mannam, P., Shaw, K., Bauer, D., Oh, J., Pathak, D., and Pollard, N. S. (2023). "Designing anthropomorphic soft hands through interaction". *IEEE-RAS International Conference on Humanoid Robots*, pp. 1–8.

Mao, Y., Yoo, U., Yao, Y., Syed, S. N., Bondi, L., Francis, J., Oh, J., and Ichnowski, J. (2025). "Visuo-acoustic hand pose and contact estimation". *arXiv preprint arXiv:2508.00852*.

Mason, M. T. (2001). *Mechanics of Manipulation*. MIT Press. URL: https://direct.mit.edu/books/monograph/3869/Mechanics-of-Robotic-Manipulation.

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

Miller, A. and Allen, P. (1999). "Examples of 3d grasp quality computations". *IEEE International Conference on Robotics and Automation*. Vol. 2, pp. 1240–1246.

Mirtich, B. and Canny, J. (1994). "Easily computable optimum grasps in 2-d and 3-d". *IEEE International Conference on Robotics and Automation*, pp. 739–747.

Mitchell, J. S. B., Mount, D. M., and Papadimitriou, C. H. (1987). "The discrete geodesic problem". *SIAM Journal on Computing* 16.4, pp. 647–668.

Murray, R. M., Sastry, S. S., and Zexiang, L. (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc.

Pai, D. K., Rothwell, A., Wyder-Hodge, P., Wick, A., Fan, Y., Larionov, E., Harrison, D., Neog, D. R., and Shing, C. (2018). "The human touch: measuring contact with real human soft tissues". *ACM Transactions on Graphics* 37.4, pp. 1–12.

Pang, T., Suh, H. J. T., Yang, L., and Tedrake, R. (2023). "Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models". *IEEE Transactions on Robotics* 39.6.

Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A. A. A., Tzionas, D., and Black, M. J. (2019). "Expressive body capture: 3d hands, face, and body from a single image". *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10975–10985.

Pollard, N. S. and Zordan, V. B. (2005). "Physically based grasping control from example". *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 311–318.

Pollard, N. (1997). "Parallel algorithms for synthesis of whole-hand grasps". *IEEE International Conference on Robotics and Automation*. Vol. 1, pp. 373–378.

Prattichizzo, D. and Trinkle, J. (2008). *Grasping*. Springer.

Pressure Profile Systems (2026). *TactileGlove: Hand pressure and force measurement*. https://pressureprofile.com/body-pressure-mapping/tactile-glove. Accessed: 2026-01-26.

Qi, H., Kumar, A., Calandra, R., Ma, Y., and Malik, J. (2022). "In-hand object rotation via rapid motor adaptation". *Conference on Robot Learning*.

Qi, H., Yi, B., Suresh, S., Lambeta, M., Ma, Y., Calandra, R., and Malik, J. (2023). "General in-hand object rotation with vision and touch". *Conference on Robot Learning*.

Qin, Y., Su, H., and Wang, X. (2022). "From one hand to multiple hands: imitation learning for dexterous manipulation from single-camera teleoperation". *IEEE Robotics and Automation Letters* 7, pp. 10873–10881.

Qin, Y., Wu, Y. H., Liu, S., Jiang, H., Yang, R., Fu, Y., and Wang, X. (2021). "Dexmv: imitation learning for dexterous manipulation from human videos". *European Conference on Computer Vision*, pp. 570–587.

Rajeswaran, A., Kumar, V., Gupta, A., Vezzani, G., Schulman, J., Todorov, E., and Levine, S. (2018). "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations". *Robotics: Science and Systems*.

Rempe, D., Guibas, L. J., Hertzmann, A., Russell, B., Villegas, R., and Yang, J. (2020). "Contact and human dynamics from monocular video". *European Conference on Computer Vision*, pp. 71–87.

Roa, M. A. and Suarez, R. (2009). "Computation of independent contact regions for grasping 3-d objects". *IEEE Transactions on Robotics* 25.4, pp. 839–850.

Roa, M. A. and Suárez, R. (2015). "Grasp quality measures: review and performance". *Springer Autonomous Robots* 38.1, pp. 65–88.

Romero, C., Casas, D., Chiaramonte, M. M., and Otaduy, M. A. (2022). "Contact-centric deformation learning". *ACM Transactions on Graphics* 41.4, pp. 1–11.

Romero, J., Tzionas, D., and Black, M. J. (2017). "Embodied hands: modeling and capturing hands and bodies together". *ACM Transactions on Graphics* 36.6.

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

Rubert, C., Kappler, D., Bohg, J., and Morales, A. (2019). "Predicting grasp success in the real world-a study of quality metrics and human assessment". *Robotics and Autonomous Systems* 121, p. 103274.

Sawhney, R. (2021). *Fcpw: fastest closest points in the west*. URL: https://github.com/rohan-sawhney/fcpw.

Schmidt, R., Grimm, C., and Wyvill, B. (2006). "Interactive decal compositing with discrete exponential maps". *ACM Transactions on Graphics* 25.3, pp. 605–613.

Sharp, N. and Crane, K. (2020). "A laplacian for nonmanifold triangle meshes". *Computer Graphics Forum* 39.5.

Sharp, N., Soliman, Y., and Crane, K. (2019). "The vector heat method". *ACM Transactions on Graphics* 38.3.

Shaw, K., Bahl, S., and Pathak, D. (2022). "Videodex: learning dexterity from internet videos". *Conference on Robot Learning*.

Siciliano, B. and Khatib, O. (2007). *Handbook of Robotics*. Springer-Verlag.

Sivakumar, A., Shaw, K., and Pathak, D. (2022). "Robotic telekinesis: learning a robotic hand imitator by watching humans on youtube". *Robotics: Science and Systems*.

Starke, S., Zhang, H., Komura, T., and Saito, J. (2019). "Neural state machine for character-scene interactions". *ACM Transactions on Graphics* 38.6.

Sundaram, S., Kellnhofer, P., Li, Y., Zhu, J. Y., Torralba, A., and Matusik, W. (2019). "Learning the signatures of the human grasp using a scalable tactile glove". *Nature*, pp. 698–702.

Suresh, S., Qi, H., Wu, T., Fan, T., Pineda, L., Lambeta, M., Malik, J., Kalakrishnan, M., Calandra, R., Kaess, M., Ortiz, J., and Mukadam, M. (2024). "Neuralfeels with neural fields: visuotactile perception for in-hand manipulation". *Science Robotics* 9.96, eadl0628.

Taheri, O., Ghorbani, N., Black, M. J., and Tzionas, D. (2020). "Grab: a dataset of whole-body human grasping of objects". *European Conference on Computer Vision*, pp. 581–600.

Todorov, E., Erez, T., and Tassa, Y. (2012). "Mujoco: a physics engine for model-based control". *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033.

Tripathi, S., Chatterjee, A., Passy, J. C., Yi, H., Tzionas, D., and Black, M. J. (2023). "DECO: Dense estimation of 3D human-scene contact in the wild". *IEEE/CVF International Conference on Computer Vision*, pp. 8001–8013.

Turpin, D., Zhong, T., Zhang, S., Zhu, G., Heiden, E., Macklin, M., Tsogkas, S., Dickinson, S., and Garg, A. (2023). "Fast-grasp'd: dexterous multi-finger grasp generation through differentiable simulation". *IEEE International Conference on Robotics and Automation*, pp. 8082–8089.

UFACTORY (2026). *UFACTORY xArm 7 Robotic Arm*. https://www.ufactory.us/product/ufactory-xarm-7. Accessed: 2026-01-26.

Vicon Motion Systems (2026). *V16 camera specifications - Vantage documentation*. https://help.vicon.com/space/Vantage/15041618/V16+camera+specifications. Accessed: 2026-01-26.

Wan, W., Geng, H., Liu, Y., Shan, Z., Yang, Y., Yi, L., and Wang, H. (2023). "Unidexgrasp++: improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning". *IEEE/CVF International Conference on Computer Vision*, pp. 3891–3902.

Wang, R., Zhang, J., Chen, J., Xu, Y., Li, P., Liu, T., and Wang, H. (2023). "Dexgraspnet: a large-scale robotic dexterous grasp dataset for general objects based on simulation". *IEEE International Conference on Robotics and Automation*, pp. 11359–11366.

Wang, Y., Wang, Z., Liu, L., and Daniilidis, K. (2024). "Tram: global trajectory and motion of 3d humans from in-the-wild videos". *European Conference on Computer Vision*, pp. 467–487.

Wei, W., Wang, P., and Wang, S. (2023). "Generalized anthropomorphic functional grasping with minimal demonstrations". *arXiv:2303.17808*.

Xing, C., Li, H., Wei, Y. L., Ren, T. A., Tu, T., Lin, Y., Schumann, E., Zheng, W. S., and Cutkosky, M. R. (2025). "Taccap: a wearable fbg-based tactile sensor for seamless human-to-robot skill transfer". *arXiv preprint arXiv:2503.01789*.

Yang, L., Zhan, X., Li, K., Xu, W., Li, J., and Lu, C. (2021). "CPF: learning a contact potential field to model the hand-object interaction". *IEEE/CVF International Conference on Computer Vision*, pp. 11097–11106.

Yang, W. and Posa, M. (2024). "Dynamic on-palm manipulation via controlled sliding". *Robotics: Science and Systems*.

Ye, Y. and Liu, C. K. (2012). "Synthesis of detailed hand manipulations using contact sampling". *ACM Transactions on Graphics* 31.4, pp. 1–10.

Yin, J., Qi, H., Wi, Y., Kundu, S., Lambeta, M., Yang, W., Wang, C., Wu, T., Malik, J., and Hellebrekers, T. (2025). "Osmo: open-source tactile glove for human-to-robot skill transfer". *arXiv preprint arXiv:2512.08920*.

Zhang, H., Christen, S., Fan, Z., Hilliges, O., and Song, J. (2024a). "Graspxl: generating grasping motions for diverse objects at scale". *European Conference on Computer Vision*, pp. 386–403.

Zhang, J. Y., Pepose, S., Joo, H., Ramanan, d., Malik, J., and Kanazawa, A. (2020). "Perceiving 3D human-object spatial arrangements from a single image in the wild". *European Conference on Computer Vision*, pp. 34–51.

Zhang, Y., Gopinath, D., Ye, Y., Hodgins, J., Turk, G., and Won, J. (2023). "Simulation and retargeting of complex multi-character interactions". *ACM SIGGRAPH Conference Proceedings*, pp. 1–11.

Zhang, Z., Wang, H., Yu, Z., Cheng, Y., Yao, A., and Jin Chang, H. (2024b). "Nl2contact: natural language guided 3d hand-object contact modeling with diffusion model". *European Conference on Computer Vision*, pp. 284–300.

Zhou, Q. and Jacobson, A. (2016). "Thingi10k: a dataset of 10,000 3d-printing models". *arXiv:1605.04797*.

Zhou, Q. and Koltun, V. (2014). "Color map optimization for 3d reconstruction with consumer depth cameras". *ACM Transactions on Graphics* 33.4, pp. 1–10.

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

Zhou, W. and Held, D. (2022). "Learning to grasp the ungraspable with emergent extrinsic dexterity". *Conference on Robot Learning*.

Zhou, W., Jiang, B., Yang, F., Paxton, C., and Held, D. (2023). "Hacman: learning hybrid actor-critic maps for 6d non-prehensile manipulation". *Conference on Robot Learning*.