4-27-2016

Geometric Energies for Haptic Assembly

Morad Behandish
University of Connecticut - Storrs, m.behandish@uconn.edu

Geometric Energies for Haptic Assembly

Morad Behandish

B.Sc., University of Tehran, 2008

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

At the

University of Connecticut

2016

APPROVAL PAGE

Master of Science Thesis

Geometric Energies for Haptic Assembly

Presented by
Morad Behandish, B.Sc.,

Major Advisor _____
Horea T. Ilieş

Associate Advisor _____
Thomas J. Peters

Associate Advisor _____
Chun-Hsi (Vincent) Huang

University of Connecticut
2016

Acknowledgements

I am very grateful to my advisor Horea T. Ilieş, Professor of Mechanical Engineering at UConn, for years of selfless support and patient mentorship besides insightful research advisorship. Horea has been more than a great academic advisor by providing a truly exceptional example of *character* on top of academic and professional excellence. The ongoing pleasure of working with him has been one of the most rewarding experiences of my graduate education.

My thanks also go to Thomas J. Peters and Chun-Hsi (Vincent) Huang, Professors of Computer Science and Engineering at UConn, for introducing me to the subjects of computer graphics and computer architecture, respectively, for serving on my M.Sc. committee, and for providing me with valuable feedback.

I would like to thank my associate advisor and Dean of Engineering at UConn, Kazem Kazerounian, for his confidence in my work and his support of my professional development throughout my graduate education at UConn.

Special Thanks to Vadim Shapiro, Professor of Mechanical Engineering at UW-Madison, my advisor's Ph.D. advisor, and my soon-to-be post-doc supervisor, for all the great discussions, especially on human-computer interaction as an essential component of future's manufacturing environments.

I am grateful to my start-up mentors through UConn's Innovation Quest (iQ)[1] and other entrepreneurship resources for their tremendous support of the commercialization of the software tools developed as part of this project. Most special thanks go to Keith Fox, Managing Partner of iQ who founded the program at UConn and keeps sponsoring it; Rich Dino, Professor of Business at UConn who does the most magnificent job at directing the program; and Kevin Bouley, President and CEO of NERAC who is one of the greatest driving forces of entrepreneurship at UConn. My thanks also go to Rick Kollmeyer, Dave Menard, Lane Watson, Dave Pepin, Kathy Rocha, Hadi Bozorgmanesh, Mark Smith, Matt Cremins, and many others from whom I learned a great deal about business development.

[1] http://innovationquest.org/uconn.

Thanks to Ryan A. Pavlik, currently a software engineer at Sensics, who helped me get started with programming haptic devices when he was a graduate student at Iowa State University—and thanks to his advisor, Judy Vance, for facilitating the connection. The short C++ code snippet that he sent me back in 2013 saved me weeks of disoriented attempts that I would have spent otherwise trying to learn nontrivial programming tricks on my own.

It has been a great pleasure working at the Computational Design Lab (CDL) at UConn where I had the chance to interact with bright individuals including Hüseyin Erdim whose research was the first thing that caught my eye when applying to CDL, Ata Eftekharian who helped me a great deal when I first arrived in the US and started in CDL, Radu Corcodel, Denis Dorozhkin, Frol Periverzov, Pouya Tavousi, and Reed Williams with all of whom I have had fruitful discussions over the years.

A special tip of the hat goes to Kamyar Momeni, an exceptional (ex-)housemate for about a half decade and a true friend for life.

Thanks to the dearest Raana Kashfi for being such an amazing companion especially throughout the past few stressful months.

Most importantly, I would like to thank my parents, to whom this thesis is dedicated. My late father Morteza Behandish has been and continues to be—more than a decade after his passing—my number one role-model having taught me principles I aspire to maintain throughout my life. My beloved mother Nasrin Tabrizi has continuously supported and trusted me during every step of my personal development. She deserves the biggest gratitude for embracing my decision of pursuing graduate education in the US, which meant a (seemingly endless) separation from his only son over thousands of miles for several years.

Finally, I express my appreciation for support from the National Science Foundation (NSF),[2] without which this research would not have been remotely possible.

Disclaimer: The responsibility for any errors and omissions lies solely with the author. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the NSF.

The author would also like to emphasize that the identification, exemplification, or choice of any commercial product or trade name throughout this thesis does not imply endorsement or recommendation.

Morad Behandish
March 2016
Storrs, CT, USA

[2]Grant numbers CMMI-1200089, CMMI-0927105, and CNS-0927105.

Contents

Abstract

Haptic-assisted virtual assembly and prototyping has seen significant attention over the past two decades. However, in spite of the appealing prospects, the adoption has been slower than expected. Putting hardware limitations aside, the main roadblocks faced in software development can be traced to the lack of effective and efficient computational models. Such models must 1) accommodate the inherent geometric complexities faced when assembling objects of arbitrary shape; and 2) conform to the computation time limitation imposed by the frame rate requirements—namely, 1 kHz for haptic feedback compared to the more manageable 30−60 Hz for graphic rendering. The fulfillment of these competing objectives is far from trivial.

In this thesis, I propose the concept of a generic 'geometric energy' field to obtain the guidance forces and torques that effectively assist the user in the exploration of the virtual environment (VE), from repulsing collisions to attracting proper contact. The energy function is formulated as a cross-correlation of shape descriptors called skeletal density functions (SDF), which applies to arbitrary geometry. I show that this approach unifies the two phases of free motion (based on collision detection) and fine insertion (based on geometric constraints) widely popular in the recent implementations. The formulation can thus be regarded as a generalization of the manually specified 'virtual fixtures' or heuristically identified 'mating constraints' proposed in the literature. Although such a generalization comes at the expense of computational intensity in its original form, the computations can be streamlined by leveraging Fourier transforms. Particularly, the real-time algorithm admits an efficient implementation using fast Fourier transforms (FFT) accelerated via graphics processing units (GPU). I show that the proposed method is effective for assembling objects of arbitrary topological, geometric, and syntactic complexity, providing a meaningful trade-off between the desired fidelity and computational efficiency.

The results suggest that the proposed approach is a powerful unifying alternative to the existing myriad of ad hoc techniques, thus opens up new promising theoretical and computational directions for haptics researchers and developers.

Preface

> "And I found that of all the senses the eye was the most superficial, the ear the most haughty, smell the most voluptuous, taste the most superstitious and inconstant, touch the most profound and philosophical."
>
> Denise Diderot, 1749 [315]

Haptic technology, pertaining to sensing and manipulation of virtual objects though 'touch', is an emerging technology in the modern virtual reality (VR) systems, with applications in areas as diverse as product design and prototyping [164, 316, 366, 450], teleoperated and robot-assisted surgery [29, 247, 309, 310], oral and dental implant operations [234, 244, 308, 371], molecular simulations and training [110, 351, 377, 429], rehabilitation systems [60, 162, 379], and entertainment (e.g., gaming) [6, 59, 295].

Haptic studies are typically organized into three subareas; namely, human haptics, machine haptics, and computer haptics [372], which study the physiological, hardware, and software aspects, respectively. Being particularly relevant to this thesis, computer haptics [182] is defined as algorithms and software associated with integrating touch into human-computer interaction (HCI)—analogous to computer graphics targeting sight [372]. It has seen significant attention over the past two decades in response to the growing need for simulation of more complex phenomena in each of the aforementioned domains.

Among other applications, haptic interfaces have been found useful in several engineering design activities that can benefit from domain expertise and cognitive capabilities of human operators—which are hard to formalize for full automation. These tasks, commonly characterized as 'virtual prototyping', range from conceptual and aesthetic design to ergonomic validation and assembly planning.

In this Master's thesis I address the problem of haptic-assisted assembly of mechanical parts. During the course of this project, I observed that the state-of-the-art methods are either limited to the assembly of parts with very simple geometric features (e.g., those involved in lower kinematic pairs) that are automatically detectable, or heavily dependent on user input for constraint specification. Most methods generally presume a priori knowledge of the type of contact surfaces that one deals with, and are not generalizable to support objects of arbitrary shape.

In contrast, I propose an analytic approach that draws on recent ideas in configuration space modeling using convolution algebras [270, 271] and analytic methods for protein docking [11, 74]. I introduce the concept of 'geometric energy' field defined as a convolution of shape descriptors called skeletal density functions (SDF) that applies to arbitrary geometry. Without making any simplifying assumption on the mating features or the type of contact, the energy field automatically produces both repulsion effects as collision response and attraction effects for assembly guidance and final snapping. This formulation unifies the recently popular two-phase approach (described in Section 2.4.2) that uses (typically approximate) collision detection for free motion and (inherently ad hoc) geometric guidance for low-clearance insertion, eliminating the need for switching and blending between two phases. The persistent challenges that imposed the need for such duality; namely, the failure of collision detection in low-clearance assembly involving lower kinematic pairs, on the

one hand, and the inability of the 'virtual fixture' and 'mating constraint' methods to handle higher kinematic pairs and arbitrary contact features, on the other hand, are mitigated altogether within a single analytic model.

In addition to the desirable theoretical properties, the convolution theorem enables transforming the correlations between parts (i.e., the energies) and their differentiations (i.e., forces and torques) into simple multiplication operations in the Fourier domain using fast Fourier transforms (FFT) implemented on the graphics processing units (GPU). Unlike the existing approaches which are inevitably restricted by topological complexity (e.g., connectivity and number of holes), geometric complexity (e.g., convexity and type of surfaces) or syntactic complexity (e.g., number of triangles or voxels), this method does not impose any such restriction. Instead, it allows for a systematic trade-off between the achieved fidelity and the computational efficiency regardless of the input size or complexity. This 'graceful' trade-off is made possible by choosing on-the-fly an appropriate number of dominant modes of the truncated geometric signals (e.g., the SDF descriptors, their derivatives, and their convolutions) in the frequency domain.

To the best of my knowledge, this work is the first application of analytic methods to haptic assembly, in particular, and to physically-based modeling for real-time applications, in general. It is worthwhile noting that research on analytic solid geometry (e.g., [10, 20, 270, 271]) is still at a stage of infancy. Despite the elegant mathematical formalism and desirable computational properties, these methods require further research, testing, and validation before being integrated into current solid modeling and physics engines. I hope that this thesis will provide promising theoretical and computational directions for academic researchers and software developers to further develop and convert these ideas into widely available software packages.

Chapter 1

Introduction

> "Smartness in mechanical devices is often realized through interaction that enhances dumb algorithms so they become smart agents... Interactive systems are grounded in an external reality both more demanding and richer in behavior than the rule-based world of noninteractive algorithms."
>
> Peter Wegner, 1997 [420]

1.1 Human-Computer Interaction

The essence of 'interaction' for taking full advantage of computers in scientific advancements and day-to-day activities is becoming more evident with the rapid growth in hardware and software technologies. Although digital computers have been constantly breaking the records of data storage and processing beyond human imagination over the past 50 years—thanks to the semiconductor industry since 1950s and

the Moore's law[1] [354]—their embarrassingly persistent failure in performing tasks as natural and intuitive to humans as facial recognition, speech recognition, language translation, and similar semantic interpretation problems has raised important questions: What are the intrinsic (i.e., theoretical) limitations of the digital computers (and underlying computing models) in automating manual tasks? If computers are great at certain tasks—e.g., storing and processing vast amounts of data in short amounts of time—while humans are better at others—e.g., feeling, learning, and experiencing through the five senses—how can we characterize and leverage those capabilities simultaneously? Rather than hopelessly trying to entirely automate all tasks that were once entirely manual, is it feasible to realize the 'best of both worlds' though effective human-computer interaction (HCI)?[2]

1.1.1 Multimodal Interactions

Much like every other sort of interaction or experience that we, as humans, have with our environment, our interaction with computers can be viewed as a two-way information exchange via a combination of our five senses—namely, sight, hearing, touch, taste, and smell.[3] However, the interfaces that we use even today for this interaction are often described as an 'information bottleneck' [360]. Needless to mention, the substitution of 'punchcards'—punched offline with typewriter-like 'keypunch' devices—

[1] The 'transistor' was invented in 1947 by William Shockley and his colleagues at Bell Laboratories [354]. In 1952, Werner Jacobi at Siemens filed a patent [209] for the concept of integrated circuits (IC). However, the first working integrated prototype was demonstrated by Jack Kilby at Texas Instruments in 1958, who is credited with the invention of IC [354].

[2] Also called human-machine or man-machine interaction or interface in the literature.

[3] Perhaps with the exception of direct brain-computer interfaces (BCI) made possible for extremely simple tasks using electroencephalographic (EEG) signals collected on the human scalp. Research in this areas is still in the stage of infancy and is limited to few real-world applications. For a recent survey of BCI developments, see [127].

that endured as the primary device for information exchange even into the PC era until mid 1980s with today's interactive combinations—e.g., keyboards, mice, joysticks, display devices, and more recently, touchscreens—has been a rather significant (despite gradual) progress. However, the extent of the interactive phenomena that can be replicated by these limited set of gadgets is not nearly comparable with those of many other human interactions in social and natural environments. These activities range from regular day-to-day tasks such as tying one's shoe laces to specialized traits such as a craftsman's making artifacts with their hands or a violin player's tuning and playing the instrument.

It would be unfair not to mention the significant progress in both research and commercial capacities in incorporating at least two of the five senses quite effectively. These two senses are, obviously, the sense of hearing engaged by digital audio and the sense of sight engaged by computer graphics technologies. Since the commercialization of the television in late 1940s, the developments in both visual and auditory categories have been driven substantially by the multibillion dollar entertainment (e.g., motion picture and gaming) industry—not to mention other domains such as medical and defense applications. The animations and video games today are hardly distinguishable from real scenes, and the human-computer communication has fairly matured along at least one of the two directions of information flow. However, the tools required along the opposite direction—i.e., image processing on 2D images, surface reconstruction from 3D point clouds, speech and language recognition, and other semantic interpretation tools—despite having been developed significantly as well, are not as reliable and successful yet as demanded by real applications.

The next frontier in modeling and implementing human sensorimotor capabilities involves the addition of the sense of touch. Unlike the visual and auditory infor-

(a) (b) (c)

FIGURE 1.1.1: Haptic devices on the market: tactile feedback as in (a) touchscreen devices; and kinaesthetic feedback as in (b) single-point interaction devices (e.g., stylus-type) and (c) multiple-point interaction devices (e.g., glove-type).

mation that can be more easily replicated (i.e., interpreted, stored, and recreated) by (projections of) geometric representations and electrical signals, respectively, it is not very clear what the primitive information units for modeling touch should be. For this and other practical reasons explained in Section 1.2, the touch-enabled HCI has not seen the same rapid growth. The most notable examples of its successful deployment today are either in specific application domains with a restricted and well-understood role for touch—e.g., flight simulation, clinical training, rehabilitation, and alike—or to simplified scenarios that involve a narrow and primitive subset of sensory experience—e.g., touchscreen devices with vibration feedback. In these applications, the type of touch interaction is extremely restrained to facilitate designing special purpose devices that engage only certain parts of the body—e.g., through particular muscles/joints or only fingertips (Fig. 1.1.1 (a))—or involve confined interaction modes—e.g., force feedback at a single point while holding a stylus with a natural grasp like holding a knob or wand (Fig. 1.1.1 (b)). More examples of the commercial haptic devices are shown in Fig. 1.1.1. The reader is referred to [177, 249, 400] for a review of haptic devices on the market.

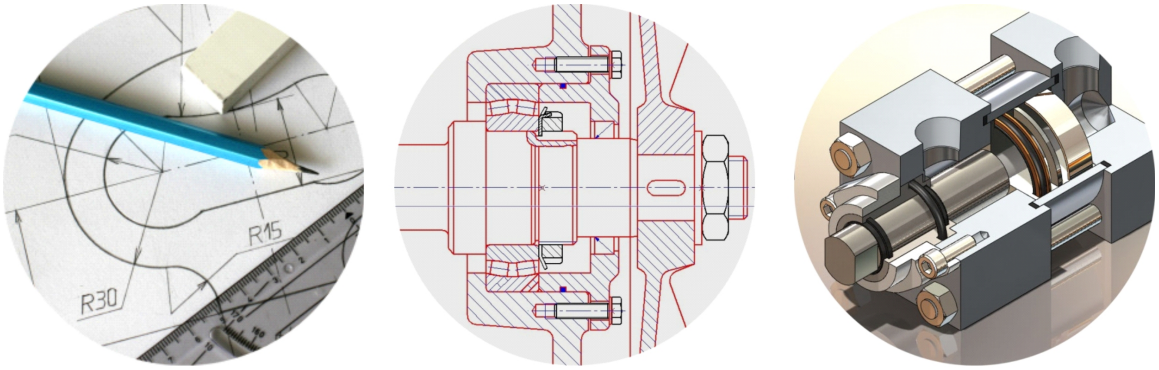
Broadly, virtual reality (VR) is defined as a high-end human-computer interface that involves real-time simulation and multimodal interactions—i.e., a user interface that engages multiple sensorial channels including but not limited to visual, auditory, and tactile modalities [62]. Often, VR applications are characterized by three key features (called the three "I"s of VR); namely, interaction, immersion, and imagination [62]. Among other applications, mechanical design tasks—especially at the conceptual stages—are highly dependent on the designer's imagination, creativity, and experience that manifests 'unselfconsciously' [5]. At least until (if ever, at all) these cognitive activities can be understood and formalized to an extent that full design automation becomes an apprehensible alternative,[4] the inherently interactive design process can be significantly improved in terms of performance and productivity by multimodal interactions—i.e., incorporating as many of the five senses into HCI as physically and computationally possible.[5]

1.1.2 User Interfaces for CAD

To understand the importance of haptic interfaces, in general, and the implications of developing foundations and algorithms to support touch-assisted engineering tools, one needs to appreciate the analogous role that the graphical user interface (GUI) has played in the evolution of computer-aided design, analysis, and manufacturing.

[4]Unlike the case with other tedious activities associated with product development whose industrial automation historically instigated the reduction (and eventually disappearance) of the human's role in the process—most notable examples being manufacturing automation in the 18th and 19th centuries with the advent of steam machines and computing automation in the 20th century with the advent of the digital computers—it is not clear whether the full automation for the design process to such an extravagant extent is possible (or even desirable).

[5]Among the recent efforts in this direction, one could exemplify IBM's 2012 5-in-5 list that predicted the availability of computers that are able to mimic all five senses and a variety of "right-brain activities" within the next five years.



(a) (b) (c)

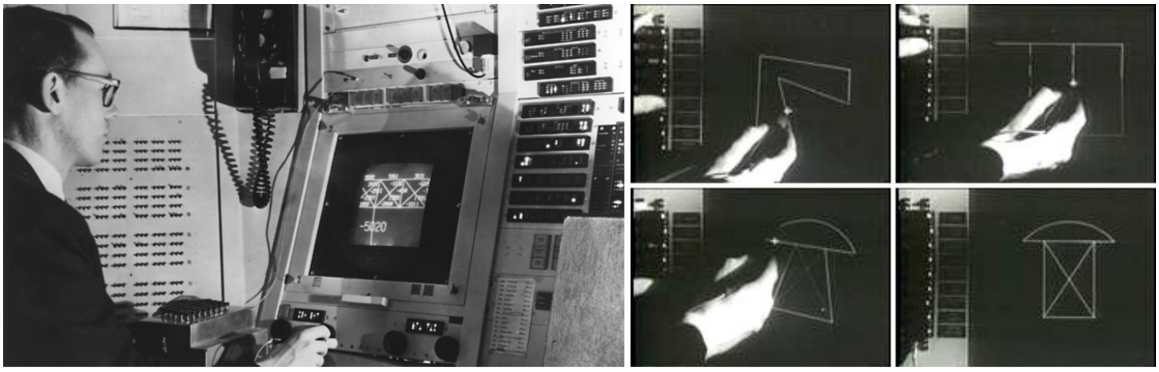
FIGURE 1.1.2: The evolution of visual aids for mechanical design; from paper-and-pencil sketching (a) to computer-aided tools from 2D drafting (b) to 3D modeling (c).

It is hard to underestimate the crucial role of visual aids (e.g., symbols, sketches, and diagrams) in carrying out even the simplest cognitive processes[6] let alone its importance in scientific and engineering practice. Architects, engineers, and artists have used sketches for hundreds of years to conceptualize, examine, and communicate their creations. The advent of digital computing and its democratization during the 20th century provided the designers with an incredibly powerful new set of tools. However, to take full advantage of the computational power, there was no other choice but to equip this new machine with visual aids to replace the traditional sketch pad, paper-and-pencil in assisting the designer's imagination. See Fig. 1.1.2.

In 1963, Ivan Sutherland[7] developed a revolutionary computer program called Sketchpad: "a man-machine graphical communication system" [383]. Sketchpad can be viewed as the first computer-aided design (CAD) software and a major break-

[6] "Two minutes with a pencil on the back of an envelope lets us solve problems which we could not do in our heads if we tried for a hundred years." –Christopher Alexander [5].

[7] Ivan Sutherland is an American computer scientist and one of the pioneers of the 'Internet'. He developed Sketchpad as his Ph.D. thesis in MIT in 1963, which earned him the Turing award in 1988 and the Kyoto prize in 2012. It is fair to call him the inventor of GUI and the father of the modern human-computer interaction.



(a) (b)

FIGURE 1.1.3: Ivan Sutherland showcasing Sketchpad at MIT's Lincoln Labs.

through in the development of CAD and computer graphics as disciplines. It gave birth to the concept of a GUI and spurred a new approach to HCI. Some of the ideas first demonstrated in Sketchpad—e.g., hierarchical structures, master vs. slave objects, global vs. local coordinate frames, constraints, etc.—are still part of the development environments today. More details about Sketchpad and the direction it provided for the late 20th century HCI can be found in [9, 384]. See Fig. 1.1.3.

In 1972, Herbert Voelcker[8] and his collaborators started the production automation project (PAP) [407] aiming to develop the solid modeler called part and assembly description language (PADL) [307]. PAP was originally established aiming to provide informationally complete computer representations to support the automation of numerical control (NC) machining. PADL-1 [131] used a combination of constructive solid geometry (CSG) and boundary representation (B-rep) and was made publicly available in 1977. PADL-2 [335] followed closely after in 1981 from which UniSolid was developed by Unigraphics^{TM}. Around the same time in 1978 the B-rep solid modeling

[8]Herbert Voelcker is a professor emeritus at Cornell University and one of the founders of the modern theory and practice of solid modeling [405, 406]. He founded PAP at the University of Rochester in 1972 and was joined by Aristides Requicha in 1973 who further contributed to the mathematical foundations [331], modeling [332], and representation [334] of mechanical parts and assemblies. A historical review of PAP can be found in [407].

kernel Romulus was released by ShapeData, which influenced the successor kernels Parasolid® and ACIS in late 1980s and early 1990s. Other notable CAD software that emerged in those years are AutoCAD®, CATIA®, and Pro/ENGINEER® in 1980s and SolidWorks®, SolidEdge®, and Autodesk Inventor™ in 1990s. See [64] for a broad historical review of the evolution of computer graphics and CAD industry, and [421] for a more detailed review of early CAD packages in particular.

In the past two decades, the 3D modeling, simulation, and visualization tools for CAD have evolved into versatile product lifecycle management (PLM) software to manage complete digital product descriptions for development, design, and manufacturing known as digital mock-ups (DMU) [61]. DMU serves as a platform for virtual (or digital) prototyping (VP/DP), which is becoming a commonly adopted design and validation practice in several industrial sectors [36]. Compared to physical prototypes, the virtual models are more rapid, less expensive, and easily reconfigurable; they support more variants and allow for several simulation runs on a single model [61]. Although today's versatile GUIs allow an interactive visual examination of 3D models at different design and validation stages, a major deficiency can be traced to handling 3D geometry and functions using 2D input/output media [61]. Almost all current CAD software use a combination of widget-based interaction using a pointing device (e.g., a mouse or touchpad) cursor on the screen—namely, the standard windows, icons, menus, and pointer (WIMP) interaction style—and alphanumeric input using a keyboard [50], while research has shown that multimodal interactions enable higher accuracy and lower execution times [180].

Several general-purpose robotic devices meant for touch-enabled input/output interactions with computers (called 'haptic interfaces') ranging from 3D to 6D input (i.e., position and orientation encoding) and output (i.e., force and torque feedback)

became widely available in 1990s. Among the most widely used commercial devices one can refer to SensAble® Phantom® line of devices[9] that originated from MIT Touch Lab [277, 350], which I shall use for testing our models in Chapter 5. Among other popular devices in the same category are Haption Virtuose6D™—popular for assembly planning with large workspace requirements—CyberGlove® CyberGrasp®, Force Dimension Sigma/Omega/Delta, and Novint Falcon.[10] Although these general-purpose devices have found numerous applications in research and development labs—as well as special-purpose devices developed in-house and optimized for certain applications—their democratization for the consumer market has been hindered by

1. high hardware retail prices due to limited number of units produced;
2. the lack of common language standards and general-purpose software development platforms to make the technology accessible to non-experts; and
3. the absence of a clear customer demand—which is both responsible for and aggravated by the above two factors.

The reader is referred to [177, 249, 400] for surveys of the evolution of haptic interfaces and [122, 373, 403] for the extent of their applications in engineering, medicine, entertainment, and education.

I postpone a more detailed review of the application of haptics to assist virtual prototyping and assembly to Section 2.1. Here it suffices to emphasize that, in a similar fashion that developments in computer graphics have been essential to achieve the

[9] SensAble® was acquired by Geomagic® in 2012, which was in turn acquired by 3D Systems in 2013. The line of devices formerly known by the trademarks SensAble® Phantom® series were subsequently rebranded as Geomagic® Touch™ series.

[10] I hereby emphasize that mentioning or using a particular device for testing purposes in this project does not imply recommendation of that device over any others by the author, and the same functionality can be replicated almost identically with similar devices on the market. See www.bracina.com/haptichardware.html for a list of haptic devices on the market.

current functional versatility of interactive 3D modeling tools for CAD, the integration of haptic interfaces to achieve their full potential calls for extensive research in computer haptics [182]. Developments in both graphics and haptics interaction modes are, in turn, dependent on the advancement of more inclusive mathematical abstractions (e.g., geometric modeling and reasoning) and richer computer representations (e.g., data structures and algorithms).

1.1.3 Practical Significance

Among the numerous applications of graphics- and haptics-enabled HCI, haptic feedback is of particular practical significance in application domains that involve an interplay of shapes and motions. More specifically, when dealing with planning problems that require complex spatial reasoning in higher-dimensional configuration spaces—e.g., assembly planning, path and motion planning, manufacturing process planning, etc.—there are two extreme ends to a computer-aided solution process:

- The algorithmic approach involves attempting to model a 'field description' of the problem [5]—e.g., define a metric to quantify the relative performance of the different plans—followed by devising a search algorithm to find a solution—e.g., an optimization algorithm that navigates through the configuration space and finds local and/or (idealistically) global optima by trial-and-error.

- The interactive approach involves leveraging the human agent's expertise and domain knowledge to reject a large subset of trials upfront and eventually select an acceptable (if not optimal) solution by interactive evaluation—e.g., visual and sensorimotor inspection via graphic and haptic feedback, respectively.

Clearly, each of the above approaches has its own advantages and drawbacks. For noninteractive optimization, one needs to come up with a purely quantitative metric to evaluate all and every candidate plan as accurately as possible, which is unrealistic when inevitable heuristics with limited applicability are involved. Even if such a uniform field description is discovered, finding the near-optimal solution requires searching a high-dimensional solution space which is computationally prohibitive, especially in the presence of constraints.[11] Most search algorithms are doomed to converge to local optima without any global guarantees, and are generally unable to systematically partition the search space to what can be referred to as "qualitatively distinct" solution subspaces or regions.[12] The interactive approach, on the other hand, allows the human agent to identify qualitative distinctions and limit the search space to a few regions while rejecting the rest. The quantitative field description can then assist the user—through proper visual and sensorimotor cues—in finding the local minima in each subspace and further refining the solution.

Although the main application of the method described in this thesis is interactive assembly planning, other related spatial planning problems can also benefit from the 'geometric energy' field description proposed in Chapter 3. In particular, automatic robot path planners (including but not limited to assembly planners) that use decomposition methods such as cylindrical algebraic decomposition (CAD*) [357–359] or faster sampling methods such as probabilistic roadmaps (PRM) [32,147,229,230] to construct a combinatorial structure over the configurations space can skip the redun-

[11] Mathematically, a highly constrained search problem—without the knowledge of a parameterization that explicitly guarantees constraint satisfaction—is an attempt to find lower-dimensional manifolds in a higher-dimensional space, which depicts the computational challenge faced in the algorithmic approach by appealing to simple probabilistic arguments.

[12] The thoughts and language are inspired by long discussions with Saigopal Nelaturi on different subjects regarding the expected role of HCI in future's design and manufacturing environments.

dant exploration of a large subset of infeasible solutions by restricting the search space to that swept by the user through interaction. Similarly, protein docking platforms that rely on FFT-based search algorithms [11,75,121,225,239] to explore the configuration space can exploit the user's ability to quickly identify qualitative docking sites before making quantitative adjustments to the local configurations.

In the next section, I briefly enumerate the current challenges and promising directions for incorporating haptics to assembly planning applications.

1.2 Research Challenges

The growth in the availability and popularity of the fairly recent haptic technology imposes increasing demands for geometric modeling and computing algorithms, to deliver realistic replication of the real-world experience in virtual environments (VE) as efficiently as possible. We identify the following main challenges in the development of haptic-enabled virtual prototyping tools—with virtual assembly and disassembly tasks being considered as the particular subarea of interest:

1. the difficulties for modeling and capturing the inherent geometric complexities faced when assembling objects of arbitrary shape; particularly, the absense of a practically effective force and torque feedback model for assembly and disassembly interactions of lower and higher mating pairs; and

2. the computation time limitation imposed by the frame rate requirements; particularly, the obligation to carry out all computations per frame within a millisecond to maintain a 1 kHz servo-loop rate for haptic feedback compared to the more manageable $30-60$ Hz for graphic rendering.

1.2.1 Geometric Complexities

The primary challenge in developing generic models for haptic feedback lies in a proper formulation of the guidance forces and torques that effectively assist the user in the exploration of the VE, from repulsing collisions to attracting proper contact. In particular, when two objects (e.g., rigid or flexible parts or subassemblies) are being assembled or disassembled in a VE, the effects of the mechanical forces and torques exchanged during the process are simulated by integrating the equations of motion subjected to physical and geometric constraints formulated using contact mechanics and friction models. As depicted in Section 2.2, for the most general type of contact—i.e., a combination of surface, curve, and point contact—collision detection (CD) algorithms are used for computing the resistance forces and torques that are central to the dynamic simulation. However, there are several computational challenges faced when attempting to perform CD in real-time interactive applications. This is especially the case in assembly scenarios that involve tight fits whose 'nominal' geometry is described by contact features that reduce the degrees of freedom (DOF) of relative motion—e.g., by restricting the 6D relative motion of a pair of rigid bodies into a lower-dimensional (e.g., 1D or 2D) subspaces of their configuration space.[13] The dynamic simulation along the lower-dimensional contact subspaces for nominal geometry is inherently unstable and difficult to compute due to the extreme sensitivity to infinitesimal perturbations. The natural solution, similar to the case in real physical assembly, is to add clearances (according to appropriate dimensioning and tolerancing standards [374]) to the nominal geometry for ease of assembly:[14]

[13]The concepts of the configuration space, configuration space obstacle, free space, and contact space for the relative motion of a pair of rigid bodies are introduced in detail in Section 3.2.2.

[14]Here I am specifically considering the virtual simulation of the insertion for 'clearance fits' that the user can do with bare hands (in both physical and virtual setups alike) and does not require

- High-clearance fits have been successfully simulated in VEs—e.g., clearances of $\sim 1-3$ mm using approximate CD with resolution of ~ 0.2 mm [362, 363]. However, most real assembly scenarios require much smaller clearances.

- Low-clearance fits, on the other hand, require more accurate CD algorithms—e.g., clearances of ~ 0.001 mm using exact CD on original B-reps [364, 365]. However, exact CD does not exhibit the required performance at 1 kHz.

In other words, there is an inevitable trade-off: the practical and efficient CD methods use some approximation (e.g., meshing, voxelization, or bounding volume hierarchies) that compromises the required accuracy for low-clearance assembly, while exact CD is not fast enough to handle numerous parts or complex shapes in real-time. The different CD methods for haptic assembly are reviewed in Section 2.2.2 and in greater detail for general applications in [222, 238, 259].

It is important to note that the dynamic instability problem of low-clearance fits whose nominal geometry typically characterizes a lower kinematic pair—i.e., contact maintained over a surface, restricting the motion to 1D or 2D—is actually twofold when it comes to interactive VR applications:

1. the intrinsic geometric representation errors of the approximate CD algorithm (e.g., voxelized or triangulated cylindrical features), which can be eliminated in principle by using exact CD at the expense of computational performance; and

2. the measurement errors/noise due to the haptic device encoder inaccuracies as well as the 'jerking' motion of the hand, which can only be alleviated (e.g., by filtering) at the software level but cannot be completely eliminated.

special tools or processes unlike the instances with 'force fits' or 'shrink fits'.

Unlike what happens in real physical assembly, the dynamic behavior in virtual assembly is simulated using a finite-difference integrator, whose stability is very sensitive to these errors.[15] As a result, CD alone has been found by several researchers [316, 402] to be insufficient for virtual assembly, especially with haptic assistance.

For the lower kinematic pairs that are completely classified into the well-known six classes [306] (detailed in Section 2.1.3), it is possible to abstract the DOF-limiting contact subspaces in terms of 'virtual fixtures' [343] or 'mating constraints' [200] between functional surfaces of simple (e.g., planar, cylindrical, spherical, or conical) shapes. The forces and torques for haptic guidance during insertion of these features are then simplified, for example, by using spring-damper models between the current and eventual configurations. Although such approaches provide faster and more effective alternatives to CD for low-clearance assembly, they also depend on at least one of the following simplifications:

- a priori assumptions on the type of surface features and corresponding kinematic pairs, their explicit semantics, and exact locations on the different parts and subassemblies *manually* prespecified by the user; or

- heuristic methods to *automatically* detect the assembly intent and associated mating features as soon as the features are brought to insertion proximity.

[15] For instance, if a penalty method is used to compute the collision response, a small vibration of the user's hand can cause a penetration during low-clearance insertion. This penetration is not detected or resisted until the next frame of the dynamic simulation when a large penalty occurs and creates a large (possibly overcompensating) force feedback. The repetition of this phenomenon, in turn, creates an inconvenient 'buzzing' effect on the haptic device. This can be alleviated by decreasing the integration time-step, which is lower-bounded by the time spent on CD and other computations per frame, or adding appropriate damping effects, whose excessive use makes the simulation unrealistic.

The abstraction of the mating features for higher kinematic pairs between features of arbitrary shape, on the other hand, is not as simple since no such finite classification exists for the more general case.

As a consequence of the disadvantages in using each method by itself, the common theme in the recent haptic assembly systems is to use hybrid techniques [402] (detailed in Section 2.4.2) that switch between CD and feature-based constraint resolution. This duality creates a great deal of extra complications for switching and blending between the two phases [316]. Moreover, when fixated on the low-clearance fit of a particular pair of features, collision events outside the insertion site can be missed. This defeats one of the main purposes of virtual assembly, which is the early detection of design issues such as unaccounted clashes between the different parts.

In Chapter 3, I propose a generic model in terms of a convolution of special shape descriptors (i.e., the SDF [19]) that unifies collision response and geometric guidance under a single analytic model, presents an implicit generalization of the virtual fixtures for both lower and higher pairs, and eliminates the need for the duality altogether. The method also offers great flexibility in terms of the trade-offs between

1. the precision of collision response and the computational performance; and

2. the 'softness' of insertion experience and the fit clearance.

1.2.2 Computational Limits

The efficiency problem appears more challenging in the case of haptic feedback, when compared to graphic rendering, due to the notorious physiological requirement of (at least) 1 kHz refresh rate necessary for satisfactory tactile experience—especially

to acquire the necessary stiffness when manipulating rigid objects [316]—while only 30−60 Hz is typically perceived as adequate for appealing to human vision [122].

The human touch perception is typically classified into 'kinaesthetic' sensations that are related to muscle control and limb motion, and 'tactile' sensations perceived at the skin receptors [122].[16] Although a bandwidth of 10 Hz is typically considered adequate for the kinaesthetic sensations [372], it is largely dependent on the task, e.g., 1−2 Hz for unexpected signals, 2−5 Hz for periodic signals, up to 5 Hz for internally generated or learned trajectories, and 10 Hz for reflex actions [372, 373]. However, to adequately simulate rigidity with a force feedback device (e.g., in VR-CAD applications), higher frequencies are required due to basic control-theoretic considerations; namely, noting that the maximum stiffness in a closed-loop system is inversely proportional to the square of the regulation period [316]. On the other hand, vibrations of up to 1 kHz can be resolved by the human tactile system, with the highest sensitivity at 250 Hz [373]. To collectively comply with all of these requirements, a response rate requirement of 1 kHz is widely accepted as the standard for most haptic applications—see [63,122,372,373] for more details on the subject. It should be noted, however, that there are studies suggesting that even higher rates of 5−10 kHz might be necessary for improved haptic performance in certain tasks [241] one example being high-fidelity texture discrimination [88, 89].

In Chapter 3, I show that our analytic formulation admits a much simpler algebraic form in the Fourier domain, thanks to the mathematical properties of convolutions

[16] More technically, the term 'proprioception' is used for the broad class of perceptions of the position, state, and movement of the body and limbs in space. This includes vestibular, kinaesthetic, and cutaneous sensations. The 'vestibular' sensations pertain to the perception of balance, head position, and acceleration/deceleration. The 'kinaesthesia' includes the sensation of movement of the body and limbs originating in the muscles, tendons, and joints. The 'cutaneous' sensations pertain to the skin itself, including sensations of pressure (from mechanoreceptors) as well as temperature and pain (from nociceptors) the former of which is more specifically referred to as 'tactile' [315].

and Fourier transforms. An important implication is that real-time force and torque computations can be significantly accelerated using fast Fourier transforms (FFT) on the graphics processing units (GPU) to sub-millisecond running times regardless of topological, geometric, or syntactic complexity.

1.3 Summary of Contributions

The current computational models for constraint-based assembly guidance are either

1. limited to the assembly of solids with very simple geometric features that are automatically detectable; or

2. heavily dependent on user input for constraint specifications.

Both methods generally presume a priori knowledge of the type of contact surfaces that one deals with, and are not generalizable to support objects of arbitrary shape. The majority of ad hoc solutions start from identifying the simplistic DOF-limiting constraints—e.g., restricting the motion to planar, cylindrical, spherical, or conical surfaces or their intersection curves—followed by what can be conceptualized as simple energy formulations to enforce those constraints—e.g., spring-damper models to penalize the violation of co-planarity or co-axiallity conditions.

I propose a generic and unified energy model for real-time assembly guidance that applies to objects of arbitrary shape. The formulation starts from the part geometries and *directly* computes the guidance forces and torques from shape descriptors of interacting features. I do not make any simplifying assumption on the geometry of the mating features and show that implicit generalizations of the so-called virtual fixtures [343,344] that are widely popular for haptic assembly [316,389,390,402] *automatically*

appear in the form of a density distribution in the 3D space, called the skeletal density function (SDF). The SDF shape descriptors are piecewise continuous functions defined over the 3D space for each individual part, whose distributions capture the topological and geometric properties of the surface features that partake in assembly. The spatial overlapping of individual part SDFs—interpreted in the analytic formulation as a *convolution*—generates an artificial potential energy (called the 'geometric energy') field which creates *attraction* forces and torques towards the proper alignment of assembly features. I show that the same energy model also provides *repulsion* forces and torques as a natural byproduct, in the case of collisions. Therefore, it unifies the two phases of free motion and precise insertion into a single interaction mode, thus avoids the duality and switch altogether. The method subsumes analytic collision detection (CD) [270], and provides a generalization to analytic feature matching for geometric guidance.

Additionally, the formulation of the energy function as a convolution allows using ideas from multivariate harmonic analysis [226] to streamline haptic feedback computations. The convolution in the physical space (where the part geometries reside) transfers into a pointwise multiplication of the Fourier expansion of the SDFs for the individual parts (i.e., the 'amplitudes' of the multi-dimensional SDF signals). Guided by this property, I show that the formulation leads to a straightforward mathematical relationship between the Fourier representations of the SDF shape descriptors and the geometric energy field, which can benefit from the efficiency of the FFT algorithms [102]. Moreover, I present explicit analytic equations for computing the gradients of the convolution function (i.e., guidance forces and torques) for arbitrary spatial translations and rotations.

I implement the process using optimized FFT implementation on the highly-parallel GPU architecture. I show that haptic-enabled simulation of realistic assembly scenarios with complex CAD models and low-clearance fits is made possible to an adequate fidelity with the application of GPU-accelerated FFT calls.

In addition to its theoretical generality, computational efficiency, and scalability with parallel-computing, the most outstanding advantages of this paradigm compared to the existing methods are the following:

1. The analytic formulation is quite generic, allowing for a variety of different shape descriptors—ranging from simple PMC to intricate SDF—to be constructed using different kernels in the general formula.

2. The decomposition of the shape descriptors into their Fourier components (i.e., the 'modes' of the 3D signals) allows for a systematic means to trade off the accuracy of computations with the amount of available computation time and resources. In the case of haptic assembly, where there is a limited budget of time (ideally 1 millisecond) available to each simulation frame, one could use truncated Fourier expansions (i.e., apply a 'low-pass filter' to the 3D signals) to significantly speed up the convolution in real-time.

3. The computational performance is indifferent to geometric and syntactic complexity of the objects in the physical domain. Unlike the existing combinatorial approaches to collision detection which scale in computation time with input complexity—e.g., the number of points, triangles, or voxels used in the representation—our method's efficiency depends solely on the degree of fidelity specified by the low-pass filter (i.e., number of retained dominant modes), and does not scale with the original input complexity.

1.4 Outline of the Thesis

This thesis is organized into 5 chapters overviewed below:

- Chapter 1 (the current chapter) serves to motivate the reader regarding the potential benefit of haptic feedback in computer-aided mechanical design activities. In Section 1.1 I attempted to answer a critical question: "why do we need touch?" by drawing an analogy between the revolutionary role of computer graphics in computer-assisted product development in the past 50 years and the potential role of computer haptics in the upcoming decades. I briefly mentioned the theoretical and computational challenges to applying haptic technology to interactive design and assembly in Section 1.2, followed by the proposed approach to addressing these challenges in Section 3.1. The main contributions of the thesis were enumerated in Section 1.3.

- Chapter 2 serves to familiarize the reader with the extensive body of literature on virtual prototyping and assembly. I attempt to illuminate the contrast between the fundamentally different approaches (e.g., physically-based versus constraint-based) and the set of computational tools that are in common use in each case. In Section 2.1 I define the concepts associated with virtual prototyping and assembling tasks and describe a selection of the significant studies and existing systems for haptic assistance in some detail. For easy access and comparison, I collect an extensive (despite incomplete) list of important implementations with brief pointers to their key features in chronological order in tables 2.1.1 and 2.1.2. The two common paradigms for real-time simulation of interactions in virtual environments are presented in Sections 2.2 and 2.3 followed by the currently dominant hybrid approach presented in 2.4.

- Chapter 3 presents the mathematical formulation of the method in extensive detail. I outline the general approach in Section 3.1 in both theoretical and algorithmic terms. Section 3.2 offers an overview of the preliminary concepts and definitions that are prerequisite to the subsequent developments. Section 3.3 introduces the general theme that underlies the analytic formulation (namely, the 'correlation' paradigm) and lays out the thought process that leads to the definition of the SDF as a generalization of traditional shape skeleton. I define the SDF elaborately in Section 3.4 as a suitable 'affinity function' and present its properties and their implications for our purposes. In Section 3.5 I revisit the correlation function defined earlier and use it to define a *shape complementarity* 'score function' (and score gradient) for a pair of objects in terms of a convolutions of their affinity functions (and affinity gradients). In Section 3.6 I transfer the convolution to the Fourier domain and present alternative formulae using Fourier transforms to streamline the computations. Lastly, the geometric energy, force, and torque are defined in Section 3.7 in terms of the the score function and its gradient.

- Chapter 4 describes the computer implementation of the method for haptic assembly. Section 4.1 presents the computational aspects pertaining to representational requirements, preprocessing steps, discretization details, and FFT computations. Section 4.2 follows by giving an overview of the software architecture along with auxiliary tools that are used to apply the presented ideas to a graphics- and haptics-enabled VE.

- Chapter 5 presents a few test-cases to validate the applicability of the method. It starts with simple peg-in-hole assembly problems in Section 5.1 that are

commonly used as benchmark examples in the literature. In Section 5.2 an assembly problem of more realistic geometric complexity is tested using the FFT-based convolution method whose performance is compared on the CPU and GPU. A simple analysis is then presented to come up with some rules for choosing the low-pass filtering threshold to enable a desirable simulation frame rate (e.g., 50 Hz for graphics and 1 kHz for haptics). Finally, I present other applications of the method such as protein docking in Section 5.3.

- The final chapter closes with the concluding remarks and future directions.

Chapter 2

Related Work

> "Just as the primitive man forged hand tools to triumph over harsh nature, we need to develop smart devices to interface with information-rich real and virtual worlds... In order to be efficient in this interaction [with information], it is essential that we utilize all of our sensorimotor capabilities."
>
> Mandayam A. Srinivasan, 1995 [372]

2.1 CAD/CAAP Applications

Today, most engineering design tasks are heavily assisted by powerful and widely available computer simulation and visualization tools. Although a large subset of analysis and synthesis tasks have been partially (if not fully) automated, the designer's decision-making remains central to certain aspects of the design process. This in turn creates a demand for more effective human-computer interfaces to explore more efficient, creative, and cost-effective design solutions in semi-automatic setups.

In particular, haptic assistance has been found useful in several design activities that can benefit from domain expertise and cognitive capabilities of human operators (which are hard to formalize for full automation), such as

- conceptual design [36, 37, 40] and aesthetic design [39, 45, 46];
- design review and functionality validation [111, 128, 292, 439];
- ergonomics and human factors evaluation [35, 47, 219, 367];

and many more reviewed in Section 2.1.1. In particular, haptic manipulation has also been leveraged for editing parametric CAD models and freeform surfaces for designing individual parts [107, 108, 139–141, 264–266]. The focus of this project, however, is on developing computational tools that enable applying haptics to the evaluation and planning of assembling and disassembling (already designed) rigid and flexible parts reviewed in Section 2.1.2—another critical step in the product design process, and certainly among the costliest steps not so long ago [33].

A thorough review of research efforts and organization of the literature on graphics- and haptics-assisted virtual assembly can be frustrating, as appreciating its current position and potential implications in the modern product life-cycle management (PLM) requires an overview of a range of different topics. In this section, I provide a sufficiently broad review—brief in text, less so in the number of citations—of some of the relevant research studies in applying virtual reality (VR) and augmented reality (AR) tools to computer-aided design and assembly planning (CAD/CAAP), virtual prototyping (VP) (Section 2.1.1), and virtual assembly (VA) (Section 2.1.2). Of course I do not intend, by any means, to provide a complete survey of VP/VA related research or all published industrial implementations. Rather, I shall focus on providing the reader with clear *definitions* and *classifications* of the existing approaches

along with introducing common terminologies, comparing advantages and drawbacks, and citing (more than enough) pointers if further details are sought. Particular attention is given to the different types of *constraints* (Section 2.1.3) that emerge in VR-CAD assembly and disassembly problems and are central to solving the parts' motion in real-time using physics-based (Section 2.2), constraint-based (Section 2.3), or combined (Section 2.4.2) techniques.

2.1.1 Virtual Prototyping

Recently, an early-stage examination of different product life-cycle aspects related to design, manufacturing, maintenance, service, and recycling has been made possible by integrating VR tools into the modern CAD environments, a practice referred to as virtual (or digital) prototyping (VP/DP) [61,112,164,411]. Such an evaluation results in a significant reduction of time and cost associated with physical prototyping (PP), and allows for the elimination of a large subset of design issues in the earlier stages of the process [103, 104]. Although they cannot yet completely replace physical prototypes, virtual prototypes are less expensive, more repeatable, and easily configurable for different variants, hence provide significant insight into the functionality of the product while eliminating redundant design trials and excessive tests [36].

In the larger context of modern PLM, one often encounters the notion of a digital mock-up (DMU). DMU consists of a complete digital descriptions of the product during its entire life-cycle, serves as a platform for product and process development, and includes geometric, ergonomic, and functional information—with or without human-computer interaction element. DMU construction accounts to realistic computer simulations that are capable of replicating different functionalities rang-

ing from design, manufacturing, maintenance, service, and recycling of the physical mock-up (PMU) [150]. The interactive application of immersive VR tools for a subset of those tasks, including (but not limited to) assembly and disassembly process verification, ergonomics and functional assessment, and other design evaluations are the subset of DMU development technologies referred to as VP [150].[1]

For reviews of PP versus VP techniques and their classifications, capabilities, and limitations for product development, I refer the reader to [28, 148, 450]. Currently, the most notable industrial applications of VP are found in the automotive and aeronautic industries [26, 109, 449]. Instructive (although not very up-to-date) surveys of manufacturing applications in general can be found in [292, 297, 369]. In particular, an assessment of the capabilities of VR hardware and software tools available in early 2000s to support VR integration into product life-cycle management (PLM) is given by Jayaram et al. [216].

Rather than focusing on particular application domains, here I briefly review several important and relatively recent studies and systems that use haptic support for a variety of design, analysis, validation, and manufacture process planning activities. The following review is by no measure comprehensive, but presents a good selection of a range of different research developments. Table 2.1.1 provides a more extensive and chronological list of studies along with their hardware and software components and key features. This review excludes the assembly-related applications which I review in greater extent in Section 2.1.2 and Table 2.1.2.

[1]The notions of physical/digital mock-ups (PMU/DMU) and physcial/digital prototyping (PP/DP) are sometimes used interchangeably, but it is safe to say that a more precise definition puts the latter as a subset of the former that involves VR technologies [150].

TABLE 2.1.1: A chronological review of haptic-enabled virtual prototyping platforms for various product development tasks and their key features.

References	Year	System	Methods	Software	Hardware	Key Features
[187] [186, 393, 394] [274, 301] [392]	1996 1997 1998 2005	—	PBM, CD.	Utah's Alpha-1, ControlShell, TCP and UDP.	Sarcos Dextrous Arm Master.	exoskeleton haptic interface for CAD; uses mesh and DPT for elastic CD; asynchronous networking with device.
[108] [107]	1999 2001	—	PBM, MSS, CD.	GHOST® API.	Phantom® 1.0.	freeform sculpting using B-splines; mass-spring discretization of surfaces; dynamic optimization of control points.
[284, 285] [70, 282] [281] [283]	2001 2002 2005 2007	Digital Sculpture	PBM, DSS, CD.	GHOST® API.	Phantom® 1.0.	'virtual clay' using subdivision solids; mass-spring discretization of lattice; dynamic optimization of mass points.
[124] [85] [123]	2000 2001 2005	—	PBM, CD.	FreeForm®.	Phantom® Desktop®.	concept generation via sketch elevation; form approximation; surface shaping; wire-cutting, smoothing, and mirroring.
[263] [264, 265] [266]	2003 2004 2005	Virtual DesignWorks	CBM, CD.	GHOST® API, MS COM+.	Phantom® Desktop®, Unspecified OST-HMD, 5DT FOB® OTs.	COM-based CAD-VR interoperability; freeform-based B-rep surface operations; shape control functions and SVD.
[435] [78, 80, 436] [81, 437, 439] [79]	2003 2004 2005 2007	—	PBM, RLE, FEM, CD.	FreeForm®, GHOST® API, VTK Toolkit.	Phantom® Desktop®, ABB IRB 1400.	uses mass-spring model for elastic CD; uses S-RLE description for plastic CD; both additive and subtractive forming; models milling and path generation.
[77] [82, 438] [412] [416]	2004 2005 2006 2009	HVCMM	PBM, RLE, CD.	GHOST® API,	Phantom® Desktop®.	uses S-RLE and custom model for CD; models CMM inspection path planning; models CMM accessibility analysis.
[138] [139, 140] [137, 141] [142]	2004 2005 2006 2007	—	PBM, MSS, CD.	GHOST® API.	Phantom® Premium®.	freeform sculpting using B-splines; mass-spring discretization of surfaces; dynamic optimization of control points; implicit modeling of prob/tool heads;
[100] [319–321] [370] [50]	2004 2008 2009 2010	VRAD	PBM, CD.	EVI3d Drivers, VEserver, OpenCASCADE, OpenGL.	CAVE-like System, head tracking devices, WTP haptic devices, IBM ViaVoice.	allows implicit edition of CHG data; models haptic selection and extrusion; enables multimodal interactions; future developments aimed at CATIA®.
[34] [36–38] [105] [40]	2004 2006 2007 2010	T'nD	PBM, CD.	Device APIs.	FCS-HapticMaster, nVisor ST OST-HMD, Vicon M2-460.	uses tesselated models for CD; uses chip removal theory for PBM; conceptual design and ergonomics; creates NURBS for downstream.
[34] [36] [35, 47]	2004 2006 2007	VeRVE	PBM.	Device APIs, UGS Jack.	Haptic Knob(s), nVisor ST OST-HMD, Vicon M2-460.	uses 'smart' haptic feedback (tacton); models ergonomic interactive testing.
[42] [43, 44] [65] [66, 67]	2008 2009 2010 2011	PROGIMM	PBM, CD.	3DVIA Virtools, KRL™+XML.	KUKA KRC and KR3, nVisor ST OST-HMD, Vicon M2-460.	mixed reality and mixed prototyping; ergonomic assessment of driver seats; haptic tools for automotive industry.
[39] [45, 111, 128] [46]	2008 2010 2012	SATIN	PBM, CD.	SML Manager, ThinkCore API, OpenIVI, TCP/IP.	FCS-HapticMaster×2, DVHDS Components, nVisor ST OST-HMD, Vicon M2-460.	haptic strip for curve approximation; various haptic modules and knobs; CAD modeling and shape analysis; conceptual and aesthetic design.
[260, 260] [262] [267]	2012 2014 2015	DesignWorks	CBM, CD.	GHOST® API, MS COM+, VBS Kernel.	Phantom® Desktop®, Unspecified HMD, 5DT FOB® OTs.	variational B-spline editing techniques; real-time energy minimization.

Abbreviations used for generic terms: physically-based modeling (PBM), constraint-based modeling (CBM), collision detection (CD), virtual environment (VE), virtual reality (VR, augmented reality (AR), computer-aided design (CAD), application programmer's interface (API), software development kit (SDK), transmission control protocol (TCP), user datagram protocol (UDP), internet protocol (IP), direct parametric tracing (DPT), mass-spring system (MSS), dynamic subdivision solids (DSS), optical see-through (OST), head-mounted display (HMD), boundary representation (B-rep), optical tracker (OT), singular value decomposition (SVD), run-length encoding (RLE), finite element method (FEM), spatial run-length encoding (S-RLE), coordinate measuring machine (CMM), wand-type pointer (WTP), construction history graph (CHG), nonuniform rational B-splines (NURBS), extensible markup language (XML).

Abbreviations used for brand names: general haptic open software toolkit (GHOST), haptic virtual coordinate measuring machine (HVCMM), virtual reality aided design (VRAD), cave automatic virtual environment (CAVE), touch and design (T'nD), virtual reality system for validation of equipment controls (VeRVE), la progettazione immersiva multisensoriale (PROGIMM), sound and tangible interfaces for novel product design (SATIN), Microsoft (MS), Component Object Model (COM), Flock of Birds (FOB).

×2 means two-handed haptic device or a pair of one-handed devices.

Literature Survey: Examples of Haptic-Enabled VP

Hollerbach et al. at the University of Utah developed a 'haptic display' for grasping and manipulating virtual mechanisms (e.g., linkages and chains) [274, 301] using an exoskeleton haptic device Sarcos Dextrous Arm Master—later upgraded to Sarcos DTS Master Exoskeleton for subsequent work [132]. They also integrated the haptic interface with Utah's Alpha-1 geometric modeling system to enable manipulation of both polygonal (i.e., mesh) and freeform (i.e., parametric) surfaces, particularly using direct parametric tracing (DPT) [394] for tracing untrimmed and trimmed NURBS surfaces [392, 393] and physics-based models—e.g., stick-slip friction [349] and nonlinear viscosity [276]—for rapid virtual prototyping [186,187]. Among other related works of the group is nonlinear device modeling for VR applications [96, 97].

Qin et al. at the State University of New York at Stony Brook developed a variety of haptic sculpting systems by applying physics-based modeling based on lumped mass-spring networks—made of 'control points' and 'mass points' connected within a control mesh and a network of springs—and Newtonian dynamics to different solid representations. The representations include B-spline surfaces discretized with linear springs over the control mesh [107, 108]; dynamic subdivision solids [280] discretized with both linear and angular springs over the control lattice (called the 'virtual clay' method)* [282,284,285]; volumetric implicit functions (i.e., 'density fields') used to define semialgebraic sets bounded by a finite number of B-spline patches discretized into a grid of 'density springs' [191–193]—which are also capable of performing CSG operations and knot insertion; and dynamic pointset surfaces by fitting implicit functions to local distance fields and applying ideas from implicit modeling [161]. The group later implemented the ideas from both volumetric subdivisions and implicit modeling into a system called DigitalSculpture [281] for interactive surface editing. Among their other relevant works are direct mesh editing using PDE-based geometric surface flow in a system called HapticFlow [114] and applying FEM to incorporate flexibility into subdivision solid geometry for haptic sculpting [283]. A Phantom® 1.0 device (3 DOF input, 3 DOF output) was used for all applications.

Liu et al. form the Queen's University of Belfast developed the first VR-CAD system called Virtual DesignWorks [263, 265] that used Microsoft' COM+ technology for real-time exchange of models between a CAD module (e.g., NURBS-based B-reps for flexible editing) and a haptic module (e.g., polygon/voxel-based for fast rendering) for freeform surface editions (e.g., pulling, pushing, and dragging). The COM-based implementation enabled real-time interoperability between a typical CAD kernel (e.g., used in SolidWorks®, Uni-

graphics®, etc.) and an approximate representation of geometry for haptics. In a follow-up study [264,265] they used 'shape control functions' to simulate surface deformations which yield a linear system of constraints solved by the singular value decomposition (SVD) method. Later in Hebei University of Technology, Liu also implemented variational B-spline (VBS) techniques and real-time energy minimization (using Hebei's VBS kernel) for surface hole filling [261,267] and for interactive surface editing [260,262] into DesignWorks.† A Phantom® Desktop® device (6 DOF input, 3 DOF output) was used for all applications.

Chen et al. at the University of Hong Kong developed a product development platform with a wide range of haptic functionalities [79–81] including machine tool path planning [435]—e.g., for 5-axis milling based on the method used in MIT Suzuki haptic system [12]; real-time mechanical property analysis [439]—e.g., using a hierarchical finite element method (FEM) from [224]; reverse engineering and shape digitizing [436, 437]—e.g., using haptic-guided volume sculpting method from [208]; and a module called HVCMM [77, 82, 412,416,438] for inspection path generation for coordinate measurement machines (CMM). The platform uses their own volumetric enumeration data structure called spatial run-length encoding (S-RLE) [78] for geometric rasterization and haptic rendering. The group's subsequent works also include haptic-guided repair of triangular meshes (e.g., hole-filling) [178], haptic-assisted evaluation of compliant mechanisms [385], and surface texture and friction modeling for tactile feedback [76]. A Phantom® Desktop® device (6 DOF input, 3 DOF output) was used for all applications.

Gao et al. form the University of Hong Kong and Gibson et al. from the National University of Singapore developed a haptic sketching system for manipulating 3D B-spline curves [138] and a haptic sculpting system [139–141] to create and modify B-spline surfaces using a variety of prob/tool head geometries. The sculpting system enables intuitive pushing and pulling operations on freeform surfaces to relate the virtual modeling experience to the physical world experience. The model uses a mass-spring discretization of the surface checked against the implicit representation of the prob profile for elastic collision response. The implementation also enables wavelet-based multiresolution representations, which in turn enables sweep editing and 3D texture reuse in the frequency domain [141]. The system was later added with functionalities to design and evaluate multimaterial products [137] and to work with point clouds and NURBS [142]. A Phantom® Premium® device (6 DOF input, 6 DOF output) was used for all applications.

The VENISE research team (Bourdot et al.) at the CNRS/LIMSI laboratory in partnership with Universite Paris-Sud developed an integrated and immersive VR-CAD system called VRAD [50] that supports modification of CAD semantics directly within the VE.

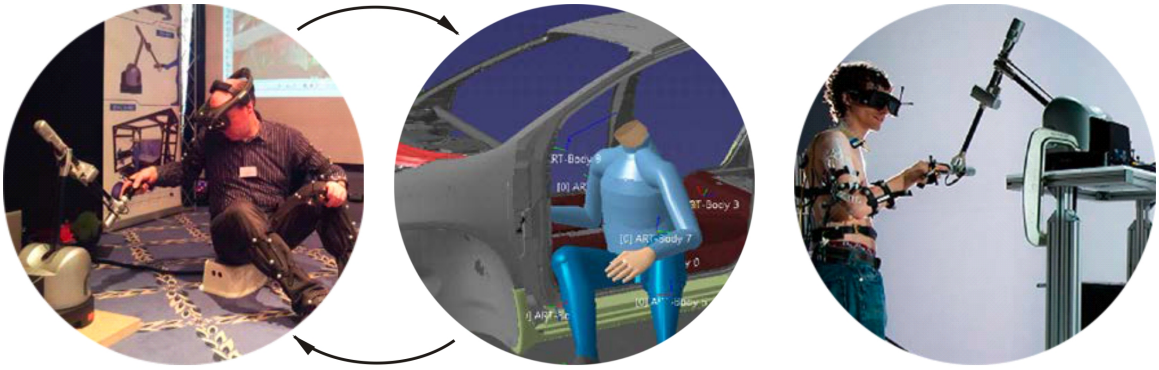
The system enables direct modification of construction history graphs (CHG) by selection of primitive elements on the evaluated B-reps [100]. The haptic functionality [319] was added using a variety of force models—namely, spring elasticity and attraction force models for haptic selection [320], 'reference-based', 'haptic-grid', and graduated virtual fixtures (GVF) methods for haptic extrusion [321], and potential field approaches for haptic selection [370]. The implementation supports multimodal interactions using previous tools developed by the group such as EVI3d [51, 398] including multimodal 'fusion' using their own VEserver [397] concurrently running on multiple computers, and multimodal 'fission' for managing large and complex data in future work [53]. Among other works of the group is flexible model rendering [101]. The device brands were not specified.

The KAEMaRT research group (Cugini et al. and Bordegoni et al.) at the Politecnico di Milano developed a system called T'nD [36,37,40] to perform conceptual design activities using virtual tools attached to a 6DOF FCS-HapticMaster device to simulate physical form-making activities—e.g., surface scraping using rakes [38] and finishing using sandpaper [105] modeled using chip removal theory [289]. The group also developed other prototypes such as VeRVE [35, 36, 47] for ergonomic validation using haptic knobs [95], and PROGIMM [42–44] as a mixed-reality platform for validating prototypes in the automotive industry in collaboration with Caruso et al. [65–67]. These conceptual design and ergonomic validation functionalities were also integrated, in addition to aesthetic design tools [39, 45, 46] into a multimodal and multisensory system called SATIN [111, 128] using a deformable strip attached to two FCS-HapticMaster devices installed in a parallel configuration and a display system called DVHDS composed of projectors and mirrors for superimposing the virtual scene over the user's hands.

Table 2.1.1 presents a more inclusive list (including studies and systems not described above) and a summary of their key features.

*Rossignac et al. at the Georgia Institute of Technology used a similar project name (the 'digital clay' project) [49, 144, 345] for the development of a new type of haptic interface for finger sculpting; namely, a computer-controlled physical surface that deforms in response to the pressure changes exerted by bare hands [144, 345], built using 'formable crust architectures' [49]. Ishii et al. at the MIT Media Lab recently developed a similar concept called InForm [250, 251] for 'physical telepresence' and remote collaboration.

†The later versions of Virtual DesignWorks [263, 265] was called DesignWorks [260, 262] in subsequent publications.



(a) (b) (c)

FIGURE 2.1.1: Application of haptic technology to virtual assembly and prototyping. The pictures are courtesy of Perret et al. [316].

2.1.2 Virtual Assembling

Among other VP activities, virtual assembly (VA), defined as a simulated assembly of the virtual representations of mechanical parts in an immersive 3D user interface using natural human motions [231, 232], characterizes an important subarea of VP, to which applying haptic feedback has been shown particularly beneficial in terms of task efficiency and user satisfaction by multiple researchers [150, 258, 408, 427]. For instance, a user survey carried out in BMW by Gomes and Zachmann [150] predicted an important role for VR tools in prototyping and assembling activities in the future of the automotive industry. In particular, they showed that VR-enabled DMUs reduce the need for PMUs and facilitate an improvement of the overall product quality. However, the study concluded that a widespread use of VR in manufacturing industries is contingent upon its seamless and complete integration into CAD/CAAP. Volkov and Vance [408] showed that haptic assistance improves the ability of a user for design decision making in VP/VA environments; particularly in terms of task efficiency (e.g., less time taken for user evaluation) and user satisfaction when evaluating

automotive design examples. Lim et al. [258] showed that small (i.e., visually insignificant) geometric features—e.g., chamfer or fillets in a simple peg-in-hole pair—can significantly affect user performance in haptic assembly, with measurements showing a similar trend to those of physical assembly. Their results demonstrate that adding haptic feedback to the VP/VA process enables exploiting shape information that are significantly underused when only visual feedback is provided. Wildenbeest et al. [427] conducted experiments to investigate the impact of haptic feedback quality in the performance of teleoperated assembly in the context of four sub-tasks; namely, free-space movement, contact transition, constrained translational, and constrained rotational tasks. They concluded that low-frequency haptic feedback improves overall task performance and control effort in constrained translational and rotational tasks.

In the past two decades, there have been numerous studies and systems focused on the development of immersive virtual environments for solving assembly and disassembly problems. These systems have used a variety of visualization tools (e.g., stereoscopic displays and goggles) and tracking devices (e.g., head tracking devices and data gloves) to assist the user in virtual object manipulation tasks. More recently, an increasing number of studies have leveraged haptic devices to provide a more realistic assembly experience with force feedback, a thorough account of which would constitute enough material for a full book on the subject. Instead of repeating them, I refer the reader to the following survey papers:

- See Seth et al. [366] for a comprehensive review of earlier studies and existing systems on VP/VA with and without haptic feedback.

- See Xia et al. [433] for a review of new ideas and future directions in VR and particularly haptics for product assembly from rigid parts to soft cables.

TABLE 2.1.2: A chronological review of haptic-enabled virtual prototyping platforms for assembly and disassembly tasks and their key features.

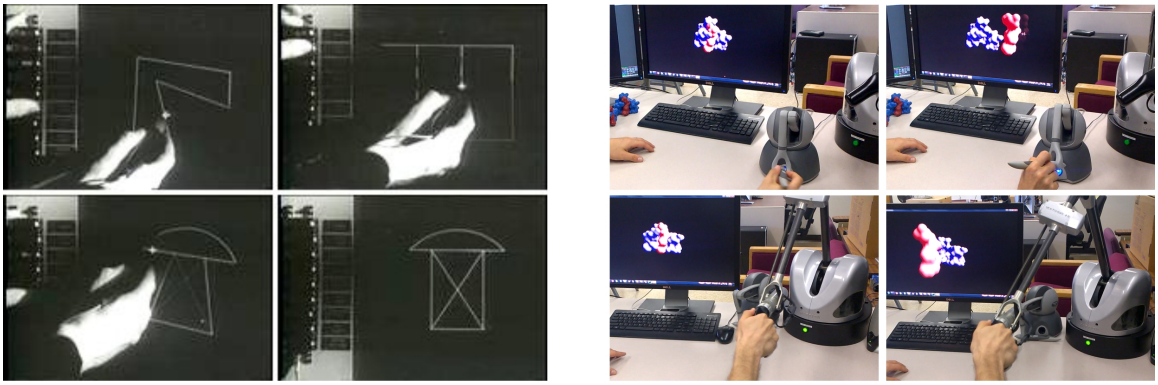
References	Year	System	Methods	Software	Hardware	Key Features
[165] [163, 164]	1995 1997	VEDA	PBM, CD.	GHOST® API,	Phantom® 1.0×2, SGI® Indigo™.	'vertex-edge' pair contact modeling; uses Coulomb's law of friction; limited to 2D convex/concave polygons.
[212] [215, 217, 434] [214, 220] [216, 413] [368, 415] [216, 367] [219] [213]	1997 1999 2000 2001 2003 2004 2006 2007	VADE	PBM, CBM, CD.	Device APIs.	CyberGrasp®×2, SGI® Onyx2™, InfiniteReality Pipes, 5DT FOB® OTs, Unspecified HMD, ImmersaDesk™.	one- & two-handed dexterous assembly; approximate line-polygon intersections; maintains link between VE and CAD; imports "assembly intent" from CAD; sweep generation and trajectory editing; simulates fasteners and screwdrivers; ergonomic evaluation tools; collaborative/distributed assembly.
[279] [104] [103]	1999 2001 2002	HIDRA	CD.	GHOST® API, OpenGL, Motif™ Qhull, V-Clip, SWIFT(++).	Phantom® 1.0.	CD-based assembly and disassembly; uses Qhull for convex hull generation; uses V-Clip and SWIFT(++) for CD; limited use to simple peg-in-holes.
[353] [48]	2002 2004	REVIMA	PBM, CD.	Device APIs, OpenGL.	CEIT LHIfAM.	maintenance of aircraft engines; accessibility and interference analysis.
[409, 447]	2004	MIVAS	PBM, CBM, CD.	OpenGL Optimizer, Pro/TOOLKIT® IBM ViaVoice, TCP/IP.	CyberGrasp®, SGI® Onyx2™, CAVE-like System CrystalEye SG, 5DT FOB®.	multimodal VE for grasp+assembly; direct constraint import from CAD; allows assembly sequence generation; allows assembly trajectory generation; optimization techniques for assembly.
[361] [362] [364] [363] [365]	2005 2006 2007 2008 2010	SHARP	PBM, CBM, CD.	VR Juggler, OpenGL PSG, OpenHaptics®, VPS™, D-Cubed, TCP/IP.	Phantom® Omni®×2, Barco Baron PT, CAVE-like System, Unspecified HMD.	uses VPS™ for high-clearance CD; uses D-Cubed for low-clearance CD; swept volumes for maintainability; capable of creating subassemblies; capable of networking with others.
[254, 255] [256, 258] [257] [152] [151, 153, 154]	2006 2007 2010 2013 2014	HAM(M)S	PBM, CD.	OpenHaptics®, VTK Toolkit, PhysX™ SDK, Bullet Physics, MFC library.	Phantom® Omni®×2, Phantom® Desktop®.	TCT evaluations for performance; EMG evaluations for motor control; MTL & therblig analysis of motion; multiple PSEs for PBM+CD; assembly process modeling & planning.
[203, 204] [206] [205]	2006 2006 2006	(C)HAS	CBM, CD.	OpenHaptics®, Labein's DATum, RAPID.	Phantom® Omni®, Phantom® Premium®, PERCRO GRAB.	distributed & collaborative assembly; combines assembly & haptic simulators; tested on an aeronautical assembly.
[198] [199] [200] [202] [201]	2007 2008 2011 2012 2013	CVE	PBM, CBM, CD.	VirtuoseAPI, VTK Toolkit, ODE.	Virtuose6D™ 35-45, 2D Wall Display, Crystal Eye SG*, Christie HD3 SP.	collaborative VE; modular behavior; automatic contact constraint detection; mobility trajectory characterization; classifies simple 'functional surfaces'; uses dual-quaternion representation.
[30] [31, 221]	2008 2009	HIIVR	PBM, CD.	Device APIs, SmartCollisions®.	Phantom® Omni®, 5DT FOB® & DGs, eMagin® Z800 HMD, NEC® SP.	used for procedural skills development; different difficulty levels for training; assists the user by visual cues; evaluated using SE & PVE scales;
[91] [92] [90]	2008 2009 2011	—	CBM, CD.	CHAI3D,	Phantom® Omni®.	optimization of path & sequence; AABB CD for 2D polygonal shapes; improved assembly (time & distance).
[143, 441]	2009	VEDAP-II	PBM, CD.	VHT Toolkit, PhysX™ SDK, OpenGL, V-Clip.	CyberGlove®, CyberGrasp®, CyberForce®,	models grasp+move+locate+secure; focuses on virtual coupling dynamics; models 'visual dynamic behavior'.
[41]	2009	—	PBM, CD.	VirtuoseAPI, Cyviz Viz3D™. Dassault® Virtools, ARTrack™, IPP.	Virtuose6D™ 35-45, WiiRemote™, Unspecified OTs.	evaluation of two-handed assembly; heuristic criteria for quality assessment; low- vs. high-cost device assessment. real-scale projection and tracking.
[376]	2009	—	CD.	Novint SDK,	Novint Falcon®	telepresence for in-space assembly; uses 'virtual walls/boundaries' for CD; tested on NASA's SPHERES testbed.

References	Year	System	Methods	Software	Hardware	Key Features
[246] [245]	2009 2010	—	PBM, PP, CD.	VirtuoseAPI,	Virtuose6D™ 35-45,	PRM PP methods (A-star, RDT/RRT); guides parts along 'following zones'; updates the path when user deviates;
[445] [430]	2009 2012	DPVAE	PBM, CBM, CD.	Device APIs, TCP/IP.	CyberGlove®, CyberTouch®, 5DT FOB® & DGs.	collaborative assembly environment; supports data conversion from CAD; parallel rendering based on PC-cluster; 'high-efficient' CD and HLA/RTI.
[190, 446]	2010	GCVAE	PBM, CBM, CD.	Device APIs, TCP/IP.	CyberGlove®, CyberTouch®, 5DT FOB® & DGs.	collaborative assembly environment; network gridbased support platform; supports large and complex scenes;
[167]	2010	IMA-VR	CBM, CD.	OpenHaptics®, Device APIs.	Phantom® Omni®, CEIT LHIfAM, PERCRO GRAB.	multimodal assembly training system; cognitive and motor skills transfer; spring-damper model for CD response.
[389, 390]	2010	—	PBM, CBM, CD.	VirtuoseAPI.	Virtuose6D™ 35-45,	VCG method for hybrid PBM+CBM; uses nonsmooth rigid body dynamics; uses virtual fixtures for DOF-limiting;
[172] [173, 174] [175]	2010 2011 2014	MAD	PBM, CD.	OpenHaptics®, OpenInventor® GLUT.	Phantom® Omni®.	aircraft assembly and maintenance; CACO path & sequence optimization; active and passive haptic guidance.
[146, 418, 419] [166] [417] [145]	2011 2012 2013 2015	—	PBM, CD.	Device APIs.	Haptic bracelet, Generic webcam, Generic tablet.	AR for maintenance & assembly; enables skills acquisition & transfer; AVA and vibrotactile feedback; low-cost AR training platform.
[431]	2011	HVAS	PBM, CBM, CD.	OpenHaptics®. WTK Toolkit, PhysX™ SDK.	Phantom® Premium®, Unspecified SG*.	combined PBM+CBM for assembly; automatic data transfer from CAD; hierarchical constraint data model.
[432]	2012	HITsphere	PBM, CBM, CD.	OpenHaptics®. WTK Toolkit, PhysX™ SDK, TechViz XL.	Phantom® Premium®, Cybersphere System, Unspecified SG*.	motion simulator for free walking; combined PBM+CBM for assembly; automatic data transfer from CAD; hierarchical constraint data model.
[298]	2013	Poster	PBM, CD.	PTAMM, ODE, ARToolKit.	HapticGEAR, Unspecified HMD,	wearable backpack-type haptic device; markerless AR with large workspace.
[311]	2013	Snap-to-Fit	PBM, CD.	H3DAPI,	Unspecified Device,	point-to-point attraction force model; applied to facial surgery & archaeology.
[18, 19] [22, 23] [24]	2014 2015 2016	—	PBM, CD, GE.	OpenHaptics®, VirtuoseAPI, OpenGL, Havoc3D. GLUT, Win32 API.	Phantom® Omni®, Virtuose6D™ 35-45.	unified CD and geometric guidnace; automatic GE for arbitrary geometry; does not depend on CAD constraints; does not scale with syntactic size;

Abbreviations used for generic terms: physically-based modeling (PBM), constraint-based modeling (CBM), collision detection (CD), virtual environment (VE), virtual reality (VR, augmented reality (AR), computer-aided design (CAD), application programmer's interface (API), software development kit (SDK), transmission control protocol (TCP), user datagram protocol (UDP), internet protocol (IP), optical see-through (OST), head-mounted display (HMD), boundary representation (B-rep), shutter glasses (SG), stereo glasses (SG*), projection table (PT), task completion time (TCT), electromyography (EMG), motion timeline (MTL), physics simulation engine (PSE), data glove (DG), stereo projector (SP), self-efficacy (SE), perceived virtual environment (PVE), axis-aligned bounding box (AABB), optical tracker (OT), probabilistic roadmap (PRM), path planning (PP), rapidly-growing deterministic trees (RDT), and rapidly-exploring random trees (RRT), high level architecture (HLA), runtime infrastructure (RTI), virtual constraint guidance (VCG), degrees of freedom (DOF), combined ant colony optimization (CACO), adaptive visual aids (AVA), geometric energies (GE).

Abbreviations used for brand names: virtual environment for design for assembly (VEDA), general haptic open software toolkit (GHOST), Silicon Graphics, Inc. (SGI), virtual assembly design environment (VADE), Flock of Birds (FOB), haptic integrated dis/reassembly analysis (HIDRA), open graphics library (OpenGL), Voronoi clip (V-Clip), speedy walking via improved feature testing (SWIFT), realidad virtual para el estudio de mantenibilidad en sistemas aeronàuticos (REVIMA), large haptic interface for aeronautic maintainability (LHIfAM), multimodal immersive virtual assembly system (MIVAS), cave automatic virtual environment (CAVE), system for haptic assembly and realistic prototyping (SHARP), Voxmap PointShell (VPS), haptic assembly, manufacturing, and machining system (HAMS), haptic assembly and manufacturing system (HAMS), visualization toolkit (VTK), Microsoft Foundation Class (MFC), perceptual robotics (PERCRO), haptic assembly simulator (HAS), collaborative haptic assembly simulator (CHAS), open/object dynamics engine (ODE), haptically enabled interactive and immersive virtual reality (HIIVR), computer haptics and active interfaces 3D (CHAI3D), virtual environment for the design and assembly planning (VEDAP), virtual human toolkit (VHT), interactive physics pack (IPP), National Aeronautics and Space Administration (NASA), synchronized position hold, engage, reorient, experimental satellites (SPHERES), distributed parallel virtual assembly environment (DPVAE), gridenabled collaborative virtual assembly environment (GCVAE), industrial maintenance and assembly with virtual reality (IMA-VR), maintenance assembly/disassembly (MAD), graphics library utility toolkit (GLUT), haptic-based virtual assembly system (HVAS), world toolkit (WTK), parallel tracking and multiple mapping (PTAMM).

×2 means two-handed haptic device or a pair of one-handed devices.



(a) (b)

FIGURE 2.1.2: The evolution of human interaction for CAD: from Sketchpad [383] (a) to current graphic- and haptic-enabled systems (b).

- See Leu et al. [252] for a state-of-the-art report on the methodologies for developing CAD-VR systems for assembly simulation, planning, and training.
- See Vance and Dumont [402] for a popular proposal of the expected future directions in haptic assembly—to which I will return in Section 2.4.2.
- See Perret et al. [316] for a discussion of the scientific challenges and technical issues faced in haptic assembly along with an assessment of the technical maturity using the technology readiness level (TRL) index.

Other decent reviews on the subject from multiple points of view—followed by presentations of specific systems or methods—can be found in [202, 257]. Here I present a brief review of a fairly sizeable collection of systems for haptic-enabled assembly and disassembly activities along with their key features and differences in terms of modeling and implementation. Once again, the following review is by no measure comprehensive, but presents a good selection of a range of different research developments. Table 2.1.2 provides a more extensive and chronological list of studies along with their hardware and software components and key features.

Literature Survey: Examples of Haptic-Enabled VA

Gupta et al. at the MIT Media Lab developed one of the earliest multimodal desktop VA systems (i.e., with visual, auditory, and touch feedback) called VEDA [163–165] that used physically-based modeling (PBM) for part behavior, with the end-goal of integrating design evaluation techniques such as design for assembly (DFA) into the existing CAD systems. The shapes were limited to either convex or concave rigid 2D polygons whose contact was modeled using Coulomb's laws of (static and dynamic) friction. Dual Phantom® 1.0 devices (6 DOF input, 3 DOF output) were used. The system's performance was evaluated for simple peg-in-hole examples.

Jayaram et al. at the Washington State University in a research partnership with Lyons and Hart from the National Institute of Standards and Technology (NIST) developed the immersive assembly planning and evaluation platform called VADE [212,215,217]. The system directly imported CAD models from Pro/ENGINEER® and performed assembly using constraint-based modeling (CBM) [415] in addition to some basic PBM. It was further augmented with additional functionalities including interactive swept volume generation and modification using synchronous links between the VE (providing trajectory data) and the CAD (providing shape data) sub-systems [434], inclusion of hand-held tools (e.g., screwdrivers) and the corresponding assembly operations and tool/hand/part interactions [214], and distributed environments for collaborative assembly [220]. Additionally, workplace ergonomic evaluation tools—e.g., the rapid upper-limb assessment (RULA) algorithm [278]—were integrated later into the system [367,368]. Subsequent industrial case studies [213, 218] concluded the feasibility of VA methods for deployment in the actual PLM cycle, and identified the key issues to be resolved in terms of ease of use, portability of the applications, and preparation of the evaluation models. Both one- and two-handed assembly was enabled using CyberGrasp® haptic gloves (5 DOF input, 5 DOF output).

Bras et al. at the Georgia Tech. developed another early haptic-enabled assembly and disassembly simulation environments called HIDRA [103, 104, 279]. The implementation used two concurrent loops (one for graphics and one for haptics+CD) to comply with the difference in frame rate requirements. To allow for fast CD, they used University of Minnesota's Qhull library for computing the convex hull of parts (or an obvious decomposition of the parts). For the CD itself between convex elements, Mitsubishi Electric Research Laboratory (MERL)'s V-Clip library [290] was used in the earlier versions of HIDRA [104,279]. Later, to enable faster multibody processing, sweep generation, and prune sorting, the use

of SWIFT [119] and SWIFT++ [120] was investigated in [103], where the ++ version enables nonconvex CD by decomposing into convex elements. The study discusses optimization techniques such as part anchoring, dynamic loading of haptic representations, and partitioning of part updates to minimize the effects of inherent computational limitations. The results of the study indicated limited usability of haptics for simple peg-in-hole examples. A Phantom® 1.0 device (6 DOF input, 3 DOF output) was used.

Borro et al. at the CEIT research group (CEIT-IK4 research alliance as of 2004) developed the system called REVIMA [48,353] for haptic-assisted maintainability simulation in aeronautics (e.g., assembling aircraft engine mock-ups). A specialized large haptic device called LHIfAM (6 DOF input, 3 DOF output) was developed for this purpose.

Wan et al. at the Zhejiang University developed a multimodal VE system for assembly named MIVAS [409,447] with voice input, stereo visual feedback (CAVE-like system), sound feedback, and force feedback. To facilitate an automatic translation of the assembly model from Pro/ENGINEER® CAD models, an interface was developed using Pro/TOOLKIT® which automatically imports geometry and topology (for PBM) as well as assembly constraints (for CBM) into the VE. It also incorporated models of virtual hand kinematics and grasping heuristic patterns for realistic user interaction. Virtual hand/part CD for grasp feedback was carried out using RAPID [155] while fast part/part CD was implemented using VPS™ [286–288, 410]. CyberGrasp® haptic gloves (5 DOF input, 5 DOF output) were used to enable force feedback from the virtual hand.

Vance et al. at the Iowa State University developed a series of VA tools with and without haptic support: Johnson and Vance developed VEGAS [223], an assembly simulator that used Boeing Corp.'s Voxmap PointShell (VPS)™ library [286–288,410] for CD between high-polygon parts without PBM for haptic feedback. Kim and Vance investigated different CD and part behavior algorithms [231, 232] and modified VEGAS to include PBM but no haptics. Kim and Vance also developed NHE [233] to facilitate collaborative assembly over the web using a combination of peer-to-peer and client-server models. Howard and Vance [188] developed a prototype desktop system for haptic assembly using PBM. From the same group, Seth et al. later developed SHARP [361–365] which expanded the VA functionality to include dual-handed haptics, swept volume representation, subassembly modeling, and more realistic part behavior via PBM. The earlier versions of SHARP [361–363] used VPS™ for approximate CD between a 'voxmap' representation of an stationary part and a 'pointshell' representation of the moving part, which was inadequate for low-clearance assembly. The subsequent versions of SHARP [364, 365] implemented a combination of PBM and CBM using the so-called automatic geometric constraints (AGC) method [402];

namely, it used D-Cubed's CDM module for exact CD between original CAD B-rep data and D-Cubed's DCM module for constraint management. Faas and Vance [125] proposed a hybrid method to tie the B-rep and voxel-based representations for simultaneous collision response and constraint-based guidance. Dual Phantom® Omni® devices (6 DOF input, 3 DOF output) were used.

Cheng-Jun et al. at the Qingdao Technological University proposed the use of dynamically constructed oriented bounding box (OBB) tree-based CD [155] for haptic assembly [84]. They also proposed a 'increment-along-constraint' (IAC) method [83] to solve the separation problem between the haptic proxy and the parts under constrained motion. A Phantom® Desktop® device (6 DOF input, 3 DOF output) was used, for which the specialized OBB and IAC algorithms were implemented.

Lim et al. at the Heriot-Watt University developed a system called HAM(M)S [152, 154, 254–258] as a testbed to investigate and measure user interactions and response while performing various engineering tasks in a haptic-enabled VE including assembly.* Different physics simulation engines were utilized; namely Ageia PhysX™ SDK [68] was used in the earlier versions [256] and Bullet Physics SDK [348] was added into the later versions [152, 154], whose pros and cons were evaluated and compared in [151,153]. They conducted peg-in-hole assembly experiments (using insertion routines with and without chamfers) on both real and virtual setups for comparison in terms of task completion times (TCT) [258] and motor control measured via muscle Electromyography (EMG) [257]. Further investigation of the user-object interaction was made with the objective of assembly plan generation by analyzing chronocycle-graph motion timelines (MTL) and "therblig" units [381] for both peg-in-hole examples and a more realistic pump assembly [339–341]. Phantom® Omni® and Desktop® devices (6 DOF input, 3 DOF output) were used interchangeably. The group also developed a cable harness VR system called COSTAR [380–382] (without haptic feedback) for user-logging in order to analyze the design process, capture design knowledge, and produce assembly plans.

The Fundación LABEIN research team (Iglesias et al.) at Tecnalia developed the distributed haptic assembly system HAS [203] using their own geometric modeler DATum for creation of 3D virtual scenes from CAD models. An assembly simulator was developed based on RAPID [155] for CD and simple semantics for constraint detection and enforcement; namely, along co-axial axes and co-planar planes. However, it did not include real-time physics-based simulation. The collaborative extension called CHAS [204–206] using a peer-to-peer architecture and specific consistency maintenance schemes was developed later and tested on an aeronautical assembly test-case. Phantom® Omni® and Premium®

devices (6 DOF input, 3−6 DOF output) and a PERCO GRAB device providing two points of contact (4 DOF input, 4 DOF output) were used.

Christiand et al. at the Gyeongsang National University developed an assembly simulation system [90–92] with haptic guidance along an optimized path. The sequence identification and path planning were carried out offline using an optimization genetic algorithm (GA) that took into account the part geometries and gripper data. Once a sequence was identified and a path was generated for each part in the sequence, a potential field method was used in real-time to combine repulsive and attractive forces to avoid obstacles and guide each part to its final position along the known path. The implementation was limited to the assembly of 2D polygonal parts whose collisions were detected using an axis-aligned bounding box (AABB) method. The experiments concluded better result with haptic guidance compared to those without haptics, both in terms of assembly time and travel distance. Hassan and Yoon from the same group developed the assembly and disassembly maintenance system called MAD [172]. The system imported CAD files from CATIA® using OpenInventor® API and performs sequence and path planning using parallel GA-based planners. Given the initial and final configurations, the GA minimizes the number of required gripper exchanges and orientation changes to reduce the assembly time according to the findings in [314]. To accommodate the geometric complexity and high-DOF that led to the failure of GA, Hassan and Yoon later developed a two-stage combined ant colony optimization (CACO) algorithm [174], which performed sequence optimization in the first stage followed by path planning in the second stage. The CACO algorithm was introduced to MAD [173] to investigate the effects of active and passive haptic guidance to improve the user performance. A Phantom® Omni® device (6 DOF input, 3 DOF output) was used for all experiments.

The KAEMaRT research group (Cugini et al. and Bordegoni et al.) at the Politecnico di Milano evaluated haptic-assisted manual assembly in a mixed reality environment for the case study of grabbing, holding, and positioning pairs of mechanical components relative to each other [41]. A Virtuose6D™ 35-45 (6 DOF input, 6 DOF output) and a Nintendo™ WiiRemote™ were used together in a two-handed setup. They used a set of heuristic criteria for assessing the quality of the application and identifying usability problems to be fixed in future studies.

Ladeveze et al. at the Université de Toulouse developed a system for haptic assembly and disassembly task assistance [245, 246] using probabilistic roadmap (PRM) path planners—namely, tools such as the A-star algorithm, rapidly-growing deterministic trees (RDT), and rapidly-exploring random trees (RRT) [4]. The planner first identified a colli-

sion free path in the 6D configuration space of a rigid part. The haptic control loop then used that information to guide the user into and along a so-called 'following zone' formed around the free path, which was discretized for simplifying and stabilizing force and torque computation. A Virtuose6D^{TM} 35-45 device (6 DOF input, 6 DOF output) was used.

Wu et al. at the Shanghai Jiao Tong University developed a grid-based VA server called GVAS [446] for large and complex (e.g., automotive and ship) assemblies, which used parallel computing and network resources for demanding VA computations such as model rendering, image processing (i.e., fusion), and CD. To fulfill security requirements, product data was managed independently using the concept of role-based access control (RBAC). The group later developed two systems for automobile VA: 1) a grid-based collbaorative VA environment called GCVAE [190], which comprised of a grid-based support platform, a service-based parallel rendering framework, and a multi-user collaborative VA environment; and 2) a distributed parallel VA environment called DPVAE [430, 445], which used high-level architecture and runtime infrastructure (HLA/RTI) event synchronization mechanisms. The systems made extensive use of parallel processing to enable complex assembly scenarios with intensive rendering and CD requirements at multiple levels of detail (LOD). CyberTouch® haptic gloves (with small vibrotactile feedback) were used.

Tching et al. at the IRISA–Bunraku developed the virtual constraint guidance (VCG) method [389,390] for haptic guidance. The method decomposes a task into 1) a guiding step which use virtual fixtures [343] to guide the objects into position; and 2) a functional step which use kinematic constraints via mechanical joints to restrict the DOF for insertion. The idea is to use both PBM-based exploration of the VE (using nonsmooth dynamics [387]) and CBM-based execution of fine insertion (using virtual fixtures [343]) while CD is locally deactivated. The method has proved very effective for peg-in-hole test-cases with simple contact constraints (i.e., lower pairs) whose virtual fixture abstractions and mechanical joint equivalents are obvious. The constraints were extracted from CAD mating pair semantics in a preprocessing step. A Virtuose6D^{TM} 35-45 (6 DOF input, 6 DOF output) was used.

Xia et al. at the University of Porto developed a multithreaded haptic assembly system called HVAS [431]. The system had an automatic data integration interface to transfer geometry, topology, assembly, and physics information from CAD systems (e.g., Pro/ENGINEER® and SolidWorks®) to the VE. A hierarchical constraint-based data model and scene-graph structure was designed to construct the VA environment. The system employed a combined PBM and CBM approach for haptic guidance; namely, a spring-damper model for collision response and dynamic simulation when parts penetrate into each other, and a geometric guidance model to generate attractive and repulsive forces

to guide the mating parts that are close to each other. The architecture was embedded into another system called HITsphere [432] for VA training which, unlike the traditional desktop or CAVE-like systems, also enabled natural human walking motion—similar to the Cybersphere system [126]—using a low-cost motion simulator. Haptic feedback was enabled by a Phantom® Premium® device (6 DOF input, 6 DOF output).

The G-SCOP group (Noël et al.) at the Grenoble Institute of Technology and Iacob eta al. at the University Politehnica of Bucharest developed a collaborative VE called CVE [322, 399] for design and assembly activities, which contained several modules—such as viewer, recorder, editor, object dynamics engine (ODE), analysis, and haptic models—integrated together for different sub-behaviors. The system performed automatic constraint detection using a contact identification process between 'functional surfaces'—restricted to planar, cylindrical, spherical, and conical surfaces—between CAD models (exported in STEP format) [198–202]. The system was evaluated for usability, efficiency, user experience, and feedback quality. A stereoscopic display and a Virtuose6D^{TM} 35-45 device (6 DOF input, 6 DOF output) was used for haptic feedback.

Table 2.1.2 presents a more inclusive list (including studies and systems not described above) and a summary of their key features.

*The system was called haptic assembly, manufacturing, and machining system (HAMMS) in the earlier publications [152, 254]. The authors chose to simplify the system's name to haptic assembly and manufacturing system (HAMS) in a subsequent paper [154].

As reviewed above, there is a bewildering variety of VR systems implemented using different software libraries and computational tools for haptic-assisted assembly and disassembly activities. In spite of their major differences in terms of setup (e.g., desktop versus CAVE-like systems), architecture (e.g., PC versus network-based systems), and hardware (wand-type versus glove-type devices), the underlying geometric and physical modeling of most systems are quite similar; namely, they all use the following basic tools:

- collision detection (CD) to avoid penetration between parts and subassemblies, typically available as part of the physics simulation engines (PSE), which can be conceptualized in terms of 'physical constraints';

- heuristic attraction and/or repulsion forces and torques formulated using spring-damper models or some other simple potential energy field derived from CAD model's mating pairs called 'geometric constraints';

or a blended combination of both. But as I shall articulate in the rest of this chapter, CD-based physical constraints are difficult to resolve in the vicinity of tight (i.e., low-clearance) fits for a variety of reasons (see Section 2.2), while the artificial geometric constraints are too simplistic to handle arbitrary geometry (see Section 2.3). The following section clarifies the conceptual distinctions between different types of constraints and their implications for VR implementations.

2.1.3 Constrained Motion

The simulation of assembly and disassembly processes for rigid parts can be abstracted as a 6 DOF free motion (3 for translations and 3 for rotations) per part along with a set of *constraints* that restrict motion along those DOFs and create interdependencies across different parts. The existing approaches for simulating 'part behaviour' in VR systems are typically classified into two groups, with the following definitions [152]:

- physically-based modelling (PBM), which uses Newtonian/Lagranigian dynamics to solve for the motion trajectories of the virtual objects, under the effect of forces and torques due to physical contact between those objects (e.g., no-penetration impact forces, sliding friction forces, etc.) and environmental effects (e.g., gravity, viscosity, etc.); and

- constraint-based modeling (CBM), which uses additional geometric constraints to locate the parts in the assembly configuration by artificially reducing the

DOF of the manipulated objects, similar to a CAD system.

Although such a definition is popular in the literature with slightly variant articulations, I find it imprecise and often misleading. This becomes clear by noting that motion dynamics is essentially a constrained optimization problem in disguise, hence there is no fundamental distinction between PBM and CBM as defined by the above statements. In particular, rigid body dynamics is given by the Lagrangian formulation of the equations of motion, where the requirement of no-collision between solids is equivalent to a holonomic unilateral constraint [270]. Impulse forces and torques then originate from the Lagrange multipliers associated to these constraints by Gauss' principle of least constraint [293, 326]. Therefore, there is no fundamental difference, in terms of the underlying physics and mathematics, between finding the motion trajectories under the effect of contact forces and torques (i.e., PBM), on the one hand, and limiting the motion DOF by additional constraints (i.e., CBM) on the other hand. In fact, most implementations use both PBM and CBM, even though one of them might appear as the dominant theme.

One can still draw a more meaningful classification based on how the constraints are formulated in practice from a knowledge of individual part geometries and their spatial relations. Based on a review of different techniques in the literature, I believe that the following provides a more precise and consistent definition with respect to the existing methodologies:

- The first approach uses 'physical constraints' defined as the set of constraints that arise *organically* from part geometries (e.g., holonomic constraints due to no-collision condition) and kinematics (e.g., nonholonomic constraints due to sliding motion specifications); whereas

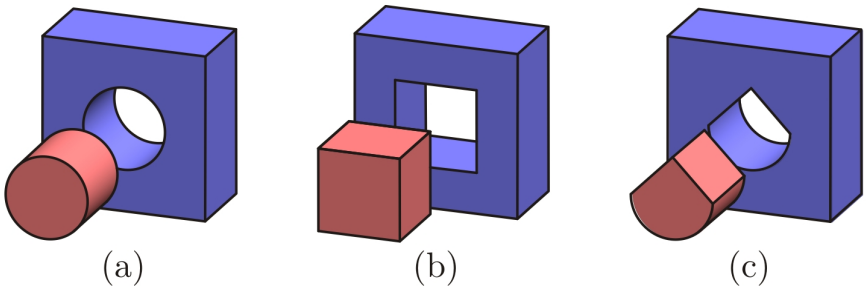


FIGURE 2.1.3: Examples of peg-in-hole assembly problems.

- The second approach uses 'geometric constraints' or 'kinematic constraints' introduced *artificially* to replace the collision response and simplify the solution, ranging from manually specified 'virtual fixtures' to heuristically identified 'mating constraints' (e.g., co-planarity, co-axiallity, vertex/edge/face coincidence, distance or angle offsets, etc.).

I realize that the use of the adjectives 'physical/geometric/kinematic' for the classes of constraints still bears the possibility of confusion. The so-called physical constraints are directly imposed due to the interplay between geometry and kinematics of different objects, and the artificial constraints are also solved by appealing to physics-based dynamic simulation.[2] Therefore, the keywords "organic" versus "artificial" would perhaps constitute more meaningful adjectives for the different types of constraints classified according to this scheme. Nevertheless, I use the former terminology (with some extra care) for the sake of consistency with the conventions in the literature.

Example 2.1.1. Consider the simple peg-in-hole assembly examples shown in Fig. 2.1.3. One could use a variety of geometric representations including exact (e.g., parametric B-rep) or approximate (polygonal mesh or voxelization) representations

[2]It is possible to devise purely kinetostatic constraint solvers by ignoring the dynamic effects—i.e., solving 1st-order, rather than 2nd-order differential equations. However, realistic simulation requires taking dynamic effects into account.

to solve for the no-collision constraints. In this case, these 'physical constraints' arise organically from the part geometries and are implicitly accounted for by ensuring—through the application of contact forces and torques—that the intersection volume of the two parts remains zero at all times. However, whenever the contact geometry is simple—e.g., cylindrical as in Fig. 2.1.3 (a) or prismatic as in Figs. 2.1.3 (b, c)—it is possible to simplify the problem by restricting the motion DOF (e.g., from the original 6 DOF to 1, 2, or 3 DOF) by artificially introducing 'geometric constraints'. In this case, the complex problem of intersection test between arbitrary shapes is reduced to that of simpler geometric abstractions (i.e., virtual fixtures) such as incidence relations between axis lines and corner points. Such incidence relations can be enforced by virtual (axial and torsional) spring-damper couplings whose equilibrium states correspond to proper alignment of virtual fixtures.

A more careful examination of the type of assembly problems similar to the above example yields a more rigorous classification of the assembly simulation methods based on the type of 'kinematic pairs' [336]:

- Lower kinematic pairs correspond to surface contact between parts and are classified completely into the well-known six classes [306]—namely, revolute, prismatic, helical, cylindrical, spherical, and planar. Each class corresponds to an automorphism (i.e., symmetry subgroup) of the 6D configuration space of rigid motions SE(3) (defined in Section 3.2.2).

- Higher kinematic pairs correspond to curve or point contact between parts, for which no such classification exists. For pair of objects of arbitrary shapes, the type of contact belongs to this class in general.

For lower kinematic pairs, the aforementioned classification enables one to encode all possible interactions between a given pair of parts into well-defined types of 'mating constraints' as is customary in many CAD systems. Examples are co-planarity, co-axiallity, co-centricity, etc. These constraints can be simplified into incidence relations between lower-dimensional geometric constructs that abstract the type of contact. For instance, the cylindrical pair in Fig. 2.1.3 (a) can be abstracted by the incidence of the axis of the peg (or at least two points on it) with the axis of the hole, which constrains the motion to a 2D subgroup of SE(3); namely, a translation along and a rotation about the same axis. The enforcement of this constraint in a VE can be realized by a spring-damper coupling between the two cylindrical axes (or certain points on them) that resists an increase in the angle between the two, but is indifferent to the relative rotation around the axes. Similarly, the prismatic pairs in Fig. 2.1.3 (b, c) can be captured by additional conditions to lock the rotation around the axes and restrict the motion further into a 1D subgroup of SE(3). See also Fig. 2.3.1 in Section 2.3.1 for a similar method [389, 390].

Looking back at the different implementations reviewed in Section 2.1.2, most PBM-based simulations use CD to identify the repulsive effects that resist the penetration of individual parts, making direct use of geometry (i.e., exact or approximate representations). On the other hand, most CBM-based simulations disregard the explicit geometric information and use the additional mating constraint semantics imported from the CAD models to implement the spring-damper couplings that contribute both attractive and repulsive effects. An exception to this theme is the Snap-to-Fit system by Olsson et al. [311] which makes direct use of explicit geometric information to create attractive and repulsive effects between the parts that are close to each other without penetration. More specifically, each point on the moving part's

surface is coupled to the nearest neighbor on the stationary part's surface using a virtual spring-damper to create the snapping effect. Although it applies to arbitrary geometry and does not depend on simplifying assumptions on the contact features and kinematic pairs, such a simplistic 'magnetic' energy model is often counterintuitive and countereffective with regard to the assembly intent. In particular, the underlying energy field merely attempts to bring the parts to proximity by pairing the closest points on their respective surfaces and is indifferent to the geometric constraints like the ones implied implicitly in the peg-in-hole examples of Fig. 2.1.3. In contrast, I offer an energy model in Chapter 3 that exhibits stronger relationships with both physical (i.e., collision resistant) and geometric (i.e., mating induced) constraints and blends the two in a single formulation.

2.2 Physically-Based Modeling

In order to realistically simulate the dynamic interactions between parts and sub-assemblies (including the user's interface object/avatar), the majority of haptic-enabled assembly systems perform an explicit real-time integration of the 2nd-order differential equations of motion. These equations are either formulated as

- Newton+Euler's equations, e.g., as in [14, 25, 352, 413, 414], or
- Lagrange's equations, e.g., as in [7, 129, 268, 329, 375, 387, 388],

both of which are equivalent in terms of the underlying mathematics, but offer different computational procedures.

The dynamic simulation typically runs at lower rates (e.g., 100 Hz) compared to the haptic rendering loop at the device level, and the two are interfaced using a

'virtual coupling' [2, 3, 94, 143, 168, 179], which is essentially a spring-damper model that connects two virtual instances of an object, one residing in the physics simulator and the other assumed to be attached to the user's interaction point. I shall elaborate on the concept of virtual coupling in Section 4.2.2.

Among the earliest attempts for using PBM in haptic assembly were earlier implementations of VEDA [164, 165], VADE [212, 215, 217], and HIDRA [103, 104, 279] systems. Examples of more recent PBM-based systems are SHARP [361–365] and HAM(M)S [152,154,254–258]. These systems (and a number of others) were reviewed in more detail in Section 2.1.2. The most challenging set of computations in PBM are due to solving physical constraints arising from contact between different objects in the scene, including rigid and flexible parts and subassemblies (typically imported from complex CAD models).

2.2.1 Physical Constraints

There are two common approaches for computing the contact forces and torques enforcing the physical constraints in real-time:

1. The first method, referred to as the 'penalty method', uses simple force and torque models that make *explicit* use of collision response—e.g., a linear spring-damper model for computing the normal contact forces proportional to a measure of penetration between objects (or their offset shells) [130, 170, 171] and a proper friction model using the normal pressures and the relative sliding/rolling kinematics to compute the tangential forces [13, 176, 291].

2. The second method, referred to as 'constraint-based'—another unfortunate ter-

minology that contributes to confusion with CBM concepts[3]—instead takes an *implicit* account of the unilateral contact constraints and solves the more complex set of constrained equations of motion, using non-smooth Lagrangian mechanics [129, 329, 387, 388].

The penalty method is easy to implement and fast to integrate—given an efficient collision response and impact/friction modeling algorithm—due to the simple form of unconstrained Newton+Euler's or Lagrange's equations of motion. However, the robustness of the penalty method is heavily dependent on small integration time-steps to ensure minimal violations of constraints and rapid response to correct them. This is difficult to achieve with accurate elasticity models for impact mechanics, inferring normal forces from penetration depth as well as friction models for rolling/sliding mechanics, inferring tangential forces from relative kinematics.

The constraint-based method, on the other hand, is more difficult to implement and takes more computing time due to the solution of complicated differential equations, especially as the number of contact features increases. However, it produces more accurate and reliable results, avoids the overhead due to predicting the penetration depth for collision response, and provides straightforward means to model tangential friction forces [268,375]. Both methods are dependent on collision detection (CD), although they might use different CD information such as minimum distance, intersection volume, interpenetration depth, contact normal vector, etc.

[3]This is a possible ground for confusion (due to the bad terminology of PBM vs. CBM) as the naming similarity suggests that the PBM class is confined to explicit penalty methods only while the implicit constraint-based methods should be classified under CBM. However, this in not the case since CBM deals completely with a different family of constraints added artificially on top of the physical constraints, as pointed out in Section 2.1.3.

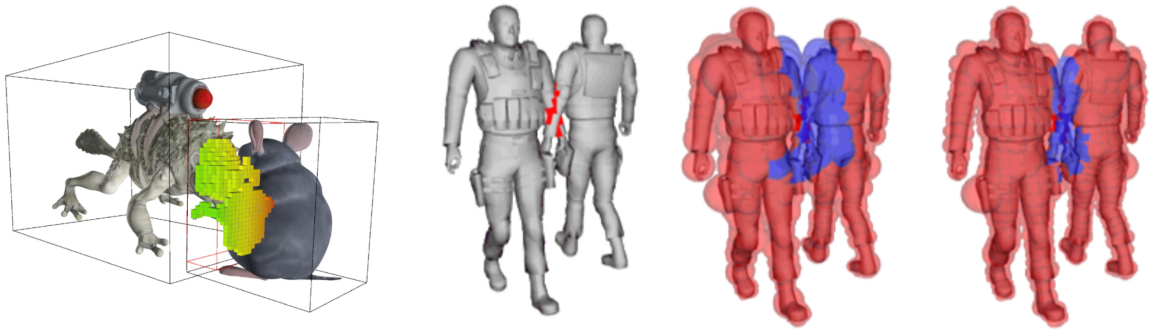
2.2.2 Collision Detection

There are several surveys of CD methods (in a general context) for rigid bodies [222, 238, 259] and flexible elements [391].[4] Here we restrict ourselves to a brief review of the most popular methods for real-time computations.

The classical polyhedral CD methods were used in the earliest systems for haptic assembly. Examples are Voronoi-clipping/marching methods—e.g., V-Clip [290], SWIFT [119], and SWIFT++ [120] used in HIDRA [103, 104, 279]. For realistic applications with geometric complexities that require mesh approximations with large polygon counts (i.e., in the order of millions of triangles), these methods are not fast enough to support the 1 kHz haptic rendering.

On the other hand, the bounding volume hierarchy (BVH) methods have been among the most popular CD methods for graphic and haptic rendering purposes. These methods operate by approximating virtual objects recursively with hierarchies of simple bounding shapes (offering fast collision predicates) stored in a tree data structure. This allows for quickly ruling out early miss configurations in the broad-phase CD as well as a trade-off mechanism between accuracy and running time; namely, by proceeding deep enough down the tree to consume as much of the ~ 1 millisecond per haptic frame as made available for CD budget. Examples are axis-aligned bounding box (AABB) tree-based methods—e.g., a simple algorithm in [294] used in an early haptic training platform [296]—and oriented bounding box (OBB) tree-based methods—e.g, I-COLLIDE [93] and V-COLLIDE [197] giving birth to the

[4]For a good collection of collision detection and proximity query packages, visit the website of the GAMMA research group (Lin et al. and Manocha et al.) at the University of North Carolina at Chapel Hill: gamma.cs.unc.edu/research/collision.
Here is a list of references in chronological order: I-COLLIDE [93], V-COLLIDE [197], RAPID [155], V-Clip [290], IMMPACT [428], H-COLLIDE [159, 160], PQP [248], SWIFT [119], SWIFT++ [120], PIVOT [184, 185], DEEP [235], CULLIDE [157], DEFORMCD [156], DVD [378], and SELF-CCD [386].



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

FIGURE 2.2.1: Combinatorial CD methods use auxiliary approximate representations ranging from voxel maps (a) to bounding spheres (b–d). Figure courtesy of Nießner et al. [305] and Kavan and Zara [227].

H-COLLIDE [159, 160] haptic module, and RAPID [155] used in MIVAS [409, 447]. Among other BVH methods are discrete oriented polytope (DOP) tree-based methods [237, 440] which are generalizations of AABB trees with cubic bounding boxes to convex polytope bounding volumes. A more successful approach is that of hierarchical bounding sphere (HBS) tree-based methods [55, 113, 194–196, 312, 313, 324], which use bounding spheres (as the name indicates) instead of boxes or other polyhedra, including variants such as the bounded deformation (BD) trees [210]. The spherical symmetry of the primitives in all levels of the HBS trees offers simple and fast radial-based collision predicates that are invariant under rotations, making them popular in more recent haptic implementations [69, 346].

For a long time, uniform volumetric enumeration methods such as the one used in Boeing Corp.'s Voxmap PointShell (VPS)^{TM} library [286–288, 410] became very popular for VR applications [223, 231, 232, 330]. VPS^{TM} works by testing the moving objects represented by a shell of vertices and normals (i.e., the 'pointshell') against the stationary obstacles represented by a map of voxels (i.e., the 'voxmap'), and was used in the earlier versions of SHARP [362, 363]. Several improvements were proposed to the

VPS^{TM} method, ranging from model enhancements—e.g., by using signed distance fields to enhance continuous force and torque response [15, 16]—to implementation speed-ups—e.g., by using improved data structures [347, 348]. Although still being popular due to its simplicity and efficiency, the approximate nature of discrete volumetric representations makes them ineffective for low-clearance assembly [362–365]. To overcome this, later versions of SHARP [364,365] employed the Collision Detection Manager (CDM) module of Siemens' D-Cubed, which makes direct use of exact B-rep information extracted from the CAD models. Of course, this comes at the expense of slowing the CD process down and making it impractical for large and complex models (e.g., with numerous NURBS patches).

Coutee and Bras [103] compared multiple polygon-based CD toolboxes—namely, V-Clip [290], SWIFT [119], and SWIFT++ [120]—with VPS^{TM} [286–288,410] in terms of their features and capabilities to provide closest point, collision features, penetration depth, geometric constructions, multibody detection and their effectiveness for haptic simulation. They argued that the lone advantage of V-Clip over the other algorithms is that it provides (a not-so-accurate measure of) penetration distance, which can be overcome by using simple tricks via SWIFT(++). On the other hand, VPS^{TM} and SWIFT++ have the attractive feature of handling arbitrary nonconvex objects, while V-Clip and SWIFT can only handle nonconvex objects as collections of convex pieces. Kim and Vance [231, 232] conducted a more inclusive study comparing a larger number of CD packages—namely, I-COLLIDE [93], V-COLLIDE [197], RAPID [155], PQP [248], and SOLID [27] in addition to the aforementioned four—in terms of their query types and response times. The study concluded that VPS^{TM} is a better choice due to its ease of CAD preprocessing, faster CD query response, and ability to model physical interactions between parts for force feedback—hence its

broad popularity ever since for developing haptic rendering systems.

In contrast to the aforementioned *discrete* methods that evaluate the collision certificate at intermittent integration time-steps, others have tried *continuous* methods based on OBB trees [325, 327, 328, 442–444], which attempt to interpolate the first instance of contact in between two subsequent time points along the dynamic time-stepping. This enables speeding up the integration of constrained motion by decreasing the number of unilateral constraints using a fast 'clash detection' algorithm based on relative motion during each time-step [316]. Although this method was used in an operational context in the industry, it was soon abandoned due to a lack of commercial support [316].

A promising method was recently developed based on hierarchical inner sphere tree (IST) packing [422, 423, 426] and successfully applied to PBM for haptic rendering [424, 425]. In contrast to the bounding sphere methods described above, the IST packing algorithms use the hierarchy of spheres to pack the *interior* of the virtual objects. Nevertheless, both methods share the computational advantage due to the spherical symmetry in primitive collision predicates. The sphere-packing approximation of the interior can be viewed as a nonuniform extension to the uniform volumetric enumeration approach used in VPS$^{\text{TM}}$, since it starts from a grid-based discretization of the shape (similar to the voxmap in VPS$^{\text{TM}}$) over which the distance function is computed and the sphere centers are populated using a simple greedy algorithm. It was shown to outperform the VPS$^{\text{TM}}$ for nonconvex moving objects, but its effectiveness to handle thin objects is yet to be tested [316].

At a very abstract level, one could classify CD methods to combinatorial techniques (including most, if not all of the aforementioned methods), and analytic methods [270]. PBM applications require not only a collision/non-collision certificate, but

a gradient of the constraint function to compute the contact forces and torques. A disadvantage of the combinatorial methods is their indirect approach to infer a gradient-like quantity from the certificate point, which is not trivial for general surface contact [270]. The analytic methods [98,149,293], on the other hand, provide a more uniform and robust alternative, which has been popular for a long time in robotics [269]. Recently, Lysenko [270] developed an efficient analytic method based on earlier works in robotics spatial planning [106, 228] and group morphology [271, 272, 342], which takes advantage of Fourier transforms to compute the collision response and its general configuration space gradient (both with respect to translations and rotations) for narrowphase CD. This method has great potential for haptic applications, though I am not aware of its implementation into any software library or simulation engine at the time of writing this thesis. Its formulation is of particular interest to this project, as it shares the same mathematical foundations and uses the same computational properties of Fourier transforms to speed up the algorithm for use in haptic rendering.

A discussion of real-time CD methods would not be complete without a reference to contact modeling between deformable objects, which are prevalent in industrial assembly (e.g., electric cables, hydraulic hoses, rubber seals, leather furnishings, etc.) [316]. Classical methods range from discrete mechanical elements (DME) [57,58,135] to finite element method (FEM) [181, 207, 318, 448], which are suboptimal for high frame rate haptic rendering. A few recent studies have successfully applied Signorini's contact model to haptic assembly of deformable objects [115–117]. Nevertheless, handling complex deformable shapes and large models remains a challenge [316].

Combinations of aforementioned algorithms have been implemented into popular physics simulation engines (PSE) for graphic and haptic rendering, such as Ageia

PhysX^{TM} SDK [68] and Bullet Physics SDK [348]. Gonzalez-Badillo et al. [151, 153] recently conducted a comparative performance evaluation of these PSEs and their CD capabilities in practice for haptic assembly. In particular, they measured and compared the task completion time (TCT), mean force feedback (MFF), and physics simulation time (PST) indices in several benchmark examples for the static trimesh/HACD module of PhysX^{TM} versus the GIMPACT module of Bullet and concluded that in general the latter outperforms the former for haptic assembly tasks.

2.3 Constraint-Based Modeling

Although implementing physics-based simulation with a combination of CD and impact/friction mechanics seems the most natural choice (at least in theory) for a virtual mimicry of real-world constrained motion, it is not reliable in practice for final insertion of the objects into position [316, 362–365, 402].

At a fundamental level, this happens due to the degeneracy of the collision-free feasible subspace (i.e., the 'free space') in the neighborhood of the final assembly configuration, leading to decreased DOF associated with common mating constraints. Most mating constraints used to model zero-clearance mechanical joints are characterized with multiple compatible unilateral (i.e., inequality) constraints that are critically satisfied during contact leading to one or more bilateral (i.e., equality) constraints, restricting the motion to a surface or a curve.[5] Such a degenerate collection of critically satisfied inequality constraints is extremely unstable with respect to small

[5] An equality constraint $g(\mathbf{p}) \geq 0$ ($\mathbf{p} \in \mathbb{R}^3$) is 'critically satisfied' if $g(\mathbf{p}) = 0$. Two constraints $g_1(\mathbf{p}) \geq 0$ and $g_2(\mathbf{p}) \geq 0$ are compatible if they define a nonempty subset of the 3−space. If that subset forms a lower dimensional subspace over the intersected boundaries, then they collectively define a bilateral equality constraint $g_1(\mathbf{p}) = g_2(\mathbf{p}) = 0$.

perturbations, rendering lower kinematic pairs and their constraint resolution practically 'incomputable'. This is because a small error in constraint specification may lead to major topological changes to the collision-free configuration subspace (i.e., the 'free space') that either completely eliminate the constraint or make it theoretically unresolvable unless tolerances are explicitly incorporated.[6]

At a practical level, it is difficult to stabilize the motion along the degenerate subspace for at least two reasons:

1. numerical geometric errors due to the approximate representations used in fast CD methods popular for haptic rendering (e.g., voxelized interior or triangulated boundary); and

2. input noise due to authentic hand vibrations and device encoder reading errors.

The latter is particularly important, implying that even with the improved accuracy of the algorithms that use fine mesh approximations or exact B-rep data, using CD alone for low-clearance insertion is still impractical [364, 365, 402]. One can alleviate the problem by using smaller integration time-steps to keep the unnecessary minor collision events (i.e., violations of the equality constraints) at a minimum, to achieve more stable dynamic response and haptic feedback. However, the finite time-step in real-time applications is lowerbounded by the computation time per frame, which is dictated by hardware capacity.

[6]Take, for example, two inequality constraints $g_1(\mathbf{p}) \geq 0$ and $g_2(\mathbf{p}) \geq 0$ corresponding to non-penetration condition for two planar surfaces between which a planar part is sandwiched without a clearance. The intersection $g(\mathbf{p}) := g_1(\mathbf{p})g_2(\mathbf{p}) \geq 0$—assuming nonnegative analytic functions $g_1, g_2, g : \mathbb{R}^3 \to [0, +\infty)$—is a 3D subspace of the 6D configurations space defining a lower kinematic pair with two translational and one rotational DOF. Upon introducing a small perturbation $\epsilon \geq 0$ (e.g., $g'_{1,2}(\mathbf{p}) := g_{1,2}(\mathbf{p}) \pm \epsilon$) that 3D subspace may widen up to a 6D region (e.g., in 'clearance fits') or completely disappear (e.g., in 'force fits' or 'shrink fits').

An alternative solution is to *artificially* introduce a set of bilateral constraints, rather than relying solely on the group of unilateral constraints organically resulted from CD. Such an approach is mathematically equivalent to a local reparameterization of the configuration space to embed the feasible subspace and to ensure the fulfillment of the constraints explicitly rather than attempting to solve for the original constraints specified implicitly. These artificial constraints are typically classified into geometric (i.e., holonomic) and kinematic (i.e., nonholonomic) constraints.

2.3.1 Geometric Constraints

In most commercially available design and assembly environments such as the modern CAD software (e.g., CATIA®, Pro/ENGINEER®, NX, etc.) the so-called 'mating constraints' are classified into simple spatial relationships between the contact features, such as co-planarity in prismatic features, co-axiallity in cylindrical features, distance and angle offsets, etc. To solve the final insertion problem for haptic assembly, one practical approach is to *manually* specify the mating constraints in close proximity of the final assembly configuration. The geometric constraints can be either extracted and imported from the CAD model—i.e., manually specified by the CAD user (e.g., the designer)—or specified on-the-fly within the VE—i.e., manually specified by the VE operator (e.g., the inspector)—using a variety of constraint management systems developed for VR-CAD applications [275, 299, 300]. For example, VADE [212,215,217] and MIVAS [409,447] directly imported pre-defined constraint information from Pro/ENGINEER® CAD models. The later versions of SHARP [364,365] used the Dimensional Constraint Manager (DCM) module of Siemens' D-Cubed for defining and solving geometric constraints within the VE itself. Rather than using

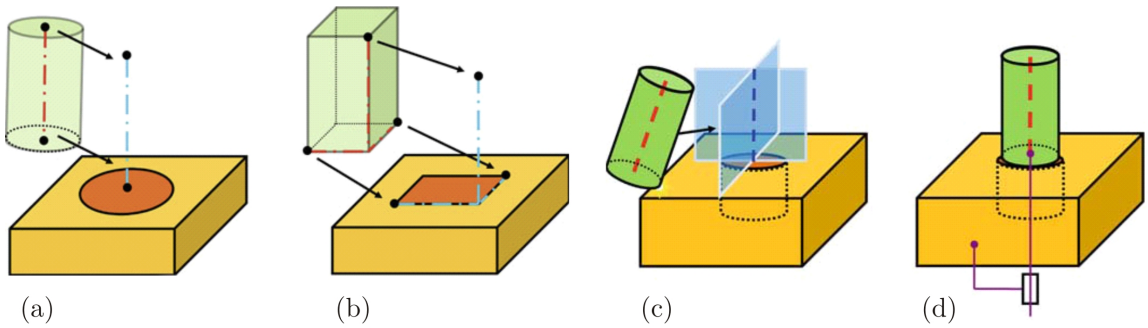


FIGURE 2.3.1: Using virtual fixtures for abstracting the mating constraints for cylindrical and prismatic pairs (a, b). At the close vicinity of the insertion site, the virtual fixtures (e.g., guiding planes in (c)) are used to align the peg along the hole. Once the proper alignment is reached, the mating pair's DOF is limited to model a suitable mechanical joint (e.g., cylindrical pair in (d)). Figure courtesy of Tching et al. [389, 390].

the assembly semantics of the original CAD models, the virtual constraint guidance (VCG) method presented by Tching et al. [389, 390] relied on user-specified 'virtual fixtures' [343, 344], which are added abstract and simple geometric elements rigidly attached to the fixed and moving parts—e.g., a pair of perpendicular planes intersecting at the axis of a cylindrical hole, to constrain and guide two points selected along the axis of a cylindrical peg. Figure 2.3.1 illustrates the use of virtual fixtures for peg-in-hole examples with cylindrical and prismatic mating pairs.

A few recent studies attempted to *automatically* identify the assembly intent and associated geometric constraints by analyzing semantic information of individual part geometries [52, 199, 200, 275], referred to by Vance and Dumont [402] as the automatic geometric constraints (AGC) method. This method relies on matching 'functional surfaces' [200]—e.g, a cylindrical surface characterized by its axis and diameter, which could be used to predict the intended mating relation and associated trajectories when a peg is brought to the proximity of a hole. However, these methods are limited to matching simple (e.g., planar, cylindrical, spherical, and conical) geometric features.

The effectiveness of both VCG and AGC methods relies heavily on either manual specification of the type of mating selected from a finite library of simple constraints, or heuristic models for identifying such mating pairs when the corresponding simple geometric primitives are in proximity. A generic solution that automates the identification and pairing for features of arbitrarily complex surface geometry is missing.

2.3.2 Kinematic Constraints

In a similar fashion to holonomic (i.e., geometric) constraints, one could specify non-holonomic (i.e., kinematic) constraints that depend on the relative linear and angular velocities of parts during assembly. An important caveat is related to proper differentiation of the motion of the haptic proxy from discrete encoder readings in the presence of the added noise and device errors. Our experiments have led to the observation that the data filtering provided by commercial haptic device libraries and APIs might not be adequate for 1st and 2nd differentiation, and additional techniques might be required [211].

2.4 Future Directions

Having outlined current practice and existing methodologies in haptic-assisted VP/VA (including PBM and CBM), this section briefly comments on promising directions for future studies and how the approach presented in this thesis contributes to these advancements.

2.4.1 Technology Readiness

Perret et al. [316] presented an assessment of the maturity of the technical solutions in haptic-assisted assembly using the technology readiness level (TRL) index originally developed by NASA in the 1980s for space-flight systems and later adopted and expanded by the US Air Force to encompass other technologies. Based on this measure, they assessed the maturity of interactive rigid-body assembly simulation with haptic feedback quoted here as follows:

- In terms of 'path finding': TRL = 8, i.e., "technology has proven to work in its final form and under expected conditions."
- In terms of 'final insertion': TRL = 5, i.e., "the basic technological components are integrated with reasonably realistic supporting elements."
- In terms of 'human positioning': TRL = 6, i.e., "representative model or prototype system, which is well beyond TRL = 5, is tested in a relevant environment."

The 'path finding' problem—more commonly referred to as 'path planning' in the robotics literature [269]—is of assessing the feasibility of assembly by identifying a collision-free path (or lack thereof) from some initial configuration of a given part or subassembly to the final configuration. If no such path exists, the goal is to identify the bottlenecks and critical collision points to modify the product or process in order to resolve the design problem. The most popular automatic approach to this problem is by using probabilistic roadmap (PRM) planners [32, 147, 229, 230] which randomly sample the configuration space of rigid motions SE(3) (introduced in Section 3.2.2) and connect the nearby points to form a graph over which a feasible path can be traced. Although automatic path planning performed in an offline preprocessing step

can be used later for haptic guidance as in [90–92,172–174,245,246], the true benefit of haptic-assisted VP/VA is realized when it is used as an alternative to automatic path planning by calling upon the human's cognitive abilities and understanding of spatial relationships—e.g., to feel bottlenecks, evaluate clearances, and explore possible improvements—rather than exploiting the computational power of the machine [316]. Hence it appears that basic CD-based PBM remains to be the most natural toolbox for this purpose. The TRL of 8 indicates an adequate level of maturity in this aspect, even though consistent computational improvements are needed to enable fast CD for increasing size and complexity of models. It has been shown that analytic methods similar to the one presented in this thesis are a promising future direction for both path planning [106, 228] and CD [20, 270].

The 'final insertion' problem appears as the least mature component with a TRL of 5, facing multiple challenges to reach effective industrial implementation. It is impossible to rely on CD alone for the final insertion of a part when low-clearance fits are involved [316]. One promising future direction to solve the final insertion problem is to use a hybrid PBM+CBM approach (detailed in Section 2.4.2) where the traditional simulation of contact-constrained motion is confined to a 'free motion' phase while fixed mating constraints are artificially added to the PSE solver for a 'fine insertion' phase. Once again, these constraints are extracted from CAD model as in [212,215,217,409,447], defined by the user within the VE as in [364,365,389,390], or determined (i.e., guessed) automatically by the system from an inspection of the geometry as in [52, 199, 200, 275]. One challenge faced by the hybrid approach is developing an algorithm to account for the switch between the two phases, using a variety of remedies such as blending algorithms as in [319, 364, 365] or guiding mechanisms as in [389,390]. The second challenge is that switching off CD altogether

during final insertion might lead to missing contact or collision outside the insertion site [316], to which no plausible solution has been published so far to the best of my knowledge. This thesis proposes an alternative approach that unifies the two phases (detailed in Section 2.4.3) without having to deal with the challenges presented by such an artificial duality.

The 'human positioning' problem is largely beyond the scope of this thesis. One of the greatest challenges in this area is of the computational intensity of introducing an avatar into the simulation environment whose realistic model involves hundreds of new rigid bodies to deal with. See [316] for more details.

2.4.2 A Hybrid Approach

Although it has been shown that reducing the DOF of motion using geometric constraints supports highly accurate manipulation and positioning in VEs [54], the ad hoc nature of the constraint detection algorithms does not provide sufficient generality to completely replace CD to constrain the motion. Consequently, the state-of-the-art in haptic assembly is a 'two-phase' approach [402], i.e., to divide the process into a 'free motion' phase accomplished with the help of CD engines, and a 'fine insertion' phase using pre-specified or computer-predicted constraints. Vance and Dumont [402] observed such a common theme among a few recent studies and systems:

1. The automatic geometric constraints (AGC) method by Seth et al. [364, 365] relied on CD between B-rep surfaces during a free motion phase when the parts move freely (i.e., with 6 DOF except for the collision-induced constraints) in the VE. Contact between B-rep elements signaled a switch to the insertion phase of the assembly simulation when the geometric constraints were used to guide

the assembly process. These geometric constraints were easily defined based on the B-rep semantics and automatically identified using simple heuristics—e.g., alignment between the cylindrical axes. Once a constraint was identified, the two parts were aligned and the number of DOF allowed for the user motion was reduced to impose the geometric constraint [402].

2. The virtual constraint guidance (VCG) method by Tching et al. [389, 390] used non-smooth dynamic simulation during an exploration phase when the parts move freely (i.e., with 6 DOF except for the collision-induced constraints) in the VE. The method relied on virtual fixtures [343, 344] to guide the moving object to a specific configuration without making any changes to the underlying CAD geometry. Once the proper insertion alignment was reached, the CD was disabled and the assembly phase was started by modeling the final insertion as a DOF-limited relative motion of simple mechanical joints (e.g., prismatic, ball, hinge, etc.). The transition between the two phases was triggered by the collision of the virtual guides between the moving and stationary objects [402].

3. The dynamic decomposition and integration of DOF (DIOD) method by Veit et al. [404] divided the assembly task into a ballistic phase and a control phase, and took a different approach based on detecting velocity changes to trigger the swith between the two phases. During the ballistic phase the user could freely move and manipulate an object at a 1:1 ratio between the tracked (i.e., device) and virtual (i.e., object) velocities. During the fine positioning phase, instead of scaling the resultant velocity to increase the dexterity, the total velocity was decomposed along 3 orthogonal directions and the scaling wass applied only to the component of the velocity that is below a given threshold [402].

There are two major difficulties faced in this approach. First, it requires developing mechanisms to detect the insertion intent (e.g., using pose or speed clues) and to model the transition between the two phases. The aforementioned implementations typically rely on CD between surface elements associated with insertion constraints [364,365], CD between the user-defined virtual fixtures [389,390], or velocity changes that hint on the user's intent to perform an insertion task [404]. Once the alignment has been reached, part CD is switched off and the number of DOF is reduced to assist the user with final insertion. Second, switching off part CD altogether is not satisfactory as a contact with geometry outside of the insertion area could oppose the movement [316]. To the best of my knowledge, the latter problem is also open.

2.4.3 A Unified Approach

Both physical constraints (i.e., captured by CD and used in PBM) and mating (geometric and/or kinematic) constraints (i.e., interpreted from CAD and used in CBM), which govern the part behavior in the two phases of the hybrid approach described in Section 2.4.2, are solutions to the same basic problem that lie on the two extreme ends of an spectrum in conceptual and computational terms:

- The former apply to arbitrary geometry (i.e., general solids with semianalytic surfaces) while the latter intrinsically work only for simple shapes (e.g., assembly features with 1st- or 2nd-order semialgebraic surfaces).

- The former are numerically complex and time-consuming even using approximate representations (e.g., voxmap enumeration, polyhedral meshing, or BVH tree-based sampling) while the latter are as simple as computing basic and fast geometric predicates.

- The former provides a sense of crisp contact (e.g., impact and friction) at the expense of instability to small perturbations while the latter creates a (sometimes unrealistically) more flexible sensation with the additional benefit of robustness to errors and noise.

Is it possible to come up with a single part behavior model that subsumes both of the above and intrinsically enables a smooth transition between them rather than salvaging one to blend the discrete phases? Chapter 3 describes a 'geometric energy' formulation that underpins such a unified approach.

Chapter 3

Geometric Energies

3.1 The Proposed Approach

In this section I outline the proposed approach to the analytic formulation of geometric interactions for interactive assembly in a virtual environment (VE).

Given a set of mechanical components of a prospective assembly in a graphics- and haptics-enabled VE, the goal is to develop a computational model to perform the following set of tasks:

1. obtaining proper 'shape descriptors' that capture the geometric and topological characteristics of the different components which are relevant to assembly and can be thought of as generic replacements for ad hoc virtual fixtures [343, 344] (Section 3.4);

2. formulating a quantitative 'score function' to measure the goodness of the 'geometric fit' between the shapes—i.e., collision-free extensive contact exhibiting

shape complementarity—for arbitrary spatial configurations, based on overlapping the previously extracted shape descriptors (Section 3.5);

3. obtaining an artificial energy-field from the score model, whose gradient can be used as the guidance and constraint forces and torques during object manipulation in the VE—hence replacing the existing penalty methods based on linear spring-damper models (Section 3.7).

The goal is to develop a potent framework that performs these tasks without making any simplifying assumption on the shape, the intended function (e.g., type of fit in the assembly [1]), or the proper spatial relationships of the parts.

The first step entails the most challenge from a theoretical point of view, since obtaining a quantitative description of the assembly features requires an understanding of the qualitative notion of a 'proper fit', and is not trivial for arbitrary geometry. The shape descriptors can be obtained in a preprocessing step for each rigid part. Therefore, the predominant computational challenges are pertaining to the real-time computations in the next two steps, particularly due to the 1 kHz haptic rendering rate requirement. This makes the choice of the overlapping mechanism that combines the individual objects' shape descriptors into their pairwise score/energy function extremely important. I show in Sections 3.3, 3.5, and 3.7 that convolution algebra provides an effective and efficient setting for this combination.

3.1.1 Formulation Steps

In particular, given a pair of parts present in a scene, I aim to formulate an energy field over some 'configuration space' (often abbreviated as the $\mathsf{C}-$space) parameterizing the relative motion of the constituent parts, that

1. penalizes collisions between the interiors of the pairs of objects, which is equivalent to an implicit representation of the so-called 'configuration space obstacles' (or $\mathsf{C}-$obstacles for short) [269] (i.e., 'ridges' of the energy terrain); and

2. rewards proper fit/contact, i.e., configurations that exhibit superior shape complementarity between assembly features, which is equivalent to a re-scoring of the feasible collision-free $\mathsf{C}-$space (i.e., 'valleys' of the energy terrain).

A recently proposed analytic approach to collision detection (CD) [270] holds great promise for computing the narrowphase collision response in real-time for haptic assembly. The method defines a continuous and nonnegative 'gap function' [169] between the two objects as a convolution of some implicit representation of the shapes—e.g., the 'indicator function' (defined in (3.2.4) ahead) [228], distance-based depth functions [337], or smooth 'bump functions' [270]. The gap function implicitly defines the $\mathsf{C}-$space obstacles—i.e., the set of infeasible positions and orientations that correspond to interior collisions—and is fundamentally related to Minkowski operations [303, 304] and group morphology [271, 272].

Unfortunately, even *exact* CD is not sufficient for effective low-clearance insertion, since it computes large penalties for minor collision events that can occur due to hand vibration or device encoder inaccuracies [365, 366],[1] as depicted in Section 2.2.2. Furthermore, CD alone does not accomplish the second objective stated above. It assigns zero penalty to feasible configurations corresponding to zero intersection volume regardless of the nature of contact, hence does not discriminate between unassembled (i.e., no-contact) and assembled (i.e., proper surface, curve, or point contact), as will

[1] This can get more problematic if one attempts to use truncated Fourier expansions leading to small sinusoidal artifacts, without which the ultimate efficiency advantage of the convolution paradigm would be lost (Section 4.1.5).

be detailed in Section 3.3. In contrast, by making use of a special class of shape descriptors (i.e., the SDF [19]) that better capture the geometric information of the assembly features, both problems are overcome in a unified framework with the additional benefit of providing an adjustably 'relaxed' collision penalty.

3.1.2 Algorithmic Steps

As depicted in Section 1.2.2, computational efficiency is critical for haptic assembly. The main underlying idea is that the numerically intensive convolution and differentiation operations in the physical domain convert to simple algebraic multiplication operations in the frequency domain, as detailed in Section 3.6. To exploit this property, we proceed as follows:

1. precompute the SDF descriptions of the individual parts offline;

2. transform the SDFs to their frequency domain representations using forward FFTs on the GPU—without worrying about the configuration of the parts;

3. compute the frequency representation of the geometric energy field as a product of the SDF frequencies (up to the desired truncation), either for sampled translations and/or orientations or for a particular singleton configuration; and

4. retrieve the geometric energy field in the physical domain using inverse FFTs on the GPU for sampled translations and/or orientations, or using direct integration for a singleton configuration.

The important point is that the first two steps can be precomputed offline, hence the mathematical and computational complexity of SDF is not an issue. To efficiently

implement the next two steps in real-time, one has the freedom to use a low-pass filter (Section 4.1.5), i.e., to retain only a subset of the dominant modes of the SDF descriptors and geometric energy function. Furthermore, this method allows an exact computation of the energy gradient (i.e., unified collision and guidance forces and torques) up to a desired accuracy from the truncated Fourier expansions.

3.2 Definitions and Preliminaries

In this section I present the basic definitions and preliminaries, particularly, of the class of computable geometric shapes and configurations. We present what I mean by analytic (in contrast to combinatorial) methods and how one can describe and compute shapes and configurations within this paradigm.

3.2.1 Geometric Modeling

Following the good tradition of separating mathematical models [332] from computational representations [334], let us restrict our attention to the class of 'well-behaved' solid objects (i.e., 'r-sets') $\mathsf{S} \subset \mathscr{P}(\mathbb{R}^3)$, defined as compact (i.e., bounded and closed), regular semianalytic subsets of the Euclidean metric space $\mathbb{R}^3$ endowed with the usual topology based on the L^2-metric [332–334].[2] Hereafter, I use the common terms 'solid' and 'r-set' interchangeably to refer to a member of this class.

The regularity condition $(S = \text{r}S)$,[3] on the one hand, ensures that for every

[2]The collection $\mathscr{P}(A) = \{B \mid B \subset A\}$—also sometimes denoted as 2^A for good reasons [189]—is the 'power set' of a set A, i.e., the set of all subsets of A. The collection of all r-sets is thus a proper subcollection of the power set of the Euclidean 3−space.

[3]The regularization $\text{r}S$ of a set $S \subset \mathbb{R}^3$ is the closure of its interior $\text{r}S = \text{k}(\text{i}S)$, i.e., the smallest closed set that contains the largest open set contained in S. A regular set is one that equals its regularization $S = \text{r}S$ [331].

set $S \in \mathsf{S}$, the set's 'interior' $\dot{\iota}S$, 'exterior' $\dot{c}S$, and 'boundary' ∂S are well-defined notions [331], and prevents undesirable artifacts such as 'dangling' edges or isolated points that do not correspond to physically realizable shapes [332].

The semianalytic requirement, on the other hand, guarantees 'triangulability' (hence *finite describability*) of the set [332] and prevents undesirable pathological behavior at the boundary [332] and the skeleton [73]. Both conditions are sufficiently specific to enable theoretical developments as well as algorithmic tractability, yet general enough to encompass all practically significant shapes for most solid modeling applications, including virtual assembly and disassembly tasks.

I choose to make an additional assumption that the boundary ∂S is an oriented piecewise C^1-manifold, i.e., can be decomposed into a finite number of differentiable surface patches that are sewed together along at most a finite number of sharp edges and corners (if any at all). This enables formulating flux integrals over the boundary (Section 3.4.3) as a finite summation of surface integrals over those patches, each specified with well-defined and consistent outward normal vectors throughout their manifold interiors.[4]

It is worthwhile noting that our formulation does not impose, in principle, any restriction on the representation scheme, as long as it satisfies the informational completeness requirement [334]—particularly, it suffices to support Euclidean distance queries and point membership classification (PMC) tests [395]. This applies to *exact* representations—e.g., parametric B-reps ranging from simple surfaces to nonuniform rational B-splines (NURBS) extracted from the CAD models—as well as *approximate* representations—e.g., triangular mesh or volumetric enumerations of the exported

[4]It is common to assume the boundary of r-sets to be piecewise smooth (i.e., $C^\infty-$) manifolds that admit infinitely differentiable parameterizations. However, I will not make use of second- and higher-order differential properties (e.g., curvatures) for SDF formulation in Section 3.4.3.

CAD models. It is important to note that, especially when dealing with approximate representations, the employed shape descriptors must be stable and robust with respect to small perturbations in the boundary; otherwise they cannot be used effectively for designing computational algorithms [8, 118].

3.2.2 Configuration Space

The concept of a 'configuration space' (commonly abbreviated as the $\mathsf{C}-$space) was introduced to the field of robotics by Lozano-Perez [269] to parameterize the relative motions of rigid bodies (e.g., between a robot and its workspace) as points $T \in \mathsf{C}$ that reside on a higher-dimensional embedded topological manifold.

For a pair of arbitrary solids $S_1, S_2 \in \mathsf{S}$ each solid representing one part or subassembly as a single object, the rigid motion of both objects at any instant of time can be described by the 'absolute configurations' $T_1, T_2 \in \mathrm{SE}(3)$ that transform some fixed absolute coordinate frame to an orthonormal triad attached to each object. The $\mathsf{C}-$space is characterized with the special Euclidean group $\mathrm{SE}(3) \cong \mathrm{SO}(3) \ltimes \mathrm{T}(3)$, i.e., combination of proper orthogonal rotations $\mathrm{SO}(3)$ and translations $\mathrm{T}(3)$, together representing all possible rigid body motions. Each rigid body motion $T \in \mathrm{SE}(3)$ can thus be represented by a 4×4 homogeneous matrix $[T]_{4\times4}$, or alternatively, by a tuple $T := (R, \mathbf{t})$ in which the rotation component $R \in \mathrm{SO}(3)$ is represented by a 3×3 orthogonal matrix $[R]_{3\times3}$ with $\det(R) = +1$ and the translation component $\mathbf{t} \in \mathrm{T}(3) \cong \mathbb{R}^3$ is represented by an arbitrary $3-$vector $[\mathbf{t}]_{3\times1}$. I use the latter representation for the rest of this thesis, with the following properties:

- Two subsequent motions $T_1 = (R_1, \mathbf{t}_1)$ and $T_2 = (R_2, \mathbf{t}_2)$ applied in respective order are equivalent to the motion $T = T_2T_1 = (R_2R_1, \mathbf{t}_2 + R_2\mathbf{t}_1)$.

- The inverse of a motion $T = (R, \mathbf{t})$ is defined by $TT^{-1} = T^{-1}T = \mathsf{id}_{\mathrm{SE}(3)}$—where $\mathsf{id}_{\mathrm{SE}(3)}$ is the identity motion—and is obtained as $T^{-1} = (R^{\mathrm{T}}, -R^{\mathrm{T}}\mathbf{t})$.
- The action of a motion $T = (R, \mathbf{t})$ on the points of the 3−space is defined as $T\mathbf{p} = (R\mathbf{p}) + \mathbf{t}$ for all $\mathbf{p} \in \mathbb{R}^3$—i.e., a noncommutative sequence of a rotation $\mathbf{p} \mapsto R\mathbf{p}$ followed by a translation $\mathbf{p} \mapsto (\mathbf{p} + \mathbf{t})$—hence $T^{-1}\mathbf{p} = R^{\mathrm{T}}(\mathbf{p} - \mathbf{t})$.

Every moved instance of a rigid solid $S \in \mathsf{S}$ is denoted by $S' := TS \in \mathsf{S}$:

$$TS = (R, \mathbf{t})S := \{T\mathbf{p} = (R, \mathbf{t})\mathbf{p} \mid \mathbf{p} \in S\}\,. \tag{3.2.1}$$

In a virtual assembly environment, if S_1 and S_2 represent initial instances of the rigid solid parts (e.g., at rest on the assembly table), any subsequently moved instances of the two parts are given by T_1S_1 and T_2S_2, respectively. The 'relative configuration' of S_2 as observed from a coordinate frame fixed on S_1 is $T = T_1^{-1}T_2 \in \mathrm{SE}(3)$. Every such configuration can be conceptualized as a point in the 6D C−space SE(3).

Configuration space modeling [302] addresses questions of shape and motion and establishes correlations between them—e.g., pertaining to detecting collisions, similarity, complementarity, or symmetry. For example, the 6D geometric constructs that characterize the subsets of SE(3) that correspond to collisions between S_1 and TS_2—or equivalently between T_1S_1 and T_2S_2 in the actual assembly scene—are called 'configuration space obstacles' (or C−obstacles for short), which can be obtained as a homogeneous Minkowski product of the two sets [271, 272]:

$$S_1 \boxtimes S_2 = \{T \in \mathrm{SE}(3) \mid S_1 \cap^* (TS_2) \neq \emptyset\}\,, \tag{3.2.2}$$

where the asterisk indicates regularized Boolean operations [396]. It is often more

convenient for theoretical reasons—especially within analytic modeling presented in Section 3.2.3—to work with the regularized $\mathsf{C}-$obstacle $S_1 \boxtimes^* S_2$, its regularized complement $\complement^*(S_1 \boxtimes^* S_2) := \mathrm{SE}(3) -^* (S_1 \boxtimes^* S_2)$ called the 'free space', and their shared boundary $\partial(S_1 \boxtimes^* S_2)$ called the 'contact space'.

3.2.3 Analytic Methods

Analytic modeling relies on describing shape and configuration pointsets in terms of *functions* and formulating fundamental operations—e.g., pertaining to detecting collisions, similarity, complementarity, or symmetry—in terms of correlations between those functions [20, 21]. For example, Minkowski operations [342] that are central to mathematical morphology are formalized as convolutions of constituent functions and computed efficiently in the Fourier domain [271, 272].

As a common theme, a solid $S \in \mathsf{S}$ can be implicitly described as a sublevel set[5] of a real-valued function $f(\mathbf{p}; S)$ such that, for example, $f(\mathbf{p}; S) \geq 0$ or $f(\mathbf{p}; S) \leq 0$ iff $\mathbf{p} \in S$. In this case, S can be retrived from the functional description $f(\cdot; S)$:

$$S = \left\{ \mathbf{p} \in \mathbb{R}^3 \mid \mathrm{E}\left[f(\mathbf{p}; S) \right] = \mathrm{true} \right\}, \tag{3.2.3}$$

where $\mathrm{E}[f(\mathbf{p}; S)]$ is a logical expression about the outcome of $f(\cdot; S)$ at $\mathbf{p} \in \mathbb{R}^3$, e.g., $f(\mathbf{p}; S) \geq 0$ or $f(\mathbf{p}; S) \leq 0$. In this sense, the functional description $f(\cdot; S)$ is informationally complete, i.e., sufficient to reconstruct the set $S \in \mathsf{S}$. However, one might choose to use incomplete descriptions that capture only a subset of geometric and/or topological properties of the shape needed for a particular purpose. In either

[5] More rigorously, it is sometimes necessary to use regularized sublevel sets, for instance, when convolving functions that define shapes into functions that define their Minkowski combinations [271, 272]. For our purposes, omitting this detail does not make much difference.

case, I call the generic function $f : \mathbb{R}^3 \times \mathsf{S} \to \mathbb{R}$ (or $\mathbb{C} = \mathbb{R} \times (\mathbf{i}\mathbb{R}) \cong \mathbb{R}^2$) a 'shape descriptor' if the following important assumption holds:[6]

Assumption 3.2.1. The function $f : \mathbb{R}^3 \times \mathsf{S} \to \mathbb{R}$ (or $\mathbb{C}$) is invariant under rigid body transformations, i.e., $f(\mathbf{p}; TS) = f(T^{-1}\mathbf{p}; S)$ for all $S \in \mathsf{S}$ and $T \in \mathrm{SE}(3)$.

This simply means that repositioning and/or reorienting the object $S \in \mathsf{S}$—or equivalently, a change in the choice of the orthonormal coordinate system from which the object is observed—induces the same transformation on the field $f(\cdot; S)$ without affecting its profile. Hence obtaining its value at any point $\mathbf{p} \in \mathbb{R}^3$ against the moved instance $S' := TS$ amounts to a query at $\mathbf{p}' := T^{-1}\mathbf{p}$ against S.

Perhaps the simplest and most common analytic shape descriptor is the 'indicator function' $\mathbf{1} : \mathbb{R}^3 \times \mathsf{S} \to \{0, 1\}$ defined as

$$\mathbf{1}(\mathbf{p}; S) = \begin{cases} 1 & \text{if } \mathbf{p} \in S, \quad \text{i.e., } \mathsf{PMC}(\mathbf{p}; S) \leq 0 \\ 0 & \text{if } \mathbf{p} \notin S, \quad \text{i.e., } \mathsf{PMC}(\mathbf{p}; S) > 0. \end{cases} \tag{3.2.4}$$

Another important shape descriptor is the point membership classification (PMC) function $\mathsf{PMC} : \mathbb{R}^3 \times \mathsf{S} \to \{-1, 0, +1\}$ defined as[7]

$$\mathsf{PMC}(\mathbf{p}; S) = \begin{cases} -1 & \text{if } \mathbf{p} \in \imath S, \\ 0 & \text{if } \mathbf{p} \in \partial S, \\ +1 & \text{if } \mathbf{p} \in \complement S. \end{cases} \quad = \mathbf{1}(\mathbf{p}; \complement S) - \mathbf{1}(\mathbf{p}; \imath S). \tag{3.2.5}$$

The PMC function partitions the 3−space into the three disjoint sets, namely, the

[6] I show in Section 3.5 that there are computational benefits to use complex- instead of real-valued shape descriptors for fit score function formulation.

[7] It is sometimes more descriptive to define the range of the PMC function as the set $\mathsf{PMC}(\mathbb{R}^3, S) :=$ {"in", "on", "out"} [333, 395], but I choose numerical outcomes in (3.2.5) to facilitate the definition of a signed distance function in (3.2.6) through (3.2.8).

interior ($\mathsf{PMC}(\mathbf{p}, S) < 0$), exterior ($\mathsf{PMC}(\mathbf{p}, S) > 0$), and boundary ($\mathsf{PMC}(\mathbf{p}, S) = 0$), the former two being dense 3D sets with nonzero volume, while the latter is a 2D set with zero volume and nonzero surface area [331]. It can also be used to construct a signed distance function (also a shape descriptor) $\xi : \mathbb{R}^3 \times \mathsf{S} \to \mathbb{R}$ defined as

$$\xi(\mathbf{p}; S) = \mathbf{1}(\mathbf{p}; \complement S) \inf_{\mathbf{q} \in iS} \|\mathbf{p} - \mathbf{q}\|_2 - \mathbf{1}(\mathbf{p}; iS) \inf_{\mathbf{q} \in \complement S} \|\mathbf{p} - \mathbf{q}\|_2 \tag{3.2.6}$$

$$= \mathbf{1}(\mathbf{p}; \complement S) \min_{\mathbf{q} \in \partial S} \|\mathbf{p} - \mathbf{q}\|_2 - \mathbf{1}(\mathbf{p}; iS) \min_{\mathbf{q} \in \partial S} \|\mathbf{p} - \mathbf{q}\|_2 \tag{3.2.7}$$

$$= \mathsf{PMC}(\mathbf{p}; S) \min_{\mathbf{q} \in \partial S} \|\mathbf{p} - \mathbf{q}\|_2, \tag{3.2.8}$$

where $\|\cdot\|_2 : \mathbb{R}^3 \to \mathbb{R}$ is the Euclidean L_2-norm. For the case of r-sets, the infima (i.e., greatest lower bounds) over the open sets iS and $\complement S$ in (3.2.6) can be replaced with minima over their closures $S = (iS \cup \partial S)$ and $(\complement S \cup \partial S)$, respectively, which is in turn replaced with minima over the closed shared boundary ∂S. Note that the PMC information is also directly retrievable as the sign of the signed distance function:

$$\mathsf{PMC}(\mathbf{p}; S) = \operatorname{sign}\left(\xi(\mathbf{p}; S)\right), \tag{3.2.9}$$

where we define $\operatorname{sign}(x) := x/|x|$ for all $x \in \mathbb{R} - \{0\}$ and $\operatorname{sign}(0) := 0$.

It is important to note that the indicator, PMC, and distance functions given in (3.2.4) through (3.2.8), respectively, all satisfy Assumption 3.2.1. Conversely, every shape descriptor that satisfies Assumption 3.2.1, i.e., is invariant under Euclidean isometries, can by definition be constructed as an explicit function of the distance function in (3.2.8). As I show in Section 3.4, the SDF family of shape descriptors are constructed by integrating a kernel applied to the distribution of the distance function over the boundary of the object.

Analytic representations provide powerful tools for shape and configuration modeling. For example, the regularized $\mathsf{C}-$space obstacle $S_1 \boxtimes^* S_2$ defined in (3.2.2) is the regularized sublevel set of the convolution of the indicators $\mathbf{1}(\cdot, S_1)$ and $\mathbf{1}(\cdot, S_2)$ [271, 272], which I elaborate in Section 3.3.

3.3 The Correlation Paradigm

Let $f_1, f_2 : \mathbb{R}^3 \to \mathbb{R}$ be real-valued functions over the 3−space that implicitly define the solids $S_1, S_2 \in \mathsf{S}$, respectively, i.e., $f_{1,2}(\mathbf{p}) := f(\mathbf{p}; S_{1,2})$ for all $\mathbf{p} \in \mathbb{R}^3$ such that $S_{1,2}$ are the regularized sublevel sets corresponding to $f_{1,2}(\mathbf{p}) \geq 0$ or $f_{1,2}(\mathbf{p}) \leq 0$, whichever is more convenient. Many important $\mathsf{C}-$space problems can be formulated and solved in terms of the cross-correlation function $f_{\text{CC}} : \text{SE}(3)^2 \times \mathsf{S}^2 \to \mathbb{R}$ defined as the following volume integral

$$f_{\text{CC}}\big(T_1, T_2; S_1, S_2\big) := \int_{\mathbb{R}^3} f\big(\mathbf{p}'; T_1 S_1\big) f\big(\mathbf{p}'; T_2 S_2\big)\, dV'. \tag{3.3.1}$$

The integration variable is $\mathbf{p}' = (x'_1, x'_2, x'_3) \in \mathbb{R}^3$ at which the volume element is denoted by $dV' = dx'_1 dx'_2 dx'_3$. This integration basically relocates the two objects $S_1, S_2 \in \mathsf{S}$ into their moved instances $T_1 S_1, T_2 S_2 \in \mathsf{S}$, respectively, overlaps the shape descriptor functions $f_1, f_2 \in L^2(\mathbb{R}^3)$ in the moved configurations, and aggregates their pointwise multiplication over the entire 3−space. Noting that by Assumption 3.2.1 the integrand functions are invariant under rigid isometries, i.e., $f(\mathbf{p}'; TS) = f(T^{-1}\mathbf{p}'; S)$

for both $S_1, S_2 \in \mathsf{S}$ and $T_1, T_2 \in \mathrm{SE}(3)$, by rearranging the terms we obtain

$$f_{\mathrm{CC}}\big(T_1, T_2; S_1, S_2\big) = \int_{\mathbb{R}^3} f\big(T_1^{-1}\mathbf{p}'; S_1\big) f\big(T_2^{-1}\mathbf{p}'; S_2\big)\, dV' \tag{3.3.2}$$

$$= \int_{\mathbb{R}^3} \big[(f_1 \circ T_1^{-1})(f_2 \circ T_2^{-1})\big](\mathbf{p}')\, dV', \tag{3.3.3}$$

where the notation $(f \circ T^{-1})(\mathbf{p}') = f(T^{-1}\mathbf{p}')$ is used for both $f_{1,2}(\mathbf{p}') = f(\mathbf{p}'; S_{1,2})$ and $T_1, T_2 \in \mathrm{SE}(3)$. The integral in (3.3.3) is a familiar algebraic construct:

$$f_{\mathrm{CC}}\big(T_1, T_2; S_1, S_2\big) = \Big\langle (f_1 \circ T_1^{-1}), (\bar{f}_2 \circ T_2^{-1}) \Big\rangle, \tag{3.3.4}$$

where $\langle \cdot, \cdot \rangle : L^2(\mathbb{R}^3) \times L^2(\mathbb{R}^3) \to \mathbb{C}$ denotes the inner product between square-integrable functions, detailed in Assumption 3.3.1. The notation $\bar{f}(\mathbf{p}) := \overline{f(\mathbf{p})}$ is used for the complex conjugate of $f(\mathbf{p})$ for the general case of complex functions $f : \mathbb{R}^3 \to \mathbb{C}$, which can be safely replaced with $f(\mathbf{p})$ for real functions $f : \mathbb{R}^3 \to \mathbb{R}$.

The formulation can be significantly simplified using a kinematic inversion. Let $T = (R, \mathbf{t}) = T_1^{-1} T_2 \in \mathrm{SE}(3)$ be the relative motion of S_2 observed from a frame attached to S_1. Using Assumption 3.2.1 once again and rearranging the terms in (3.3.1) using the change of integration variable to $\mathbf{p} = T_1^{-1}\mathbf{p}'$—i.e., the coordinates of the query point $\mathbf{p}' \in \mathbb{R}^3$ measured with respect to the frame attached to S_1—give the simplified function with a lower-dimensional domain $f_{\mathrm{CC}} : \mathrm{SE}(3) \times \mathsf{S}^2 \to \mathbb{R}$ as

$$f_{\mathrm{CC}}\big((R, \mathbf{t}); S_1, S_2\big) = \int_{\mathbb{R}^3} f\big(\mathbf{p}; S_1\big) f\big((R, \mathbf{t})^{-1}\mathbf{p}; S_2\big)\, dV \tag{3.3.5}$$

$$= \Big\langle f_1, (\bar{f}_2 \circ (R, \mathbf{t})^{-1}) \Big\rangle. \tag{3.3.6}$$

This time, the integration variable is $\mathbf{p} = (x_1, x_2, x_3) \in \mathbb{R}^3$ at which the volume

element is denoted by $dV = dx_1 dx_2 dx_3$.[8] Arriving from (3.3.5) to (3.3.6) follows the same set of steps taken to arrive from (3.3.1) to (3.3.4).

The integrals in (3.3.2) or (3.3.5) overlap the shape descriptor functions of the two r-sets in their transformed positions and orientations. In order for $f_{\mathrm{CC}}(T; S_1, S_2)$ to remain bounded, the integrand functions need to either be compactly supported—e.g., $f(\mathbf{p}; S) = 0$ if $\mathbf{p} \notin S$ as in the indicator function—or approach zero as $\mathbf{p} \to \infty$ with a sufficiently rapid rate for the integral to converge. This calls for the following additional assumption on the shape descriptors:

Assumption 3.3.1. The function $f : \mathbb{R}^3 \times \mathsf{S} \to \mathbb{R}$ (or $\mathbb{C}$) is square-integrable for all $S \in \mathsf{S}$, i.e., $f(\cdot; S) \in L^2(\mathbb{R}^3)$ meaning that the following integral converges:

$$\left\| f(\cdot; S) \right\|_2^2 := \left\langle f(\cdot; S), f(\cdot; S) \right\rangle = \int_{\mathbb{R}^3} \left| f(\mathbf{p}; S) \right|^2 \, dV. \tag{3.3.7}$$

This ensures that the inner product $\langle f_1, f_2 \rangle$ exists for $f_{1,2} = f(\cdot; S_{1,2})$ for all $S_1, S_2 \in \mathsf{S}$:

$$\left\langle f(\cdot; S_1), f(\cdot; S_2) \right\rangle := \int_{\mathbb{R}^3} f(\mathbf{p}; S_1) \bar{f}(\mathbf{p}; S_2) \, dV. \tag{3.3.8}$$

The inner product in (3.3.6) can also be interpreted as a 6D 'cross-correlation', i.e., a noncommutative convolution of the two shape descriptor functions over the 6D space SE(3), or equivalently, the commutative (and homogeneous) convolution $f_1 * (\bar{f}_2 \circ R^{\mathrm{T}})$ over the 3D space T(3) $\cong \mathbb{R}^3$ after a decomposition of the relative motion into rotational and translational components. But first, let us consider the physical and geometric meanings of these correlations.

[8] More technically, $dV = d\mu^3[\mathbf{p}]$ is the infinitesimal Lebesgue measure for a 3D element at $\mathbf{p} \in \mathbb{R}^3$. Depending on the global or local coordinate system that parameterizes the volume, the differential element can take different forms.

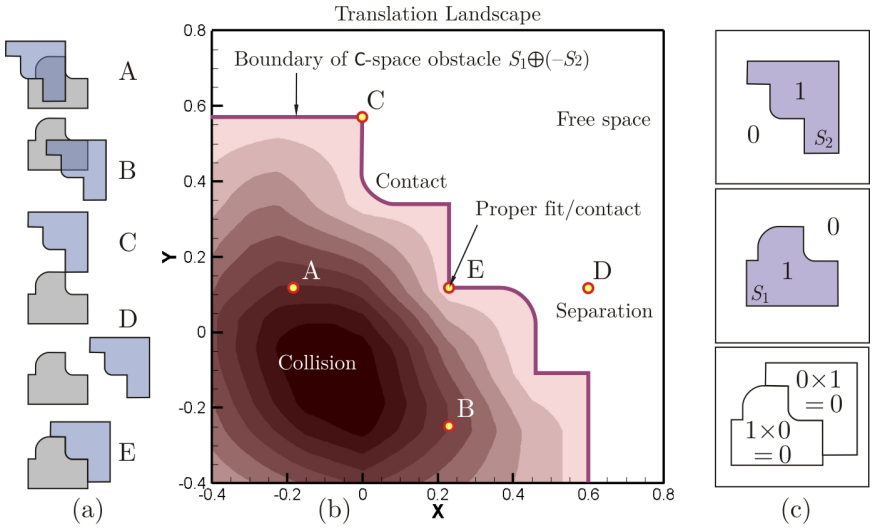


FIGURE 3.3.1: Different configurations (a) are evaluated using a gap function (b), based on overlapping indicator functions (c), formulated as a cross-correlation in (3.3.6).

3.3.1 Collision Response

The correlation paradigm presented above is central to important applications in geometric modeling [20, 21, 270] and group morphology [271, 272, 342]. For example, if we use the indicator function $f := \mathbf{1}$ defined earlier in (3.2.4), the integrals in (3.3.5) simply computes the volume of the intersection $(S_1 \cap TS_2)$—or equivalently, the volume of the intersection $(T_1S_1 \cap T_2S_2)$ of the moved instances of parts. Consequently, $f_{\mathrm{CC}} = 0$ characterizes the collection of feasible configurations corresponding to zero intersection volume, including unassembled (i.e., no-contact) and assembled (i.e., proper surface, curve, or point contact), while $f_{\mathrm{CC}} > 0$ implicitly defines the $\mathsf{C}-$space obstacles—i.e., regions of the $\mathsf{C}-$space that correspond to an interpenetration of parts. The holonomic unilateral contact constraints can be implemented by penalizing the moving part's energy function proportional to the collision volume given by the so-called gap function $f_{\mathrm{CC}}(T; S_1, S_2)$, and the collision impulse forces

and torques can be obtained from differentiating (3.3.5) using Lie algebras [270], to which I will return in Section 3.5.3.

Figure 3.3.1 (b) illustrates the translational $\mathsf{C}-$space landscape for a pair of 2D solids shown in panel (a), along with the colormap for the gap function. To make the illustration possible, the motion is restricted to translation only, i.e., the landscape in panel (b) is a section (corresponding to zero rotation) through the full 3D $\mathsf{C}-$space.[9] Each of the relative positions in panel (a) are represented by a point in panel (b). The gap function penalizes collision as in positions A and B ($f_{\mathrm{CC}} > 0$), but does not differentiate point contact in C and separation in D from proper fit/contact in E ($f_{\mathrm{CC}} = 0$). This is clearly due to the property $f_{1,2}(\mathbf{p}) := \mathbf{1}(\mathbf{p}; S_{1,2}) = 0$ for $\mathbf{p} \notin S_{1,2}$ as a result of the definition in (3.2.4), and the inevitable fact that the formulation in (3.3.5) as a volume integral is only able to measure properties that correspond to regions of nonzero volume (i.e., Lebesgue-measurable sets in 3−space) while those of point, line, and surface contact regions vanish during the course of integration.[10]

In principle, one could use this gap function approach along with fast Fourier methods for real-time analytic narrowphase CD [270] instead of traditional combinatorial CD methods (e.g., voxel-based methods [15, 287, 288, 347]) for guiding haptic assembly and disassembly. However, the difficulties faced in low-clearance assembly will continue to exist as a natural result of input data noise due to device inaccuracies and hand vibration, leading to unstable dynamic response and undesirable 'buzzing' in haptic feedback.

[9] In this case the $\mathsf{C}-$space obstacle defined implicitly with $f_{\mathrm{CC}} > 0$ is identical in shape with the Minkowski sum $S_1 \oplus (-S_2)$, where $-S_2$ denotes the reflection of S_2 with respect to the origin [20,21].

[10] This is a major caveat to keep in mind when using analytic approaches in general which only allow a one-to-one correspondence between regularized morphological concepts and measure-theoretic equivalents [271, 272].

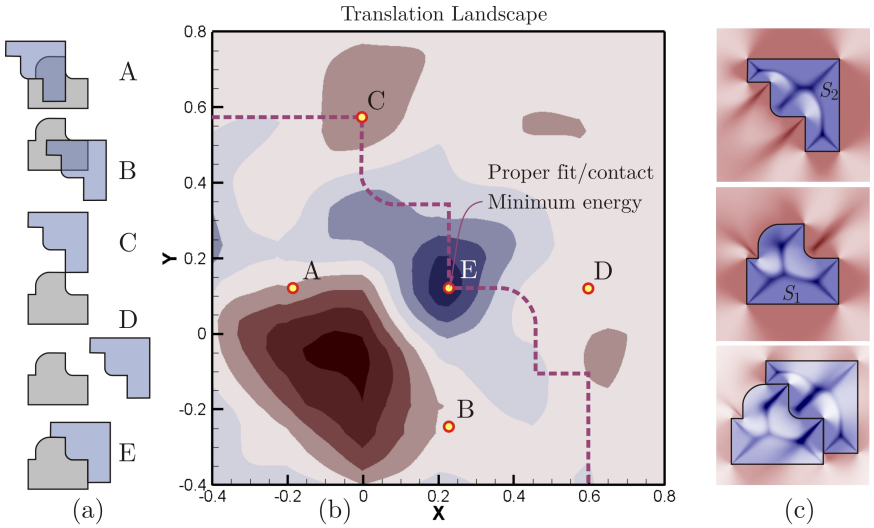


FIGURE 3.3.2: Different configurations (a) are evaluated using a score function (b), based on overlapping skeletal densities (c), formulated as a cross-correlation in (3.3.6).

3.3.2 Geometric Guidance

The question remains as how to modify the aforementioned approach in order to

1. incorporate nonzero values to the cross-correlation function over the free space and the contact curve in Fig. 3.3.1 (b), such that it rewards proper fit/contact (e.g., position E), slightly penalizes separation and insufficient contact (e.g., positions C and D), in addition to the high penalty already assigned to collision (e.g., positions A and B); and

2. provide a mechanism to adjust relaxation of the collision penalty, and control the smoothness of the transition between the free space and the obstacle space.

In other words, how can we add 'valleys' to the free space in addition to the 'ridges' inside the obstacle space, and how can we control the steepness of the transition between the two? The former allows additional haptic assistance for insertion, e.g., 'magnetic' attraction forces and torques that guide the assembly by restricting the

DOF of motion, and eventually snap the peg into the hole. The latter is important to let the user tune the flexibility of the collision response (i.e., 'soft' versus 'hard' assembly) as well as the degree of strictness with which the guidance constraints are enforced. This significantly improves performance and user experience in low-clearance assembly.

Figure 3.3.2 (b) illustrates a desirable score function that exhibits the aforementioned properties in contrast to the gap function in Fig. 3.3.1 (b). This is achieved by replacing the binary indicator function in panel (c) with a more powerful shape descriptor that effectively captures the geometric and topological properties of assembly features. The step-by-step development of these 'skeletal densities' is elaborated in Section 3.4 that follows.

3.4 Skeletal Density Functions

The assembly components need to be individually processed, each to be abstracted by certain shape descriptors that capture the most relevant geometric and topological characteristics to the virtual assembly task. This is probably the most challenging part of the entire process, especially when dealing with an infinitely large number of possibilities for complex surface features, each of which may or may not be the key determinant of proper assembly.

The answers to both questions in Section 3.3.2 lie in the development of more sophisticated shape descriptors $\rho_{1,2} = \rho(\mathbf{p}; S_{1,2})$ with $\rho(\mathbf{p}; S_{1,2}) \neq 0$ outside the object, rather than using, for example, the binary indicator functions or bump functions. I shall consider a special class of complex-valued functions $\rho_\sigma : (\mathbb{R}^3 - \partial S) \to \mathbb{C}$ that

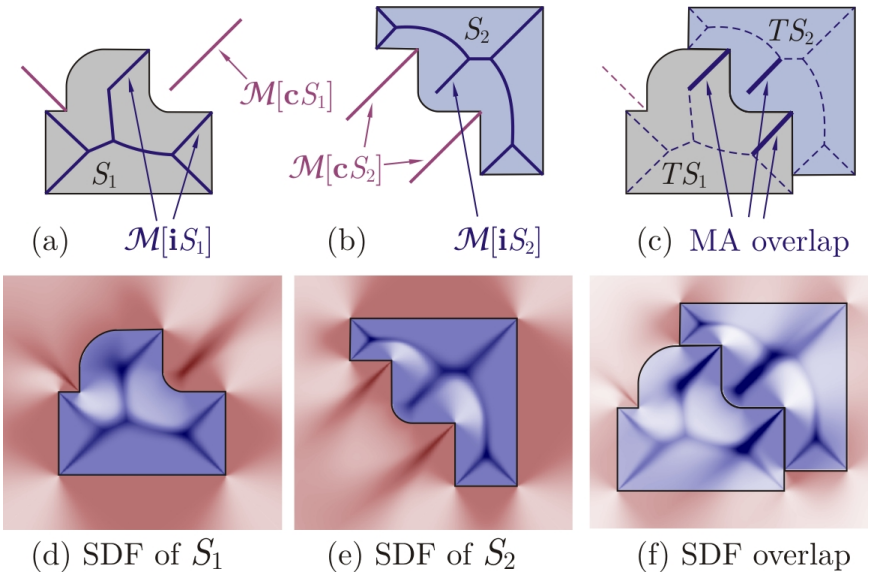


FIGURE 3.4.1: Assembly features captured by skeletal branches (a, b), which replace the virtual fixtures for assembly (c). The implicit skeletal density distribution (d, e) provides a robust substitute to facilitate measuring the overlap (f).

can be defined in the form of a surface flux integral over the solid boundary [19, 23].[11] The subsequent sections illustrate the step-by-step development.

3.4.1 Skeletal Overlaps

The basic premise of the proposed approach is that automatic identification of a proper fit in virtual assembly requires a quantification of the degree of effective geometric alignment, or *shape complementarity*, between pairs of objects. To achieve this, I make use of the new concept of continuous shape skeletons that I introduced in [19] for shape complementarity analysis of objects of arbitrary shape.

Geometric skeletons, such as the medial axis (MA), can be regarded as abstractions

[11] The exclusion of the object boundary ∂S from the domain of the shape descriptor function $\rho(\cdot; S)$ is necessitated by the ambiguity of the flux integral. However, as I shall explain in Section 3.4.3, this does not affect the outcome of the correlation integral.

of certain combinatorial, topological, and geometric information of the shape [253]. Figure 3.4.1 (a, b) shows the MAs of interiors $\mathscr{M}[\imath S_{1,2}]$ and exteriors $\mathscr{M}[\jmath S_{1,2}]$ of the 2D r-sets $S_1, S_2 \in \mathsf{S}$ (in this case $\mathsf{S} \subset \mathscr{P}(\mathbb{R}^2)$ for ease of illustration). These notions are defined as the loci of points $\mathbf{p} \in \mathbb{R}^2$ that have more than one *exact* nearest neighbor on the boundary ∂S [8], whose closures correspond to the singular points of the graph of the distance function $\xi(\mathbf{p}; S)$ defined in (3.2.8). It has been shown that this lower-dimensional set retains important topological information (e.g., the homotopy type [253]) of the original shape in a much more compact form.

The MA branches can be used as abstractions of the shape for assembly features—e.g., the two branches associated with the sharp corners and the one branch associated with the fillet feature in Fig. 3.4.1 (a, b). Therefore, one could try to overlap the external MA branches of one object with the internal MA branches of its mating object (and vice versa) to guide the assembly process, as in Fig. 3.4.1 (c). This suggests using MA geometry as a generic replacement for the virtual fixtures [343] mentioned earlier. This treatment is applicable to features of arbitrarily complex shape, and requires no user specification prior to or during the assembly, hence liberates automatic computation of the guidance forces and torques regardless of the model complexity.

Unfortunately, the traditional definition of the MA is very unstable with respect to small perturbations in the boundary, making it extremely difficult to compute and prune [8]. This provided the motivation to define a related concept in terms of a well-defined, space-continuous, and robust density distribution, called the skeletal density function (SDF), whose sublevel sets in the limit are related to an implicit definition of the MA [19]. Figure 3.4.1 (d, e) shows the SDF field, that depends on a 'thickness parameter' $\sigma > 0$. For $\sigma \ll 1$, the SDF value of the points on the MA—particularly those with more extensive nearest neighbors on ∂S—differentiates significantly from

the points outside the MA. As $\sigma \to 0^+$, the SDF is related to the defining function of the MA under certain restricting conditions [19]. We originally proposed the SDF in [19] as a proper shape descriptor for the purposes of completely automatic prediction of assembly relations (e.g., in assembly planning or protein docking). However, the concepts can be used more effectively to aid semiautomatic or manual assembly or docking in haptic-enabled interactive applications, as demonstrated here. I shall briefly review the concepts that lead to the formal definition of SDF, skipping rigorous elaborations in favor of clarifying the main ideas. The intention is to provide some insight into the applications of the SDF shape descriptors to define an energy model for haptic-assisted virtual assembly.

3.4.2 Distance Mapping

Given a solid $S \in \mathsf{S}$ of arbitrary shape, let us start by defining a Euclidean L^2-distance-based projection $\zeta : (\mathbb{R}^3 \times \partial S) \to \mathbb{C}$ of the boundary ∂S to the complex plane,[12] with respect to an arbitrary query point $\mathbf{p} \in \mathbb{R}^3$ as

$$\zeta(\mathbf{p}, \mathbf{q}; S) = \xi(\mathbf{p}; S) + \mathbf{i}\eta(\mathbf{p}, \mathbf{q}) = r(\mathbf{p}, \mathbf{q}; S)\ e^{\mathbf{i}\varphi(\mathbf{p},\mathbf{q};S)} \tag{3.4.1}$$

$$= \mathsf{PMC}(\mathbf{p}; S) \min_{\mathbf{q}' \in \partial S} \|\mathbf{p} - \mathbf{q}'\|_2 + \mathbf{i}\|\mathbf{p} - \mathbf{q}\|_2, \tag{3.4.2}$$

for which the simplified notation $\zeta = \xi + \mathbf{i}\eta = re^{\mathbf{i}\varphi} \in \mathbb{C}$ is sometimes used ($\mathbf{i}^2 = -1$).

The real-part $\Re\{\zeta(\mathbf{p}, \mathbf{q}; S)\} = \xi(\mathbf{p}; S)$ is the signed distance function defined earlier in (3.2.8), i.e., the distance from the nearest neighbor on the boundary $\mathbf{q}' \in \partial S$ to

[12] To keep it consistent with the notational conventions of the rest of this thesis, it is technically more accurate to use the signature $\zeta : \mathbb{R}^3 \times \mathbb{R}^3 \times \mathsf{S} \to \mathbb{C}$, but the 3rd argument is exceptionally thought of as a fixed *parameter* and restrict the domain for the 2nd argument from the entire $\mathbb{R}^3$ to ∂S in order to emphasize on the interpretation of this mapping as a projection of the boundary.

the query point $\mathbf{p} \in \mathbb{R}^3$, whose sign is determined by the PMC function $\mathsf{PMC}(\mathbf{p}; S)$ defined in (3.2.5), i.e., $\xi < 0$ for interior points ($\mathbf{p} \in \imath S$), $\xi = 0$ for boundary points ($\mathbf{p} \in \partial S$), and $\xi > 0$ for exterior points ($\mathbf{p} \in \complement S$). The imaginary-part $\Im\{\zeta(\mathbf{p}, \mathbf{q}; S)\} = \eta(\mathbf{p}, \mathbf{q})$ is simply the L_2-distance between one particular boundary point $\mathbf{q} \in \partial S$ and the query point $\mathbf{p} \in \mathbb{R}^3$. The magnitude $|\zeta(\mathbf{p}, \mathbf{q}; S)| = r(\mathbf{p}, \mathbf{q}; S)$ and the phase angle $\angle\zeta(\mathbf{p}, \mathbf{q}; S) = \varphi(\mathbf{p}, \mathbf{q}; S)$ are constrained as $r \geq \sqrt{2}|\xi|$ and $\pi/4 \leq \varphi \leq 3\pi/4$, respectively, due to the fact that by definition $|\xi| \leq \eta$.

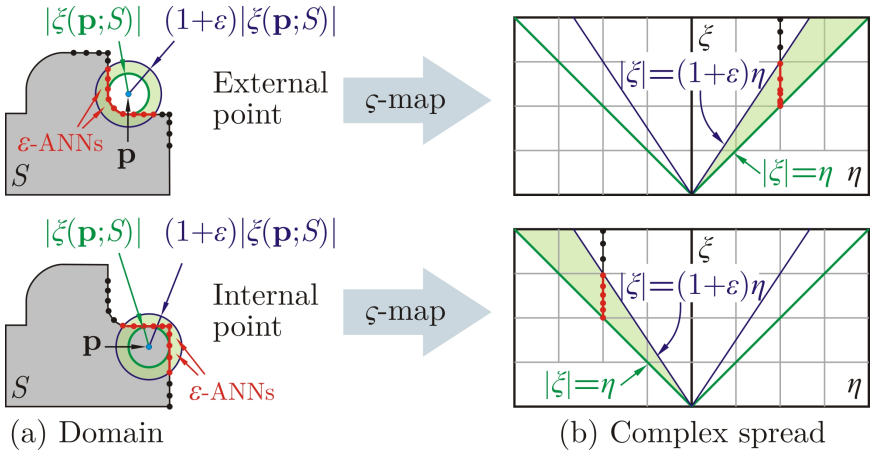
The so-obtained $\zeta-$mapping can be conceptualized as a projection of the boundary ∂S with respect to an arbitrary query point $\mathbf{p} \in \mathbb{R}^3$, with the following properties:

- The real-part $\xi(\mathbf{p}; S)$ is constant for a fixed query point $\mathbf{p} \in \mathbb{R}^3$, hence different boundary points $\mathbf{q} \in \partial S$—not to be confused with the nearest neighbor(s) $\mathbf{q}' \in \partial S$ of the query point on the boundary in (3.4.2)—are mapped to a segment along the vertical line $\xi = \text{const.}$ on the complex plane, called the 'complex spread' of the boundary and denoted as $\zeta(\mathbf{p}, \partial S; S)$.

- The location of the complex spread with respect to the imaginary axis is defined by the PMC; i.e., it is to the left, along, or to the right of $\xi = 0$, if the query point $\mathbf{p}$ is internal ($\mathbf{p} \in \imath S$), on the boundary ($\mathbf{p} \in \partial S$), or external ($\mathbf{p} \in \complement S$), respectively—see Fig. 3.4.2 (b).

- By definition, if $|\xi| \leq \eta$ then $|\tan\varphi| \geq 1$, the equality being exclusive to the boundary points $\mathbf{q} \in \partial S$ that are the closest to the query point $\mathbf{p} \in \mathbb{R}^3$—i.e., $|\tan\varphi| = 1$, i.e., $\varphi = \pi/4$ or $3\pi/4$, and $r = \sqrt{2}\xi = \sqrt{2}\eta$ iff $\mathbf{q} = \mathbf{q}'$ is the *exact* nearest neighbor of $\mathbf{p}$ on ∂S.

- For other boundary points at which $|\xi| < \eta$, the phase angle $\varphi = \angle\zeta$ can be



(a) Domain (b) Complex spread

FIGURE 3.4.2: The $\zeta-$mapping projects the boundary with respect to an internal/external query point to the left/right of the complex plane, respectively.

used as a determinant of the extent of normalized deviation for the boundary point $\mathbf{q} \in \partial S$ from being the nearest neighbor to the query point $\mathbf{p} \in \mathbb{R}^3$; namely, $|\tan\varphi| = \eta/|\xi| \leq (1+\epsilon)$ identifies the $\epsilon-$approximate nearest neighbors ($\epsilon-$ANNs)—see Fig. 3.4.2 (b).

The MAs of interior $\mathscr{M}[\imath S]$ and exterior $\mathscr{M}[\complement S]$ of an r-set $S \in \mathsf{S}$ are defined as the loci of points in the interior $\mathbf{p} \in \imath S$, and exterior $\mathbf{p} \in \complement S$, respectively, that have strictly more than one *exact* nearest neighbor on the boundary [8],[13] which can be implicitly defined by counting the number of points on the boundary ∂S that map to the same complex point $\zeta \in \mathbb{C}$ with $|\tan\varphi| = 1$. The strict condition on the existence of at least two points $\mathbf{q}_1, \mathbf{q}_2 \in \partial S$ that *exactly* satisfy $\eta(\mathbf{p}, \mathbf{q}_1) = \eta(\mathbf{p}, \mathbf{q}_2) = |\xi(\mathbf{p}; S)|$ makes the MA extremely unstable with respect to C^0- and C^1-perturbations of the boundary resulting from noise/errors in shape data, since a small perturbation of the surface geometry may result in large changes in the topology and geometry of

[13] Although I loosely refer to the MA as a type of shape skeleton, the exact definitions of 'medial axis', 'skeleton', and 'cut locus' (i.e., the closure of MA) are different (but closely related) for general open sets [73, 253].

the MA [8, 73]. In addition to the extremely difficult computation and refinement of the MA in practice, another challenge is of obtaining a shape complementarity score function that changes continuously with deviations in spatial relationships—i.e., a score function that properly rewards approximate overlap between MA branches, and penalizes separation between them. This is particularly difficult because not all branches are supposed to overlap (see Fig. 3.4.1 (c)) and those branches that do overlap might not exactly be coincident, especially in the presence of approximations. These problems can be solved by relaxing the aforementioned strict condition using *approximate* nearest neighbors, and by redefining the skeletal shape descriptors as space-continuous scalar fields whose overlaps can be quantified easily and robustly by SDF inner products (i.e., function integrals).

3.4.3 Shape Descriptors

Using the aforementioned distance mapping in (3.4.2), let us consider a special family of complex-valued shape descriptor functions $\rho_\sigma : (\mathbb{R}^3 - \partial S) \times \mathsf{S} \to \mathbb{C}$ that can be defined in the form of the following surface flux integral over the solid boundary [19]:

$$\rho(\mathbf{p}; S) = \oint_{\partial S} \phi\Big[\zeta\left(\mathbf{p}, \mathbf{q}; S\right)\Big] \, dA_\perp, \tag{3.4.3}$$

where $dA_\perp$ is the signed area element normal to the line segment that connects $\mathbf{p} \in \mathbb{R}^3$ to $\mathbf{q} \in \partial S$, i.e., $dA_\perp = \cos\theta(\mathbf{p}, \mathbf{q}; S) \, dA$, in which

$$\cos\theta(\mathbf{p}, \mathbf{q}; S) = \mathbf{n}(\mathbf{q}; S) \cdot \frac{\mathbf{p} - \mathbf{q}}{\|\mathbf{p} - \mathbf{q}\|_2}, \tag{3.4.4}$$

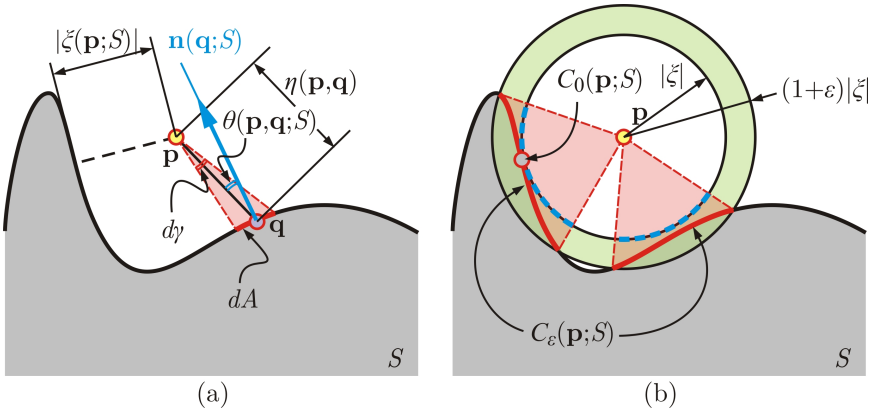


FIGURE 3.4.3: The terminology for the distance mapping in (3.4.2) and the shape descriptor integral in (3.4.3).

and $\theta(\mathbf{p},\mathbf{q};S)$ represents the smaller angle between the vector $(\mathbf{p}-\mathbf{q})$ and the outward unit normal vector $\mathbf{n}(\mathbf{q};S) \in \mathbb{R}^3$ on the boundary at $\mathbf{q} \in \partial S$. Figure 3.4.3 illustrates the different terms used in (3.4.3) and (3.4.4).[14]

The kernel $\phi : (\mathbb{C} - \{0\}) \to \mathbb{C}$ is the key, whose choice characterizes the set of topological and geometric properties that are extracted from the distance distribution embedded in the $\zeta-$map. The formulation as a surface integral in (3.4.3) characterizes a linear combination of the shares of different surface elements superimposed into $\rho_\sigma(\mathbf{p};S)$. In other words, the contribution of a boundary point $\mathbf{q} \in \partial S$ to the integral is determined by the profile of the $\phi-$kernel and how it changes depending on the relationship between the real- and imaginary-parts of $\zeta(\mathbf{p},\mathbf{q};S)$.

Before presenting a particular choice of the $\phi-$kernel in Section 3.4.4 for haptic assembly, let us consider a few simpler cases below. I aim to show that the gap function formulation in (3.3.5) with $f := \mathbf{1}$ in (3.2.4) as explained in Section 3.3.1 can be conceptualized as a special case of the generic formulation with $f := \rho$ in (3.4.3)

[14]More technically, $dA = d\mu^2[\mathbf{q}]$ is the infinitesimal Lebesgue measure for a 2D element at $\mathbf{q} \in \partial S$. Depending on the global or local coordinate system that parameterizes the surface, the differential element can take different forms.

using a particular $\phi-$kernel, which can later be modified to fulfill the additional requirements enumerated in Section 3.3.2.

First, consider a kernel that depends only on the imaginary-part of the argument (i.e., $\frac{\partial}{\partial \xi}\phi(\xi + \mathbf{i}\eta) = 0$) using an inverse-square decay formula:

$$\phi(\xi + \mathbf{i}\eta) := -\frac{1}{4\pi\eta^2}, \quad \text{i.e.,} \quad \phi(re^{\mathbf{i}\varphi}) := -\frac{1}{4\pi r^2 \sin^2 \varphi}. \tag{3.4.5}$$

Substituting this into (3.4.3) and rearranging the terms yield

$$\rho(\mathbf{p}; S) = \oint_{\partial S} -\frac{dA_\perp}{4\pi\eta^2} = \oint_{\partial S} -\frac{d\gamma}{4\pi} = -\frac{1}{4\pi}\Omega(\mathbf{p}; S) = \mathbf{1}(\mathbf{p}; S), \tag{3.4.6}$$

where $d\gamma = dA_\perp/\eta^2 = A_\perp/\|\mathbf{p} - \mathbf{q}\|_2^2$ is the signed infinitesimal spatial angle by which the query point $\mathbf{p} \in (\mathbb{R}^3 - \partial S)$ observes the area element dA at the boundary point $\mathbf{q} \in \partial S$, whose aggregation over the entire boundary is denoted by $\Omega(\mathbf{p}; S)$. Therefore, the surface integral in (3.4.6) computes the 'winding number'—which is commonly used for inclusion (i.e., PMC) testing of B-reps, e.g., polyhedral meshes [236]—i.e., the winding number is 1 in the interior, 0 in the exterior, and undefined on the boundary of the solid. Disregarding the boundary points—which do not contribute to the volume integral in (3.3.5)—this gives the indicator function defined earlier in (3.2.4) whose convolution yields the gap function in (3.3.5) for the collision response.[15]

I argued earlier in Section 3.3.2 that shape complementarity between objects of arbitrary shape can be related to the overlapping of shape skeleton branches that are generated by their assembly features—see Fig. 3.3.2 (d), for example, where the as-

[15]More precisely, the winding number is a 'restriction' of the indicator function denoted by $-\frac{1}{4\pi}\Omega(\cdot; S) = \mathbf{1}|_{(\mathbb{R}^3 - \partial S)}(\cdot; S)$, where the notation $f|_B : B \to f(B)$ denotes the restriction of a function $f : A \to f(A)$ to a subset of its domain $B \subseteq A$.

sembly features (i.e., the two sharp corners and one filleted corner fitting together) are each captured by overlapping branches of the MA. To develop an implicit continuous function that highlights the skeletal features, let us first modify the aforementioned choice of the kernel as $\phi(\zeta) \propto \eta^{-2}$ in (3.4.5)—which led to the winding number and a restriction of the indicator function as described above—to $\phi_\sigma(\zeta) \propto \eta^{-2} g_\sigma(\epsilon)$:

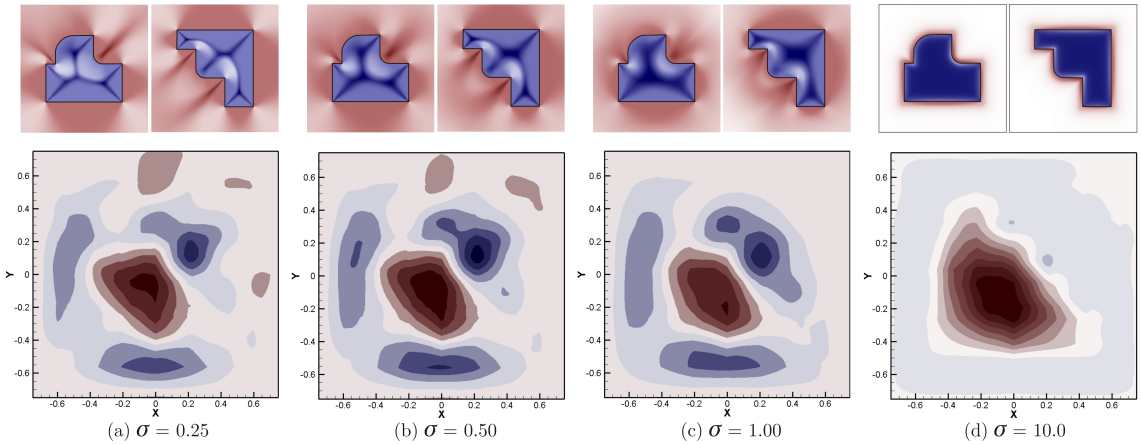
$$\phi_\sigma(\xi + \mathbf{i}\eta) := -\frac{1}{4\pi\eta^2} g_\sigma(\frac{\eta}{|\xi|} - 1), \quad \text{i.e.,} \tag{3.4.7}$$

$$\phi_\sigma(re^{\mathbf{i}\varphi}) := -\frac{1}{4\pi r^2 \sin^2\varphi} g_\sigma(|\tan\varphi| - 1), \tag{3.4.8}$$

where the additional term $g_\sigma : \mathbb{R} \to (0, \infty)$ is the Gaussian function:

$$g_\sigma(\epsilon) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}(\epsilon/\sigma)^2}, \tag{3.4.9}$$

where the parameter $\sigma > 0$ is the 'thickness factor'. The argument of the Gaussian $\epsilon = \eta/|\xi| - 1 = |\tan\varphi| - 1$ measures the normalized deviation from being an exact nearest neighbor of the query point $\mathbf{p} \in (\mathbb{R}^3 - \partial S)$ for the boundary point $\mathbf{q} \in \partial S$ for which $\zeta(\mathbf{p}, \mathbf{q}; S)$ in (3.4.2) is being computed. The additional Gaussian term thus incorporates a higher contribution to the surface integral in (3.4.3) for boundary elements at which $\tan\angle\zeta \approx \pm 1$ (i.e., $\eta \approx |\xi|$) which means $\mathbf{q}$ is an approximate nearest neighbor (ANN) of $\mathbf{p}$—or more precisely, one of its $\epsilon'-$ANNs for all $\epsilon' \geq \epsilon$. Query points near the shape skeleton have more extensive ANNs, hence receive more such contributions. As depicted earlier in Fig. 3.4.1, a continuous measure of skeletal overlap can be obtained by convolving such density functions using (3.3.5) for a given relative position and orientation of the two parts. The resulting change in the convolution function over the translational $\mathsf{C}-$space can be observed by comparing



(a) $\sigma = 0.25$ (b) $\sigma = 0.50$ (c) $\sigma = 1.00$ (d) $\sigma = 10.0$

FIGURE 3.4.4: The extent of geometric details captured by the skeletal density distribution is adjustable by the thickness factor σ.

Figs. 3.3.1 and 3.3.2, where the energy well is located at position E (i.e., the proper fit configuration) in the latter.

Figure 3.4.4 (a–d) illustrates the effect of changing the 'thickness factor' $\sigma > 0$ on the geometric energy landscape. Clearly, as $\sigma \to +\infty$ the Gaussian flattens out and the density function approaches the indicator function—with the exception of a signed coefficient, explained in Section 3.4.4—and as $\sigma \to 0^+$ the high density regions further resemble the MA.

As detailed in Section 3.4.4, for 3D interactions in haptic assembly I use a slightly different complex structure $\phi(\zeta) \propto \zeta^{-2} g_\sigma(\epsilon)$ with 1) a different inverse-square term, which also contributes a phase change; and 2) different signed coefficients $\pm\lambda_{1,2}$ for the interior and exterior query points, determined by the PMC function—as depicted by the blue $(-)$ and red $(+)$ colors in Figs. 3.3.2, 3.4.1, and 3.4.4. This allowed for positive real contributions to f_{CC} in (3.3.5) due to external-internal and internal-external skeletal overlaps (i.e., rewarding 'proper fit'), and negative real contributions due to internal-internal and external-external overlaps (i.e., penalizing 'collision' and

'separation'), the relative intensities of which could be adjusted by the choice of the 'penalty factor' $\mathfrak{p} = \lambda_2/\lambda_1$, elaborated in Section 3.5.

Aligning the skeletal branches (as in Fig. 3.4.1) can be thought of as a generalization of virtual fixtures [343] to arbitrary geometry, and implicit representation using SDF offers robustness and stability with respect to boundary perturbations, in contrast to the inherently unstable MA.

3.4.4 Complex Kernel

For a given query point $\mathbf{p} \in \mathbb{R}^3$, rather than counting the number of boundary points $\mathbf{q} \in \partial S$ for which $|\tan\angle\zeta| = 1$, which gives a discontinuous integer-valued defining function of the MA, it is desired to obtain a space-continuous complex-valued density function (the SDF [19]) $\rho_\sigma(\cdot; S) : (\mathbb{R}^3 - \partial S) \to \mathbb{C}$ using the generic form in (3.4.3). This is realized by first defining a kernel $\phi_\sigma : (\mathbb{C} - \{0\}) \to \mathbb{C}$ that takes the deviation of $|\tan\angle\zeta|$ from unity for the points on the complex spread of the boundary, and assigns a larger density to the query points $\mathbf{p} \in \mathbb{R}^3$ that have more extensive $\epsilon-$ANNs. This means a denser patch of points on $\zeta(\mathbf{p}, \partial S; S)$ with $|\tan\angle\zeta| \leq (1+\epsilon)$. The following definition serves these purposes:

$$\phi_\sigma(\zeta) := \lambda(\zeta)\phi^{\mathsf{p}}(\zeta)\phi^{\mathsf{m}}_\sigma(\zeta) = \lambda(\zeta) \cdot \frac{1}{\sqrt{2\pi}\zeta^2} \cdot g_\sigma\left(|\tan\angle\zeta| - 1\right), \tag{3.4.10}$$

$$= \begin{cases} -\frac{\lambda_2}{(2\pi\sigma)\zeta^2} \exp\left[\frac{-1}{2\sigma^2}(|\tan\angle\zeta| - 1)^2\right], & \text{if } \Re\{\zeta\} < 0, \\ 0 & \text{if } \Re\{\zeta\} = 0, \\ +\frac{\lambda_1}{(2\pi\sigma)\zeta^2} \exp\left[\frac{-1}{2\sigma^2}(|\tan\angle\zeta| - 1)^2\right], & \text{if } \Re\{\zeta\} > 0. \end{cases} \tag{3.4.11}$$

where a product of three different terms is used:

1. The 'coefficient function' $\lambda : \mathbb{C} \to \{-\lambda_2, 0, +\lambda_1\}$ that depends solely on the sign of the real-part of its argument:

$$\lambda(\xi + \mathbf{i}\eta) = \begin{cases} -\lambda_2 & \text{if } \xi < 0, \\ 0 & \text{if } \xi = 0, \\ +\lambda_1 & \text{if } \xi > 0, \end{cases} \tag{3.4.12}$$

   using the coefficients $0 < \lambda_1 < \lambda_2$ for reasons to become clear in Section 3.5. This gives a different sign and weight to the $\phi-$kernel based on whether $\mathbf{p}$ is external ($\xi > 0$) or internal ($\xi < 0$) to S, respectively.[16]

2. The 'proximal component' $\phi^{\mathsf{p}}(\zeta) := (\sqrt{2\pi}\zeta^2)^{-1}$, on the one hand, is provided to enforce two effects, namely,

   (a) an inverse-square decay of the skeletal density when the query point moves away from the boundary and the amplitude of the denominator $r^2 = |\zeta|^2 = (\xi^2 + \eta^2)$ increases with distance; and

   (b) a phase difference of $\angle\phi = -2\angle\zeta = -2\varphi$ which results in $\angle\phi \approx \pi \mp \pi/2$ for the high-density medial points with $\angle\zeta \approx \pi/2 \pm \pi/4$ (i.e., $|\tan\varphi| \approx 1$).

3. The 'medial component' $\phi^{\mathsf{m}}_{\sigma}(\zeta) := g_{\sigma}(\epsilon)$ for $\epsilon = |\tan\angle\zeta| - 1$, on the other hand, assigns larger densities to the medial points with $|\tan\angle\zeta| \approx 1$ with a continuous Gaussian decay given by (3.4.9) that is controllable by the parameter σ.

In the context of haptic-assisted assembly, the proximal component translates into a 'gravity' force between the assembly components (i.e., favoring proximity, hence the

[16] It would be more rigorous to put the parameters λ_1 and λ_2 next to the subscript σ of $\phi_{\sigma}(\zeta)$ in (3.4.10) and later $\rho_{\sigma}(\mathbf{p}; S)$ in (3.4.17), but I omit this syntactic detail to keep the notations clean.

name of the term), while the latter induces a sign changing mechanism that underlies the switch between the attraction and repulsion modes, when the parts are about to reach proper contact versus when they are about to penetrate, respectively. On the other hand, the medial component can be thought of as an implicit generalization of virtual fixtures [343] to arbitrary geometry. The energy contributions from overlapping medial regions impart specificity to the surface features which appear as forces and torques that try to align the skeletal branches and guide the insertion along these virtual fixtures.

Letting $1+\epsilon = |\tan\varphi| = \eta/|\xi|$ as a direct consequence of the definition in (3.4.2), one obtains the following bounds on the proximal component in (3.4.10):

$$\eta^2 \leq |\zeta|^2 = \xi^2 + \eta^2 \leq 2\eta^2 \;\Rightarrow\; \frac{1}{2}\frac{1}{\sqrt{2\pi}\eta^2} \leq |\phi^{\mathsf{p}}(\zeta)| = \frac{1}{\sqrt{2\pi}|\zeta|^2} \leq \frac{1}{\sqrt{2\pi}\eta^2}, \quad (3.4.13)$$

which results from the fact that $\epsilon \geq 0$, i.e., $0 < |\xi| \leq \eta$ for all $\mathbf{p} \notin \partial S$. On the other hand, it is easy to verify that the medial component in (3.4.10) is bounded as

$$0 < e^{-\frac{1}{2}(\epsilon/\sigma)^2} < e^{-\frac{1}{2}c^2} \;\Rightarrow\; 0 < |\phi^{\mathsf{m}}_{\sigma}(\zeta)| = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}(\epsilon/\sigma)^2} < \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}c^2}, \quad (3.4.14)$$

if $\epsilon/\sigma > c$ which means $\eta/|\xi| = |\tan\varphi| > (1+c\sigma)$ for some constant $c > 0$. Noting that the coefficient function in (3.4.12) is nonzero if $0 < \lambda_1 < \lambda_2$ and $\mathbf{p} \notin \partial S$, putting these results together yields the following lemma:

Lemma 3.4.1. *Given a query point* $\mathbf{p} \in (\mathbb{R}^3 - \partial S)$ *and a boundary point* $\mathbf{q} \in \partial S$ *and an arbitrary constant* $c > 0$*,*

$$|\tan\angle\zeta| > (1+c\sigma) \;\Rightarrow\; 0 < |\phi_{\sigma}(\zeta)| < \frac{|\lambda(\zeta)|}{2\pi\sigma\eta^2} e^{-\frac{1}{2}c^2}. \quad (3.4.15)$$

Proof. The proof is trivial by combining (3.4.13) and (3.4.14) and using (3.4.10). □

Corollary 3.4.2. *Given a query point* $\mathbf{p} \in (\mathbb{R}^3 - \partial S)$ *and a boundary point* $\mathbf{q} \in \partial S$ *and an arbitrary constant* $0 < c' < 2/\sigma$,

$$|\tan \angle\zeta| > 1 + \sigma\Big(-2\log\frac{\sigma c'}{2}\Big)^{\frac{1}{2}} \;\Rightarrow\; 0 < |\phi_\sigma(\zeta)| < \frac{|\lambda(\zeta)|}{4\pi\eta^2}c'. \tag{3.4.16}$$

Proof. Let $c'\sigma = 2e^{-\frac{1}{2}c^2}$ which can be solved for $c = (-2\log(\sigma c'/2))^{\frac{1}{2}}$. The assumption $0 < c' < 2/\sigma$ ensures $0 < \sigma c'/2 < 1$ hence $\log(\sigma c'/2) < 0$ thus the solution exists and $c > 0$. Substituting for this value of c in (3.4.15) yields (3.4.16). □

The results of Lemma 3.4.1 and Corollary 3.4.2 simply imply that if the phase of the $\zeta-$projection is large enough—i.e., $|\tan\varphi| = 1+\epsilon$ and $\epsilon > c\sigma$—then the $\phi-$kernel amplitude remains small enough—i.e., $|\phi(\zeta)| < |\lambda|(2\pi\eta^2)^{-1}c'$, while we already know about its phase being $\angle\phi(\zeta) = -2\varphi$. In other words, if the boundary point $\mathbf{q} \in \partial S$ is an $\epsilon-$ANN of the query point $\mathbf{p} \in (\mathbb{R}^3 - \partial S)$ for all $\epsilon > c\sigma$ given a large enough $c > 0$, then the contribution of that boundary point to the surface integral in (3.4.3) decays with an exponential rate $\propto e^{-\frac{1}{2}c^2}$ thus can be neglected with predictable error estimates using (3.4.15) and (3.4.16). For example, if $c = 3.4$ then $e^{-\frac{1}{2}c^2} < 10^{-4}$.

I will use these results in Section 3.4.5 to obtain error bounds on the $\rho-$function in (3.4.17) (i.e., the SDF [19]) when the computationally expensive integration over the entire boundary (i.e., global geometry) is approximated by one over the $\epsilon-$ANNs of the query point (i.e., local geometry) to gain substantial computational benefit.

3.4.5 Affinity Function

The next step is to apply the custom $\phi-$kernel in (3.4.10) to the complex spread of the object under consideration to obtain the SDF in (3.4.3), also known in this context as the 'affinity function' $\rho_\sigma(\cdot; S) : (\mathbb{R}^3 - \partial S) \to \mathbb{C}$ repeated here for convenience:

$$\rho_\sigma(\mathbf{p}; S) = \oint_{\partial S} \phi_\sigma\Big[\zeta\left(\mathbf{p}, \mathbf{q}; S\right)\Big]\, dA_\perp, \tag{3.4.17}$$

where $\zeta(\mathbf{p}, \mathbf{q}; S) = \xi(\mathbf{p}; S) + \mathbf{i}\eta(\mathbf{p}, \mathbf{q})$ was defined in (3.4.2), $\phi_\sigma(\zeta)$ was specified in (3.4.10) and (3.4.11), and $dA_\perp$ is the infinitesimal area of the projection of the surface element dA at the boundary point $\mathbf{q} \in \partial S$ on the sphere centered at the query point $\mathbf{p} \in (\mathbb{R}^3 - \partial S)$ with a radius of $\eta(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_2$, as illustrated in Fig. 3.4.3. If we let $\mathbf{v} = (\mathbf{p} - \mathbf{q})/\eta(\mathbf{p}, \mathbf{q})$ be the unit 'gaze vector', then we obtain $dA_\perp = (\mathbf{v} \cdot \mathbf{n})\, dA$ for the signed projected element and (3.4.17) becomes a flux integral of the radial vector field $\phi_\sigma(\zeta)\mathbf{v} : (\mathbb{C} - \{0\}) \to \mathbb{C}^3$ over the oriented piecewise C^1-manifold ∂S. The direction cosine $\mathbf{v} \cdot \mathbf{n} = \cos\theta$ is defined in (3.4.4) and illustrated in Fig. 3.4.3.

After substituting for $\phi_\sigma(\zeta)$ from (3.4.10) into (3.4.17) we obtain

$$\rho_\sigma(\mathbf{p}; S) = \frac{\lambda}{4\pi\sigma} \oint_{\partial S} \frac{e^{-\frac{1}{2}(\epsilon/\sigma)^2}}{(\xi + \mathbf{i}\eta)^2}\, dA_\perp = \frac{\lambda}{4\pi\sigma} \oint_{\partial S} \frac{e^{-\frac{1}{2}(\epsilon/\sigma)^2 - 2\mathbf{i}\varphi}}{\xi^2 + \eta^2}\, dA_\perp, \tag{3.4.18}$$

where $\epsilon = \eta/|\xi|$ and $|\tan\varphi| = 1 + \epsilon$ are variables within the scope of the integral, and λ can be treated as a constant since it depends on the PMC of the query point according to (3.4.12). Noting that $\eta = (1 + \epsilon)|\xi|$,

$$|\rho_\sigma(\mathbf{p}; S)| = \frac{|\lambda|}{4\pi\sigma} \oint_{\partial S} \frac{e^{-\frac{1}{2}(\epsilon/\sigma)^2}}{\xi^2 + \eta^2}\, |dA_\perp| = \frac{|\lambda|}{4\pi\sigma} \oint_{\partial S} \frac{e^{-\frac{1}{2}(\epsilon/\sigma)^2}}{1 + (1 + \epsilon)^{-2}}\, \frac{|dA_\perp|}{\eta^2}. \tag{3.4.19}$$

where $|dA_\perp| = |\mathbf{v} \cdot \mathbf{n}|\ dA$ is the unsigned projected element.

The exact computation of (3.4.17) is numerically demanding, as it requires spanning a discretization of the entire ∂S and computing the $\phi-$kernel for each and all boundary elements. However, if the SDF is meant roughly as an implicit generalization of the MA, it should (and does) depend predominantly on the local geometry; namely, on the boundary elements that carry $\epsilon-$ANNs. In what follows, I will show that the contribution of the rest of the boundary is negligible for a large enough $\epsilon > c\sigma$, and (3.4.17) can be approximated to an integral over $\epsilon-$ANNs alone:

$$\rho_\sigma(\mathbf{p}; S) \approx \oint_{C_\epsilon(\mathbf{p};S)} \phi_\sigma\Big[\zeta\left(\mathbf{p}, \mathbf{q}; S\right)\Big]\ dA_\perp, \tag{3.4.20}$$

where $C_\epsilon(\mathbf{p}; S)$ denotes the set of $\epsilon-$ANNs of the query point $\mathbf{p} \in \mathbb{R}^3$ on the boundary ∂S, i.e., those with a complex spread $\zeta(\mathbf{p}; C_\epsilon(\mathbf{p}; S); S)$ whose phase angles are bounded as $\tan^{-1}(1+\epsilon) \leq \varphi = \angle\zeta \leq \pi - \tan^{-1}(1+\epsilon)$:

$$C_\epsilon(\mathbf{p}; S) := \Big[B\big(\mathbf{p}, (1+\epsilon)|\xi(\mathbf{p}; S)|\big) -^* B\big(\mathbf{p}, |\xi(\mathbf{p}; S)|\big)\Big] \cap \partial S \tag{3.4.21}$$

$$= \big\{\mathbf{q} \in \partial S \ \big|\ |\tan\angle\zeta(\mathbf{p}, \mathbf{q}; S)| \leq (1+\epsilon)\big\}, \quad (\epsilon \geq 0) \tag{3.4.22}$$

where $B(\mathbf{p}, r) = \{\mathbf{p}' \in \mathbb{R}^3 \mid \|\mathbf{p} - \mathbf{p}'\|_2 \leq r\}$ is a closed ball of radius $r > 0$ centered at $\mathbf{p} \in \mathbb{R}^3$. The term in the brackets on the right-hand side of (3.4.21) is a closed spherical shell of inner radius $|\xi|$ and thickness ϵ, whose intersection with the boundary ∂S gives the set of $\epsilon-$ANNs. Figure 3.4.3 illustrates this definition. As depicted in Fig. 3.4.2, there is a one-to-many correspondence between every boundary point $\mathbf{q} \in C_\epsilon(\mathbf{p}; S)$ and its $\zeta-$projection $\zeta(\mathbf{p}, \mathbf{q}; S)$ that lies between the two pairs of radial lines $\tan\angle\zeta = \pm 1$ and $\tan\angle\zeta = \pm(1+\epsilon)$ in the complex plane. For the special

case $\epsilon = 0$, the set $C_0(\mathbf{p}; S)$ contains the exact nearest neighbors, and its cardinality $|C_0(\mathbf{p}; S)|$ implicitly defines the MA for both the interior $\dot{\iota}S$ and the exterior $\dot{\complement}S$:

$$\mathscr{M}[\dot{\iota}S] = \big\{\mathbf{p} \in \dot{\iota}S \;\big|\; |C_0(\mathbf{p}; S)| \geq 2\big\}, \tag{3.4.23}$$

$$\mathscr{M}[\dot{\complement}S] = \big\{\mathbf{p} \in \dot{\complement}S \;\big|\; |C_0(\mathbf{p}; S)| \geq 2\big\}, \tag{3.4.24}$$

while $C_\epsilon(\mathbf{p}; S)$ has at least one connected surface component for $\epsilon > 0$ hence has the cardinality of the continuum $|C_\epsilon(\mathbf{p}; S)| = |\mathbb{R}|$ [189]. Rather than 'counting' the number of points in these sets to define a notion of shape skeleton, the integral in (3.4.20) accumulates their contributions using the kernel structure in (3.4.10).

If $\epsilon \ll 1$, the spherical shell in Fig. 3.4.3 becomes a very thin one over which the distance $\eta(\mathbf{p}, \mathbf{q})$ is tightly bounded as $|\xi| \leq \eta \leq (1 + \epsilon)|\xi|$ hence it can be assumed to be a constant $\eta \approx \pm\xi$ hence $\zeta \approx (1 \pm \mathbf{i})\xi$ over the course of integration in (3.4.20). The minimum distance $\xi(\mathbf{p}; S)$ is already a constant within the scope of the integral, hence $\phi_\sigma(\zeta) = \phi_\sigma(\xi + \mathbf{i}\eta)$ can be factored out of the integral:

$$0 < \epsilon \ll 1 \;\Rightarrow\; \rho_\sigma(\mathbf{p}; S) = \oint_{C_\epsilon(\mathbf{p};S)} \phi_\sigma(\xi + \mathbf{i}|\xi|(1 + \epsilon))\; dA_\perp \tag{3.4.25}$$

$$\approx \phi_\sigma((1 \pm \mathbf{i})\xi) \oint_{C_\epsilon(\mathbf{p};S)} dA_\perp \tag{3.4.26}$$

$$= \phi_\sigma((1 \pm \mathbf{i})\xi) A_{\perp,\epsilon}(\mathbf{p}; S), \tag{3.4.27}$$

where $A_{\perp,\epsilon}(\mathbf{p}; S)$ denotes the total signed projected area of the the set $C_\epsilon(\mathbf{p}; S)$ over the surface of the spherical shell centered at the query point $\mathbf{p} \in (\mathbb{R}^3 - \partial S)$ depicted by the dashed patches in Fig. 3.4.3 (b). One could substitute for the first approximation for $A_{\perp,\epsilon}(\mathbf{p}; S) = \epsilon \frac{\partial}{\partial r} A_{\perp,r}(\mathbf{p}; S) + O(\epsilon^2)$.

Substituting for $\phi_\sigma(\zeta)$ from (3.4.10) in (3.4.27) and using the first approximation for $e^{-\frac{1}{2}(\epsilon/\sigma)^2} = 1 + O(\epsilon^2)$ in the Gaussian term we obtain

$$0 < \epsilon \ll 1 \;\Rightarrow\; \rho_\sigma(\mathbf{p}; S) \approx \frac{\pm\lambda_{1,2} e^{-\frac{1}{2}(\epsilon/\sigma)^2}}{2\pi\sigma(1 \pm \mathbf{i})^2 \xi^2} A_{\perp,\epsilon}(\mathbf{p}; S) \tag{3.4.28}$$

$$= \frac{-\mathbf{i}\lambda_{1,2}}{4\pi\sigma\xi^2} A_{\perp,\epsilon}(\mathbf{p}; S) = \frac{-\mathbf{i}\lambda_{1,2}}{4\pi\sigma} \Omega_\epsilon(\mathbf{p}; S), \tag{3.4.29}$$

where $\Omega_\epsilon(\mathbf{p}; S)$ denotes the total signed spatial angle of observation of the set $C_\epsilon(\mathbf{p}; S)$ from the query point $\mathbf{p} \in (\mathbb{R}^3 - \partial S)$ depicted by the shaded triangles in Fig. 3.4.3 (b). Note that $(1 \pm \mathbf{i})^{-2} = \mp\mathbf{i}/2$, and the $\lambda-$coefficients λ_1, λ_2 as well as the $+/-$ signs correspond to the exterior/interior points $\mathbf{p} \in (\mathbb{R}^3 - \partial S)$, respectively. The more extensive surface regions the query point has as its $\epsilon-$ANNs encapsulated inside the thin spherical shell of inner radius $|\xi|$ and outer radius $(1+\epsilon)|\xi|$, the larger the spatial angle $\Omega_\epsilon = A_{\perp,\epsilon}/\xi^2$ becomes, and the stronger the $\rho-$function becomes at that query point. As the query point moves away from the boundary and the inner radius of the spherical shell increases, the $\rho-$function decays with a rate $\propto \xi^{-2}$. The former explains the effect of the medial component while the latter is due to the proximal component implemented into the $\phi-$kernel in (3.4.10).

To make the notion of approximation in (3.4.20), (3.4.27), and (3.4.29) rigorous, we need to obtain a measure of the error due to *truncating* the integration domain from ∂S in (3.4.17) into $C_\epsilon(\mathbf{p}; S)$ in (3.4.20) and neglecting the share of $\partial S - C_\epsilon(\mathbf{p}; S)$. Let us first divide the integration domain in (3.4.17) as

$$\rho_\sigma(\mathbf{p}; S) = \rho^*_{\sigma,\epsilon}(\mathbf{p}; S) + E_{\sigma,\epsilon}(\mathbf{p}; S), \tag{3.4.30}$$

in which the first term on the right-hand side is the approximation given by (3.4.20),

hereafter denoted by an asterisk to make the distinction with the exact function, and the second term is the error:

$$\rho^*_{\sigma,\epsilon}(\mathbf{p};S) = \oint_{C_\epsilon(\mathbf{p};S)} \phi_\sigma(\zeta)\ dA_\perp, \tag{3.4.31}$$

$$E_{\sigma,\epsilon}(\mathbf{p};S) = \oint_{\partial S - C_\epsilon(\mathbf{p};S)} \phi_\sigma(\zeta)\ dA_\perp. \tag{3.4.32}$$

Next I wish to establish bounds on the error $E_{\sigma,\epsilon}(\mathbf{p};S) \in \mathbb{C}$ in terms of both its amplitude and its phase. To accomplish the former first, let us apply the triangle inequality theorem for integrals to (3.4.32):

$$|E_{\sigma,\epsilon}(\mathbf{p};S)| = \left| \oint_{\partial S - C_\epsilon(\mathbf{p};S)} \phi_\sigma(\zeta)\ dA_\perp \right| \tag{3.4.33}$$

$$\leq \oint_{\partial S - C_\epsilon(\mathbf{p};S)} |\phi_\sigma(\zeta)||dA_\perp| \tag{3.4.34}$$

In order to use the bounds on the $\phi-$kernel obtained in Section 3.4.4, we note that for all $\mathbf{q} \in [\partial S - C_\epsilon(\mathbf{p};S)]$, we have $|\tan\zeta| > (1+\epsilon)$ as a result of the definition in (3.4.22). Hence the condition of Lemma 3.4.1 on the left-hand side of (3.4.15) holds over the entire course of integration in (3.4.34) if we choose $c = \epsilon/\sigma$, hence

$$|E_{\sigma,\epsilon}(\mathbf{p};S)| \leq \frac{|\lambda|}{4\pi} c' \oint_{\partial S - C_\epsilon(\mathbf{p};S)} \frac{|dA_\perp|}{\eta^2} \tag{3.4.35}$$

Noting that $[\partial S - C_\epsilon(\mathbf{p};S)] \subset \partial S$ and the integrand is nonnegative, the integration over the nonempty set $C_\epsilon(\mathbf{p};S)$ is also nonnegative, hence

$$|E_{\sigma,\epsilon}(\mathbf{p};S)| \leq \frac{|\lambda|}{4\pi} c' \oint_{\partial S} \frac{|dA_\perp|}{\eta^2} = \frac{|\lambda|}{4\pi} c' \oint_{\partial S} |d\gamma| = \frac{|\lambda|}{4\pi} c' \Omega^*(\mathbf{p};S), \tag{3.4.36}$$

where $|d\gamma| = |dA_\perp|/\eta^2$ is the unsigned infinitesimal spatial angle by which the query point $\mathbf{p} \in (\mathbb{R}^3 - \partial S)$ observes the area element $|dA|$ at the boundary point $\mathbf{q} \in \partial S$, whose aggregation over the entire boundary is denoted by $\Omega^*(\mathbf{p}; S)$. These results can be summarized into the following lemma:

Lemma 3.4.3. *Given a query point* $\mathbf{p} \in (\mathbb{R}^3 - \partial S)$ *against a solid* $S \in \mathsf{S}$*, the approximation error amplitude for using* (3.4.31) *instead of* (3.4.17) *is bounded as*

$$|E_{\sigma,\epsilon}(\mathbf{p}; S)| = |\rho_\sigma(\mathbf{p}; S) - \rho^*_{\sigma,\epsilon}(\mathbf{p}; S)| \leq \frac{|\lambda|}{4\pi\sigma} e^{-\frac{1}{2}(\epsilon/\sigma)^2} \Omega^*(\mathbf{p}; S). \tag{3.4.37}$$

Proof. The proof is trivial by substituting $c' = \frac{2}{\sigma} e^{-\frac{1}{2}c^2} = \frac{2}{\sigma} e^{-\frac{1}{2}(\epsilon/\sigma)^2}$ as in Corollary 3.4.2 into (3.4.36). □

It is interesting to note that if the object is convex, we have the following relationship between the signed and unsigned total spatial angles

$$S \text{ is convex and } \mathbf{p} \in \mathit{i}S \;\Rightarrow\; \frac{1}{4\pi}\Omega^*(\mathbf{p}; S) = -\frac{1}{4\pi}\Omega(\mathbf{p}; S) = \mathbf{1}(\mathbf{p}; S) = 1, \tag{3.4.38}$$

$$S \text{ is convex and } \mathbf{p} \in \mathit{c}S \;\Rightarrow\; \frac{1}{4\pi}\Omega^*(\mathbf{p}; S) < +\frac{1}{4\pi} \cdot (4\pi) = 1, \tag{3.4.39}$$

where (3.4.6) was used along with a few geometric considerations for convex objects that are easily verifiable. For nonconvex shapes the unsigned spatial angle can be larger, nevertheless it is reasonable to assume that $\Omega^*(\mathbf{p}; S) = O(1)$ and its product with the exponential term in (3.4.37) provides a reasonably small upperbound on the error amplitude. For example, if $\epsilon > 3.4\sigma$ then $e^{-\frac{1}{2}(\epsilon/\sigma)^2} < 10^{-4}$.

To obtain similar results for the error phase angle, we first note that if two points $w_1, w_2 \in \mathbb{C}$ are close enough to each other on the complex plane—i.e., $|w_2 - w_2| \leq \delta$ for small enough $\delta > 0$—and are far enough from the origin—i.e., $|w_1| > r_1$ for

large enough $r_1 > \delta$ or $|w_2| > r_2$ for large enough $r_2 > \delta$—then by simple geometric observations $|\angle w_2 - \angle w_1| \leq \sin^{-1}(\delta/r)$ where $r = \max\{r_1, r_2\}$. This is due to the fact that each point lies within a disk of radius δ centered at the other point, whose angle of observation from the origin (i.e., phase) is bounded when the origin lies outside the disk. Let us reconsider (3.4.19) to obtain the following inequality:

$$|\rho_\sigma(\mathbf{p}; S)| = \frac{|\lambda|}{4\pi\sigma} \oint_{\partial S} \frac{e^{-\frac{1}{2}(\epsilon/\sigma)^2}}{1 + (1+\epsilon)^{-2}} \, |d\gamma| \leq \frac{|\lambda|}{8\pi\sigma} \oint_{\partial S} |d\gamma| = \frac{|\lambda|}{8\pi\sigma} \Omega^*(\mathbf{p}; S). \quad (3.4.40)$$

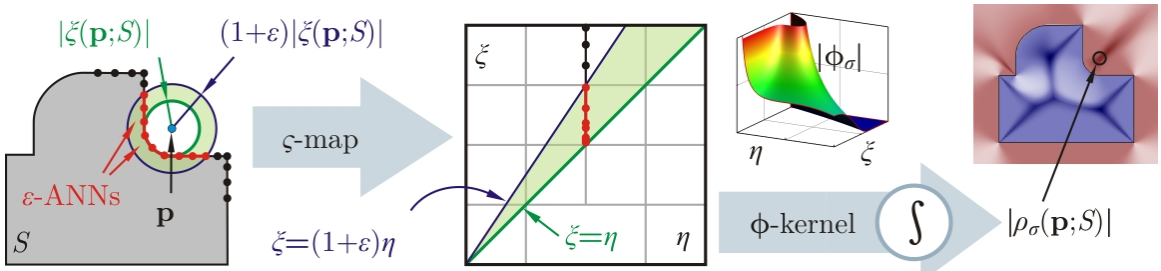
noting that $e^{-\frac{1}{2}(\epsilon/\sigma)^2} \leq 1$ and $[1 + (1+\epsilon)^{-2}] \geq 2$ for $\epsilon \geq 0$. This result combined with that of Lemma 3.4.3 leads to the following lemma:

Lemma 3.4.4. *Given a query point* $\mathbf{p} \in (\mathbb{R}^3 - \partial S)$ *against a solid* $S \in \mathsf{S}$*, the approximation error phase for using* (3.4.31) *instead of* (3.4.17) *is bounded as*

$$|\angle \rho_\sigma(\mathbf{p}; S) - \angle \rho^*_{\sigma,\epsilon}(\mathbf{p}; S)| \leq \sin^{-1}(\frac{1}{2} e^{-\frac{1}{2}(\epsilon/\sigma)^2}). \quad (3.4.41)$$

Proof. The proof is trivial by substituting for $w_1 := \rho_\sigma(\mathbf{p}; S)$ and $w_2 := \rho^*_{\sigma,\epsilon}(\mathbf{p}; S)$ and noting that $\delta = |\lambda|(4\pi\sigma)^{-1} e^{-\frac{1}{2}(\epsilon/\sigma)^2} \Omega^*$ (from (3.4.37)) and $r = |\lambda|(8\pi\sigma)^{-1}\Omega^*$ (from (3.4.40)) in the above argument. □

To summarize this section, I showed that the exact SDF given in (3.4.17) can be approximated with a *truncated* SDF given in (3.4.20) that is integrated over the regions of the boundary that form the $\epsilon-$ANNs of the query point $\mathbf{p} \in (\mathbb{R}^3 - \partial S)$, i.e., the patches of the boundary that lie within a closed spherical shell of internal radius $|\xi|$ and external radius $(1+\epsilon)|\xi|$. This can be explained in simple terms by the fact that the Gaussian term in (3.4.19) decays exponentially for the points outside the $\epsilon-$approximate nearest neighborhood; in fact, for $|\tan \angle\zeta| > (1+\epsilon)$, the exponential



(a) Domain (b) Complex spread (c) SDF

FIGURE 3.4.5: The affinity computation is decomposed into two steps: a $\zeta-$projection in (3.4.2) that characterizes the distance distribution as observed from the query point $\mathbf{p} \in \mathbb{R}^3$, followed by applying the $\phi-$kernel in (3.4.10).

term is at most $e^{-\frac{1}{2}(\epsilon/\sigma)^2}$, which is in turn less than 10^{-4} if we choose $\epsilon > 3.4\sigma$.

The surface integral in (3.4.20) aggregates the $\phi-$kernel over the $\epsilon-$ANNs to the query point on the boundary, assigning weights proportional to the spatial angles by which they are observed. This explains the choice of the inverse-square law in the proximal term for the $\phi-$kernel in (3.4.10) over any other possible decay function.

Figure 3.4.5 provides a schematic description of the SDF computation process, decomposed in principle into

1. computing a different representation of the object based on the Euclidean distance geometry in Section 3.4.2, i.e., the $\zeta-$projection of the boundary with respect to different query points; and

2. the application of a custom $\phi-$kernel in Section 3.4.4 to determine the distance distribution criteria, based on which the skeletal density is assigned in the 3−space according to the generic formulation in Section 3.4.3.

See [19] for examples of using different kernels and their applications.

3.4.6 Affinity Gradient

The affinity function developed in Section 3.4.5 will be used in Section 3.7 to compute the geometric energies for haptic assembly. Consequently, one needs to compute the affinity gradient to obtain the energy gradient (i.e., forces and torques) for real-time simulation, which we consider next.

The gradient of the affinity function $\nabla\rho_\sigma = \frac{d}{d\mathbf{p}}\rho_\sigma(\mathbf{p};S) : (\mathbb{R}^3 - \partial S) \to \mathbb{C}^3$ can be computed from (3.4.17) by applying the chain rule for differentiation:

$$\nabla\rho_\sigma(\mathbf{p};S) = \frac{d}{d\mathbf{p}} \oint_{\partial S} \phi_\sigma(\xi + \mathbf{i}\eta)\; dA_\perp \tag{3.4.42}$$

$$= \oint_{\partial S} \left[\frac{\partial \phi_\sigma}{\partial \xi}(\zeta)\nabla\xi + \frac{\partial \phi_\sigma}{\partial \eta}(\zeta)\nabla\eta \right]\; dA_\perp \tag{3.4.43}$$

$$= \oint_{\partial S} \left[\frac{\partial \phi_\sigma}{\partial \xi}(\zeta)\nabla^*\xi + \frac{\partial \phi_\sigma}{\partial \eta}(\zeta)\mathbf{v} \right]\; dA_\perp. \tag{3.4.44}$$

where the partial kernel derivatives $\frac{\partial}{\partial\xi}\phi_\sigma, \frac{\partial}{\partial\eta}\phi_\sigma : (\mathbb{C} - \mathbf{i}\mathbb{R}) \to \mathbb{C}$ can be obtained directly from (3.4.10) as follows:

$$\frac{\partial}{\partial\xi}\phi_\sigma(\xi + \mathbf{i}\eta) = \frac{\partial}{\partial\xi}\left[\frac{\lambda(\zeta)}{\sqrt{2\pi}\sigma(\xi + \mathbf{i}\eta)^2} e^{-\frac{1}{2\sigma^2}(\eta/|\xi|-1)^2} \right] \tag{3.4.45}$$

$$= \frac{\lambda(\zeta)}{\sqrt{2\pi}\sigma(\xi + \mathbf{i}\eta)^2} \left[\frac{\xi\eta}{\sigma^2|\xi|^3}\big(\frac{\eta}{|\xi|} - 1\big) - 2\frac{\xi - \mathbf{i}\eta}{\xi^2 + \eta^2} \right] e^{-\frac{1}{2\sigma^2}(\eta/|\xi|-1)^2}, \tag{3.4.46}$$

$$\frac{\partial}{\partial\eta}\phi_\sigma(\xi + \mathbf{i}\eta) = \frac{\partial}{\partial\eta}\left[\frac{\lambda(\zeta)}{\sqrt{2\pi}\sigma(\xi + \mathbf{i}\eta)^2} e^{-\frac{1}{2\sigma^2}(\eta/|\xi|-1)^2} \right] \tag{3.4.47}$$

$$= \frac{\lambda(\zeta)}{\sqrt{2\pi}\sigma(\xi + \mathbf{i}\eta)^2} \left[\frac{-1}{\sigma^2|\xi|}\big(\frac{\eta}{|\xi|} - 1\big) - 2\frac{\mathbf{i}\xi + \eta}{\xi^2 + \eta^2} \right] e^{-\frac{1}{2\sigma^2}(\eta/|\xi|-1)^2}, \tag{3.4.48}$$

noting that $\lambda(\zeta) \in \{+\lambda_1, -\lambda_2\}$ is a constant over each of the two halves of the

differentiation domain $\mathbb{C}-\mathbf{i}\mathbb{R}$. These results can be written in a simpler closed form:

$$\frac{\partial}{\partial\xi}\phi_\sigma(\zeta)=\phi_\sigma(\zeta)h_1(\zeta)=\lambda(\zeta)\phi^{\mathsf{p}}(\zeta)\phi_\sigma^{\mathsf{m}}(\zeta)h_1(\zeta) \tag{3.4.49}$$

$$\frac{\partial}{\partial\eta}\phi_\sigma(\zeta)=\phi_\sigma(\zeta)h_1(\zeta)=\lambda(\zeta)\phi^{\mathsf{p}}(\zeta)\phi_\sigma^{\mathsf{m}}(\zeta)h_2(\zeta), \tag{3.4.50}$$

where $h_1, h_2 : (\mathbb{C}-\mathbf{i}\mathbb{R}) \to \mathbb{C}$ are the two terms in the brackets in (3.4.46) and (3.4.48), respectively.

The term $\nabla\xi = \frac{d}{d\mathbf{p}}\xi(\mathbf{p};S) : \mathbb{R}^3 \to \mathbb{R}^3$ in (3.4.43) and (3.4.44) is the gradient of the signed Euclidean distance function $\xi(\mathbf{p};S)$; namely, $\nabla\xi = \pm(\mathbf{p}-\mathbf{q}^*)/\xi(\mathbf{p};S)$ where the sign depends on the PMC test of the query point, and $\mathbf{q}^* \in \partial S$ is the nearest neighbor to the query point $\mathbf{p} \in \mathbb{R}^3$ on the solid's boundary. This point is not unique when the query point belongs to the MA, i.e., if $\mathbf{p} \in (\mathscr{M}[iS] \cup \mathscr{M}[\mathscr{c}S])$, resulting in undefined $\nabla\xi$. For the sake of theoretical completeness, one could use the *extended* gradient $\nabla^*\xi$ from [253] which defines $\mathbf{q}^* \in \mathbb{R}^3$ as the center of the smallest enclosing ball that contains all nearest neighbors.[17] However, the resulted discontinuity does not create a problem when used in the convolution formulation for computing the guidance forces and torques in Section 3.7. This is because the MA of a 3D object is at most 2D (i.e., has zero volume) hence does not contribute to the volumetric integrals in (3.4.43) or (3.4.44).

The term $\nabla\eta = \frac{d}{d\mathbf{p}}\eta(\mathbf{p},\mathbf{q}) : (\mathbb{R}^3 - \{\mathbf{q}\}) \to \mathbb{R}^3$ in (3.4.43), on the other hand, is the gradient of the radial distance function $\eta(\mathbf{p},\mathbf{q})$ for a fixed $\mathbf{q} \in \partial S$, which is the same as the unit gaze vector $\mathbf{v} = (\mathbf{p}-\mathbf{q})/\eta(\mathbf{p},\mathbf{q})$.

The same approximation obtained for the $\rho-$function in Section 3.4.5 by restrict-

[17] The extended gradient of the distance function is central to the development of the concept of $\lambda-$medial axis ($\lambda-$MA) as a conditionally more stable substitute for the traditional MA [71, 72].

ing its integration domain to $\epsilon-$ANNs can be extended to the gradient function; namely,

$$\nabla\rho_\sigma(\mathbf{p};S) \approx \oint_{C_\epsilon(\mathbf{p};S)} \left[\frac{\partial\phi_\sigma}{\partial\xi}(\zeta)\nabla^*\xi + \frac{\partial\phi_\sigma}{\partial\eta}(\zeta)\mathbf{v} \right] dA_\perp, \tag{3.4.51}$$

for which similar error bounds to those expressed in Lemmas 3.4.3 and 3.4.4 can be obtained. Here I skip these details in favor of brevity.

3.5 Shape Complementarity

In a complex virtual assembly scene with many components, the analysis of the proper contact between all parts can be broken down in a bottom-up fashion into pairwise matching between the parts, and then between the resulted subassemblies, with an incremental growth of the number of constituents. This view is compatible with the actual process of semi-automatic haptic-enabled assembly, when the user drags and places the parts and resulting subassemblies one at a time.

In this section I use the SDF formulation developed in Section 3.4 for a pair of parts and combine them using the cross-correlation method presented in Section 3.3 to obtain a shape complementarity score function at every instant of the assembly process, from which the geometric energy field and its gradients (i.e., forces and torques) are derived in Section 3.7.

For a pair of solids $S_1, S_2 \in \mathsf{S}$ (each solid representing one part or subassembly as a single object), and their respective motions $T_1, T_2 \in \mathrm{SE}(3)$, the *shape complementarity* 'score function' $f_{\mathrm{SC}} : \mathrm{SE}(3)^2 \times \mathsf{S}^2 \to \mathbb{C}$ for this configuration can be obtained as a cross-correlation of individual SDF fields over the 3−space, i.e., by substituting $f := \rho_\sigma$ given by (3.4.17) or its approximation given by (3.4.20) for the shape descriptor in

the general correlation formula in (3.3.1):

$$f_{\mathrm{SC}}\big(T_1, T_2; S_1, S_2\big) = \int_{\mathbb{R}^3} \rho_\sigma\big(\mathbf{p}'; T_1 S_1\big)\rho_\sigma\big(\mathbf{p}'; T_2 S_2\big)\ dV'. \tag{3.5.1}$$

Following the same steps taken in Section 3.3 to arrive from (3.3.1) to (3.3.5), let $T = (R, \mathbf{t}) = T_1^{-1}T_2 \in \mathrm{SE}(3)$ be the relative configuration to reformulate the score function $f_{\mathrm{SC}} : \mathrm{SE}(3) \times \mathsf{S}^2 \to \mathbb{C}$ as

$$f_{\mathrm{SC}}\big((R, \mathbf{t}); S_1, S_2\big) = \int_{\mathbb{R}^3} \rho_\sigma\big(\mathbf{p}; S_1\big)\rho_\sigma\big(\mathbf{p}; (R, \mathbf{t})S_2\big)\ dV, \tag{3.5.2}$$

whose alternative form after applying Assumption 3.2.1 from Section 3.3 is

$$f_{\mathrm{SC}}\big((R, \mathbf{t}); S_1, S_2\big) = \int_{\mathbb{R}^3} \rho_\sigma\big(\mathbf{p}; S_1\big)\rho_\sigma\big((R, \mathbf{t})^{-1}\mathbf{p}; S_2\big)\ dV \tag{3.5.3}$$

$$= \Big\langle \rho_1, (\bar{\rho}_2 \circ (R, \mathbf{t})^{-1})\Big\rangle. \tag{3.5.4}$$

Once again, the notation $\rho_{1,2}(\mathbf{p}) = \rho_\sigma(\mathbf{p}; S_{1,2})$ is used for clarity. The binary operator $\langle \cdot, \cdot \rangle : L^2(\mathbb{R}^3) \times L^2(\mathbb{R}^3) \to \mathbb{R}$ is the bilinear and commutative inner product whose existence depends on Assumption 3.3.1 from Section 3.3. The standard definition of the inner product mandates the complex conjugation denoted by $\bar{\rho}_2(\mathbf{p}) = \overline{\rho_2(\mathbf{p})}$.

In the rest of this section, I investigate the decomposition of the motion in Section 3.5.1 and its application to the evaluation of the score function and its gradient in Sections 3.5.2 and 3.5.3, respectively, and their interpretation as convolutions. This will be of great utility for efficient alternative computations in the Fourier domain presented in Section 3.6.

3.5.1 Motion Decomposition

The only time-dependent variable in (3.5.2) through (3.5.4) is the 6D relative motion $T = (R, \mathbf{t}) \in \mathrm{SE}(3)$. To simplify the subsequent development, let us decompose the motion into the rotational component $R \in \mathrm{SO}(3)$ represented by a 3×3 special orthogonal matrix $[R]_{3\times3}$—'orthogonal' meaning $R^{-1} = R^{\mathrm{T}}$ (i.e., $\det(R) = \pm 1$) and 'special' meaning $\det(R) = +1$—and the translational component $\mathbf{t} \in \mathrm{T}(3) \cong \mathbb{R}^3$ described by a $3-$vector $[\mathbf{t}]_{3\times1}$.

As a result of the product structure $\mathrm{SE}(3) = \mathrm{SO}(3) \ltimes \mathrm{T}(3)$, the transformation sequence applies as $(R, \mathbf{t})\mathbf{p} = (R\mathbf{p}) + \mathbf{t}$ hence $(R, \mathbf{t})^{-1}\mathbf{p} = R^{\mathrm{T}}(\mathbf{p} - \mathbf{t})$. Noting the rotational invariance property of the SDF—required of any shape descriptor by Assumption 3.2.1 from Section 3.2.3 for the correlations to make sense—it is easy to see that for all $S \in \mathsf{S}$,

$$\rho_\sigma\big((R, \mathbf{t})^{-1}\mathbf{p}; S\big) = \rho_\sigma\big(R^{\mathrm{T}}(\mathbf{p} - \mathbf{t}); S\big) = \rho_\sigma\big((\mathbf{p} - \mathbf{t}); RS\big). \tag{3.5.5}$$

The integral in (3.5.2) thus can be interpretted as a 3D convolution over a section through the $\mathsf{C}-$space corresponding to the fixed rotation $R \in \mathrm{SO}(3)$:

$$f_{\mathrm{SC}}\big((R, \mathbf{t}); S_1, S_2\big) = \int_{\mathbb{R}^3} \rho_1(\mathbf{p}) \Big[\rho_2 \circ R^{\mathrm{T}}(\mathbf{p} - \mathbf{t})\Big]\, dV \tag{3.5.6}$$

$$= \big(\rho_1 * (\tilde{\rho}_2 \circ R^{\mathrm{T}})\big)\,(\mathbf{t}). \tag{3.5.7}$$

As before, the notation $\rho_{1,2}(\mathbf{p}) = \rho_\sigma(\mathbf{p}; S_{1,2})$ is used for clarity. The binary operator $* : L^2(\mathbb{R}^3) \times L^2(\mathbb{R}^3) \to L^2(\mathbb{R}^3)$ is the bilinear and commutative convolution which corresponds to homogeneous cross-correlations over the Euclidean $3-$space. The standard definition of the convolution mandates reflection with respect to the origin

denoted by $\tilde{\rho}_2(\mathbf{p}) = \rho_2(-\mathbf{p})$. The inverse rotation operator is treated as a (linear) function $R^{\mathrm{T}} : \mathbb{R}^3 \to \mathbb{R}^3$, hence $\rho_1 \in L^2(\mathbb{R}^3)$ corresponding to the SDF of the stationary object S_1 is convolved with $(\tilde{\rho}_2 \circ R^{\mathrm{T}}) \in L^2(\mathbb{R}^3)$ corresponding to the SDF of a reflected and rotated instance of S_2 to obtain $f_{\mathrm{SC}}\big((R, \cdot); S_1, S_2\big) \in L^2(\mathbb{R}^3)$ that scores all relative translations for the fixed rotation.

In practice, (3.5.7) is evaluated over a bounded cubic region of $\mathbb{R}^3$ that is large enough to cover the high density segments of the SDF branches, noting the inverse-square decay of the $\rho-$function due to the proximal component of the $\phi-$kernel in (3.4.10). This computation can be significantly streamlined using Fourier transforms presented in Section 3.6.

3.5.2 Scoring Function

At a given configuration, the integrals in (3.5.1) or (3.5.2) can be interpreted as an assessment of the degree of overlap between the continuous internal and external skeletons of one object, and the internal and external skeletons of the other object, hence four possible combinations contributing differently to the overall score function. I describe the four scenarios in simple terms to convey an intuitive understanding of the equations and how they fulfill the desired properties articulated in Section 3.4.1.

Consider a volume element at a point $\mathbf{p}' \in \mathbb{R}^3$ in (3.5.1) that belongs to a high skeletal density region of both $T_1 S_1$ and $T_2 S_2$, i.e., the query points happens to exhibit considerable values of both $|\rho_\sigma(\mathbf{p}'; T_1 S_1)|$ and $|\rho_\sigma(\mathbf{p}'; T_2 S_2)|$ hence contributing significantly to the integral in (3.5.1). Equivalently, one could speak of the same point as observed from the coordinate frame attached to S_1 as $\mathbf{p} \in \mathbb{R}^3$ and look at the values of $|\rho_\sigma(\mathbf{p}; S_1)|$ and $|\rho_\sigma(\mathbf{p}; T S_2)|$ in (3.5.2). Assume that the distance distribution over

the $\epsilon-$ANNs on the boundaries of the two objects as observed from the query point are similar, hence the medial and proximal components of the SDF are equally high with respect to both shapes, making the $\lambda-$function in (3.4.12) decide the separations between the following cases:

- if the query point is external to the 1st object and internal to the 2nd object, i.e., $\mathbf{p}' \in [\complement(T_1S_1) \cap \imath(T_2S_2)]$ thus $\mathbf{p} \in [\complement S_1 \cap \imath(TS_2)]$ then at that location the integrand is $\rho_1\rho_2 \propto (-\mathbf{i}\lambda_1)(+\mathbf{i}\lambda_2) = +\lambda_1\lambda_2 > 0$;

- if the query point is internal to the 1st object and external to the 2nd object, i.e., $\mathbf{p}' \in [\imath(T_1S_1) \cap \complement(T_2S_2)]$ thus $\mathbf{p} \in [\imath S_1 \cap \complement(TS_2)]$ then at that location the integrand is $\rho_1\rho_2 \propto (+\mathbf{i}\lambda_2)(-\mathbf{i}\lambda_1) = +\lambda_2\lambda_1 > 0$;

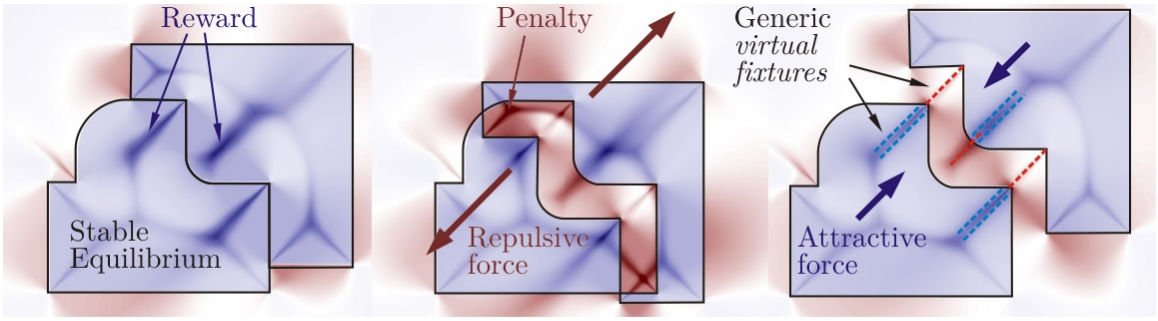
- if the query point is internal to the 1st object and internal to the 2nd object, i.e., $\mathbf{p}' \in [\imath(T_1S_1) \cap \imath(T_2S_2)]$ thus $\mathbf{p} \in [\imath S_1 \cap \imath(TS_2)]$ then at that location the integrand is $\rho_1\rho_2 \propto (+\mathbf{i}\lambda_2)(+\mathbf{i}\lambda_2) = -\lambda_2^2 < 0$;

- if the query point is external to the 1st object and external to the 2nd object, i.e., $\mathbf{p}' \in [\complement(T_1S_1) \cap \complement(T_2S_2)]$ thus $\mathbf{p} \in [\complement S_1 \cap \complement(TS_2)]$ then at that location the integrand is $\rho_1\rho_2 \propto (-\mathbf{i}\lambda_1)(-\mathbf{i}\lambda_1) = -\lambda_1^2 < 0$;

The above four cases can be categorized into the following three types of interaction:

- The first two cases above characterize the 'proper fit' alignment between the two objects since the exterior of one object is aligned with the interior of another with similar distance geometries carrying the hint of a proper complementary feature (Fig. 3.5.1 (a)).

- The third case above implies 'collision' at the observation point since the interior points are being overlapped and should be strictly prohibited (Fig. 3.5.1 (b)).



(a) Proper fit (b) Collision (c) Separation

FIGURE 3.5.1: Possible spatial relations and the corresponding interactions. The generic virtual fixtures practically restrict the DOF if the stiffness properties (i.e., 2nd-order partial derivatives of E_{G} at the energy well) are large enough.

- The fourth case above suggests a 'separation' at the observation point, which amounts less to a conclusion about the quality of fit (Fig. 3.5.1 (c)).

Hence if we choose $\lambda_1 = O(1)$ and $\mathfrak{p} := \lambda_2/\lambda_1 \gg 1$, then the first two cases contribute a positive-real *reward* of $\propto O(\mathfrak{p})$ to the score function, the third term contributes a large negative-real *penalty* of $\propto O(\mathfrak{p}^2)$, and the last term contributes a smaller penalty of $\propto O(1)$. The ratio $\mathfrak{p}$ is thus called the 'penalty factor'. These of course describe the distance geometry as observed from a single query point under consideration, which is why the overlap is integrated over different observation points via (3.5.1) or (3.5.2) to obtain the cumulative effect.

3.5.3 Scoring Gradient

For a function $f_{\mathrm{SC}} : \mathrm{SE}(3) \to \mathbb{C}$ whose domain is not a vector space, the generalized gradient function $\nabla f_{\mathrm{SC}} = (df_{\mathrm{SC}}/dR, df_{\mathrm{SC}}/d\mathbf{t}) : \mathrm{SE}(3) \to \mathbb{C}^6$ is composed of a 3D translational and a 3D rotational gradient vectors, characterizing the rate of change of the function with respect to infinitesimal translations and rotations, respectively.

The translational gradient function $df_{\mathrm{SC}}/d\mathbf{t} : \mathrm{SE}(3) \to \mathbb{C}^3$ is computed using basic concepts from linear algebra, since the translation space $\mathrm{T}(3) \cong \mathbb{R}^3$ is a vector space. Differentiating (3.5.6) and using the chain rule we obtain

$$\left\langle \frac{df_{\mathrm{SC}}}{d\mathbf{t}}, \mathbf{e} \right\rangle = - \int_{\mathbb{R}^3} \rho_1(\mathbf{p}) \Big[\nabla \rho_2 \circ R^{\mathrm{T}}(\mathbf{p} - \mathbf{t}) \Big] \cdot (R^{\mathrm{T}}\mathbf{e}) \; dV, \tag{3.5.8}$$

where $\mathbf{e} \in \mathbb{R}^3$ represents any direction in the vector space $\mathrm{T}(3) \cong \mathbb{R}^3$, along which the differentiation occurs. The term $(R^{\mathrm{T}}\mathbf{e})$ on the right-hand side of (3.5.8) can be factored out of the integral.

The rotational gradient function $df_{\mathrm{SC}}/dR : \mathrm{SE}(3) \to \mathbb{C}^3$ is more difficult to formulate, since SO(3) is not a vector space and cannot be globally parameterized by a single 3D chart. To construct a local parametrization, the tangent direction at $R \in \mathrm{SO}(3)$ is obtained as $R\Omega$ where $\Omega \in \mathfrak{so}(3)$ can be represented by a 3×3 skew-symmetric matrix $[\Omega]_{3\times 3}$, and $\mathfrak{so}(3)$ denotes the Lie algebra, which is a vector space tangent to SO(3) at the identity rotation [306]. Without getting into much detail, I present the rotational gradient as

$$\left\langle \frac{df_{\mathrm{SC}}}{dR}, \mathbf{e} \right\rangle = - \int_{\mathbb{R}^3} \rho_1(\mathbf{p}) \Big[\nabla \rho_2 \circ R^{\mathrm{T}}(\mathbf{p} - \mathbf{t}) \Big] \cdot \Omega^*(\mathbf{p} - \mathbf{t}) \; dV, \tag{3.5.9}$$

where $\mathbf{e} = R\mathbf{u}$ and $\mathbf{u} \in \mathbb{R}^3$ is the dual vector of $\Omega \in \mathfrak{so}(3)$, and $\Omega^* = R\Omega R^{\mathrm{T}}$ is sometimes referred to as the action of R^{T} on Ω.

For an in-depth treatment of translational and rotational differentiation and their interpretations as convolutions in terms of gradient and curl operators applied to the affinity functions, see [17]. The affinity gradient $\nabla\rho_2 = \nabla\rho_\sigma(\mathbf{p}; S_2)$ used in the integrands of (3.5.8) and (3.5.9) is computed from (3.4.44).

The 3D translational and rotational gradient vectors can be computed in a componentwise fashion by substituting for the base vectors $\mathbf{e} \in \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ one at a time in both (3.5.8) and (3.5.9). The complete 6D gradient $\nabla f_{\text{SC}} : \text{SE}(3) \rightarrow \mathbb{C}^6$ is defined as $\nabla f_{\text{SC}} = (df_{\text{SC}}/dR, df_{\text{SC}}/d\mathbf{t})$, obtained as

$$\nabla f_{\text{SC}} = \Big(\frac{df_{\text{SC}}}{dR}, \frac{df_{\text{SC}}}{d\mathbf{t}}\Big) = \Big(\sum_{1 \leq i \leq 3} \langle \frac{df_{\text{SC}}}{dR}, \mathbf{e}_i \rangle \mathbf{e}_i, \sum_{1 \leq i \leq 3} \langle \frac{df_{\text{SC}}}{d\mathbf{t}}, \mathbf{e}_i \rangle \mathbf{e}_i \Big). \tag{3.5.10}$$

3.6 Fourier Convolutions

The true strength of the analytic formulation using correlation functions is only realized when the convolutions are computed in the Fourier domain where they are converted to pointwise multiplications. In this section I present the alternative forms of the correlation function f_{SC} in (3.5.7) (in Section 3.6.2) and the correlation gradient ∇f_{SC} in (3.5.10) (in Section 3.6.3) using Fourier transforms.

3.6.1 Fourier Transform

Using the orthonormal Fourier basis of the form

$$e^{2\pi \mathbf{i}(\boldsymbol{\omega} \cdot \mathbf{p})} = \cos 2\pi(\boldsymbol{\omega} \cdot \mathbf{p}) + \mathbf{i} \sin 2\pi(\boldsymbol{\omega} \cdot \mathbf{p}), \tag{3.6.1}$$

where $\mathbf{p}, \boldsymbol{\omega} \in \mathbb{R}^3$ and $\mathbf{i}^2 = -1$, one could decompose a function/signal $f : \mathbb{R}^3 \rightarrow \mathbb{C}$ (defined over the 3D physical domain $\mathbf{p} \in \mathbb{R}^3$) into its components captured by the function/signal $\hat{f} : \mathbb{R}^3 \rightarrow \mathbb{C}$ (defined over the 3D frequency domain $\boldsymbol{\omega} \in \mathbb{R}^3$). The

forward Fourier transform $\hat{f} = \mathfrak{F}\{f\}$ is thus defined as the inner product

$$\hat{f}(\boldsymbol{\omega}) = \left\langle f, e^{+2\pi\mathbf{i}(\boldsymbol{\omega}\cdot\mathbf{p})} \right\rangle = \int_{\mathbb{R}^3} f(\mathbf{p}) e^{-2\pi\mathbf{i}(\boldsymbol{\omega}\cdot\mathbf{p})}\, dV, \tag{3.6.2}$$

and the inverse Fourier transform $f = \mathfrak{F}^{-1}\{\hat{f}\}$ is defined as follows, to retrieve the function as a superposition of its orthogonal components:

$$f(\mathbf{p}) = \left\langle \hat{f}, e^{-2\pi\mathbf{i}(\boldsymbol{\omega}\cdot\mathbf{p})} \right\rangle = \int_{\mathbb{R}^3} \hat{f}(\boldsymbol{\omega}) e^{+2\pi\mathbf{i}(\boldsymbol{\omega}\cdot\mathbf{p})}\, dV. \tag{3.6.3}$$

The integrals in (3.6.2) and (3.6.3) are sometimes referred to as the continuous Fourier transform (CFT), whose discretized form for computation purposes is the discrete Fourier transform (DFT) given in Section 4.1.4.

Substituting for the affinity function $f := \rho_\sigma(\cdot; S)$ from (3.4.17), let us denote its Fourier transform as $\hat{\rho}_\sigma(\cdot; S) = \mathfrak{F}\{\rho_\sigma(\cdot; S)\}$ (or simply $\hat{\rho}_\sigma = \mathfrak{F}\{\rho_\sigma\}$), i.e.,

$$\begin{aligned}
\hat{\rho}_\sigma(\boldsymbol{\omega}; S) &:= \int_{\mathbb{R}^3} \rho_\sigma(\mathbf{p}; S) e^{-2\pi\mathbf{i}(\boldsymbol{\omega}\cdot\mathbf{p})}\, dV && (3.6.4)\\
&= \int_{\mathbb{R}^3} \int_{\partial S} (\phi_\sigma \circ \zeta)(\mathbf{p}, \mathbf{q}; S) e^{-2\pi\mathbf{i}(\boldsymbol{\omega}\cdot\mathbf{p})}\, dA dV && (3.6.5)\\
&= \int_{\partial S} \int_{\mathbb{R}^3} (\phi_\sigma \circ \zeta)(\mathbf{p}, \mathbf{q}; S) e^{-2\pi\mathbf{i}(\boldsymbol{\omega}\cdot\mathbf{p})}\, dV dA && (3.6.6)\\
&= \int_{\partial S} \mathfrak{F}\Big\{ (\phi_\sigma \circ \zeta)(\cdot, \mathbf{q}; S) \Big\}\, dA. && (3.6.7)
\end{aligned}$$

The simplified notations $\hat{\rho}_{1,2} := \mathfrak{F}\{\rho_{1,2}\} = \hat{\rho}_\sigma(\cdot; S_{1,2})$ are often used for $S_{1,2} \in \mathsf{S}$.

3.6.2 Transformed Function

The *convolution theorem* states that the Fourier transform of a convolution is the pointwise product of Fourier transforms [226], hence the integral in (3.5.7) converts in the frequency domain to the simple product

$$\hat{f}_{\mathrm{SC}}(\boldsymbol{\omega}) = \mathscr{F}\big\{f_{\mathrm{SC}}\big\} = \mathscr{F}\big\{\rho_1\big\}\mathscr{F}\big\{(\tilde{\rho}_2 \circ R^{\mathrm{T}})\big\}. \tag{3.6.8}$$

As a direct consequence of the definition in (3.6.2), linear transformations (e.g., the orthogonal rotations and reflections) commute with the Fourier transform—noting that $\boldsymbol{\omega} \cdot (R^{\mathrm{T}}\mathbf{p}) = (R\boldsymbol{\omega}) \cdot \mathbf{p}$ and $\boldsymbol{\omega} \cdot (-\mathbf{p}) = (-\boldsymbol{\omega}) \cdot \mathbf{p}$ in (3.6.2)—hence

$$\mathscr{F}\big\{\tilde{\rho}_2 \circ R^{\mathrm{T}}\big\} = \mathscr{F}\big\{\tilde{\rho}_2\big\} \circ R^{\mathrm{T}} = \tilde{\mathscr{F}}\big\{\rho_2\big\} \circ R^{\mathrm{T}}, \tag{3.6.9}$$

in which the Fourier transform of the reflected function $\hat{\tilde{\rho}}_2 = \mathscr{F}\{\tilde{\rho}\}$ is identical with the reflection of the Fourier transform $\tilde{\hat{\rho}}_2 = \tilde{\mathscr{F}}\{\rho_2\}$. The same is true for rotation, hence the identity given in (3.6.9). The score function in the physical domain can thus be computed by applying an inverse transform as

$$f_{\mathrm{SC}}\big((R, \mathbf{t}); S_1, S_2\big) = \mathscr{F}^{-1}\big\{\mathscr{F}\{\rho_1\}\left(\mathscr{F}\big\{\tilde{\rho}_2\big\} \circ R^{\mathrm{T}}\right)\big\}. \tag{3.6.10}$$

At a first glance, this might appear as aggravating the computational burden, by requiring the evaluation of three volume integrals rather than the one in (3.5.6). However, (3.6.2) and (3.6.3) can be computed efficiently using the FFT algorithm [102], as will be demonstrated in Section 4.1.4.

3.6.3 Transformed Gradient

As I shall elaborate in Section 3.7, the forces and torques can be obtained from the gradient of the geometric energy field $E_{\mathrm{G}} : \mathbb{R}^3 \to \mathbb{R}$, which is proportional to the real-part of the shape complementarity score function $f_{\mathrm{SC}} : \mathbb{R}^3 \to \mathbb{C}$, i.e., its rate of change with respect to infinitesimal translations and rotations. This is possible by differentiating either the physical domain formulation (3.5.6) or the frequency domain formulation in (3.6.10). Another important advantage of working in the Fourier domain is that the translational differentiation is replaced with a multiplier:

$$\left\langle \frac{df_{\mathrm{SC}}}{d\mathbf{t}}, \mathbf{e} \right\rangle = (2\pi\mathbf{i})\mathcal{F}^{-1}\Big\{ (\mathbf{e}\cdot\boldsymbol{\omega})\ \mathcal{F}\{\rho_1\}\left(\mathcal{F}\{\tilde{\rho}_2\}\circ R^{\mathrm{T}}\right)\Big\}, \tag{3.6.11}$$

where $\mathbf{e} \in \mathbb{R}^3$ represents any direction in the vector space $\mathrm{T}(3) \cong \mathbb{R}^3$, along which the differentiation occurs. Once again, the rotational differentiation is more involved since SO(3) is not a vector space and cannot be globally parameterized by a single 3D chart. As before, a local parametrization is made possible noting that the tangent direction at $R \in \mathrm{SO}(3)$ is obtained as $R\Omega$ where $\Omega \in \mathfrak{so}(3)$ can be represented by a 3×3 skew-symmetric matrix $[\Omega]_{3\times 3}$, and $\mathfrak{so}(3)$ denotes the Lie algebra for SO(3), which is a vector space tangent to SO(3) at the identity rotation [306]. Without getting into much detail, we present the rotational gradient as

$$\left\langle \frac{df_{\mathrm{SC}}}{dR}, \mathbf{e} \right\rangle = (2\pi\mathbf{i})\mathcal{F}^{-1}\Big\{ (R^{\mathrm{T}}\Omega^{*}\boldsymbol{\omega})\cdot\mathcal{F}\{\rho_1\}\left(\mathcal{F}\{\tilde{\boldsymbol{\rho}}_2^{*}\}\circ R^{\mathrm{T}}\right)\Big\}, \tag{3.6.12}$$

where $\boldsymbol{\rho}_2^{*} : \mathbb{R}^3 \to \mathbb{C}^3$ is a vector function defined as $\boldsymbol{\rho}_2^{*}(\mathbf{p}) = +\rho_2(\mathbf{p})\mathbf{p}$ which implies $\tilde{\boldsymbol{\rho}}_2^{*}(\mathbf{p}) = -\tilde{\rho}_2(\mathbf{p})\mathbf{p}$, whose Fourier transform $\mathcal{F}\{\tilde{\boldsymbol{\rho}}_2^{*}\}$ can be obtained by componentwise FFTs of its complex components projected on the 3 coordinate axes.

See [17] for an in-depth treatment of translational and rotational differentiation, illustrating how gradient and curl operators in the physical domain convert to dot and cross products in the frequency domain, respectively. At the end of the day, the convolution and differentiation operations are all recast in terms of simple arithmetic operations on the Fourier representations.

The 3D translational and rotational gradient vectors can be computed in a componentwise fashion by substituting for the base vectors $\mathbf{e} \in \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ one at a time in (3.6.11) and (3.6.12). The complete 6D gradient $\nabla f_{\mathrm{SC}} : \mathrm{SE}(3) \to \mathbb{C}^6$ is thus obtained as before using (3.5.10) repeated here:

$$\nabla f_{\mathrm{SC}} = \Big(\frac{df_{\mathrm{SC}}}{dR}, \frac{df_{\mathrm{SC}}}{d\mathbf{t}}\Big) = \Big(\sum_{1\leq i\leq 3} \langle \frac{df_{\mathrm{SC}}}{dR}, \mathbf{e}_i\rangle \mathbf{e}_i, \sum_{1\leq i\leq 3} \langle \frac{df_{\mathrm{SC}}}{d\mathbf{t}}, \mathbf{e}_i\rangle \mathbf{e}_i\Big). \tag{3.6.13}$$

3.7 Energies, Forces, and Torques

The described generic and continuous score distribution over the configuration space SE(3) rewards shape complementarity and penalizes collision and separation. This enables defining an artificial potential energy function $E_{\mathrm{G}} \propto \Re\{f_{\mathrm{SC}}\}$ for use in real-time haptic assembly.

3.7.1 Energy Function

I define the 'geometric energy' function $E_{\mathrm{G}} : \mathrm{SE}(3) \to \mathbb{R}$ simply as:

$$E_{\mathrm{G}}\big((R, \mathbf{t}); S_1, S_2\big) := -\gamma_{\mathrm{SC}} \cdot \Re\Big\{ f_{\mathrm{SC}}\big((R, \mathbf{t}); S_1, S_2\big)\Big\}, \tag{3.7.1}$$

where $\Re\{\cdot\}$ stands for the real-part, and the constant $\gamma_{\mathrm{SC}} > 0$ is provided to scale the dimensionless score function f_{SC} to proper energy units, before applying it to objects of certain mass and inertia properties in a scene, bearing in mind the possibility of other forces being present.

One could appreciate an interesting analogy between this artificial, purely geometric energy field, and physical energy fields such as the electrostatic effect. It immediately follows that the product of affinity functions can be conceptualized as a complex 'geometric potential', which applies on a complex 'geometric charge' density, whose magnitude is dictated by the $\lambda-$function in (3.4.12). Using this analogy, on the one hand, when charges on the two objects are imaginary numbers of the same sign, there is a positive-real contribution to the energy, implying a *repulsion* force (Fig. 3.5.1 (b)). On the other hand, when the charges are imaginary numbers of opposite signs, they contribute a negative-real energy, indicating an *attraction* force (Fig. 3.5.1 (c)). Both attractive and repulsive effects decay with distance, due to the inverse-square law embedded in the $\phi-$kernel in (3.4.10).

3.7.2 Energy Gradient

The conservative 'geometric force' and 'geometric torque' are obtained as the gradient of the potential geometric energy function with respect to the translational and rotational motion, respectively, which are directly obtained from the score gradient:

$$\mathbf{F}_{\mathrm{G}}\big((R,\mathbf{t});S_1,S_2\big) = -\frac{dE_{\mathrm{G}}}{d\mathbf{t}} = +\gamma_{\mathrm{SC}}\Re\left\{\frac{df_{\mathrm{SC}}}{d\mathbf{t}}\right\}, \tag{3.7.2}$$

$$\mathbf{T}_{\mathrm{G}}\big((R,\mathbf{t});S_1,S_2\big) = -\frac{dE_{\mathrm{G}}}{dR} = +\gamma_{\mathrm{SC}}\Re\left\{\frac{df_{\mathrm{SC}}}{dR}\right\}. \tag{3.7.3}$$

This can be consolidated into the 6D general force/torque vector $(\mathbf{T}_{\mathrm{G}}, \mathbf{F}_{\mathrm{G}}) = -\nabla E_{\mathrm{G}}$, where $\nabla E_{\mathrm{G}} : \mathrm{SE}^3 \rightarrow \mathbb{R}^6$ is the complete 6D energy gradient defined as

$$\nabla E_{\mathrm{G}} = -(\mathbf{T}_{\mathrm{G}}, \mathbf{F}_{\mathrm{G}}) = -\gamma_{\mathrm{SC}} \Re\{\nabla f_{\mathrm{SC}}\}. \tag{3.7.4}$$

The forces and torques computed from (3.7.2) and (3.7.3), respectively, are used next along with other effects in the VE to guide the user during the assembly and disassembly process. The implementation details follow in Chapter 4.

Chapter 4

Implementation

4.1 Numerical Procedures

In this section I present the numerical algorithms that carry out a discrete approximation of the SDF integrals and the subsequent Fourier transforms for a particularly simple representation (namely, triangular mesh B-reps) along with a straightforward complexity analysis of each step.

4.1.1 Representation

As described in Section 3.2.1, the analytic definition of the SDF in (3.4.17) does not impose any restriction on the representation scheme used to describe the solid objects in the scene. Any representation scheme that satisfies the informational completeness requirement [334] can be used; namely, one that supports a means to compute

1. unsigned Euclidean distance queries to the boundary points; and

2. point membership classification (PMC) tests [395] to correct the sign of the distance function;

both to an adequate accuracy with respect to the smallest surface features. Nevertheless, the numerical computation of the surface integral in (3.4.17) lends itself well to B-reps, and particularly to triangular mesh approximations. In particular, we use a data structure that contains the following:

- *combinatorial structure*: the adjacency and orientation information for the boundary faces, edges, and vertices—e.g., using oriented half-edge or barycentric decomposition data structures—and
- *metric information*: the coordinates of the boundary vertices and (optionally) vertex normals, which can be used to obtain the face normals defining a consistent orientation for the boundary manifold.[1]

Given a solid $S \in \mathsf{S}$, let us denote the underlying space[2] of a triangulation that approximates its boundary ∂S with $\Delta_n(S) = \bigcup_{j=1}^{n} \delta_j$, where the closed triangles are denoted by δ_j $(1 \leq j \leq n)$, the number of triangles (i.e., faces) is n, and the number of edges and vertices are both $O(n)$ [332].

For the computations detailed in the subsequent sections, the following tools are used on triangular mesh representations:

[1]The orientation of the three edges, or equivalently, the traversal ordering of the three vertices stored in each face's data structure together with vertex coordinates are sufficient information to obtain unit normal vectors for the faces. The inward/outward normal orientation that is defined consistently with the ordered edges/vertices (according to the right-hand rule) is also equivalent to and retrievable from the PMC.

[2]The 'underlying space' of a cell complex is the union of all cells in that complex. For a 2D meshed surface embedded in 3D, it means the 2D subspace of the 3−space occupied by all faces, edges, and vertices of the triangulation [332].

- NETGEN [356] is used to triangulate the boundary of solid parts stored in STEP format (exported from any commercial CAD software). The boundary vertices need to be sampled with adequate density to capture the local geometric features of the shape.

- Havoc3D [183] is used to obtain the unsigned distance field cumulatively over a 3D grid using interpolation-based polygon rasterization of the Voronoi diagram on the graphics hardware via OpenGL rendering pipeline and depth-buffer.

- To compute the PMC and correct the sign of the distance function, I take advantage of the winding number approach in [236], which can be thought of as a special case of the general formula in (3.4.3)—i.e., one with a simple inverse-square kernel $\phi(\xi + \mathbf{i}\eta) = (4\pi\eta^2)^{-1}$ in (3.4.6) of Section 3.4.3—which can be computed using the same subroutines that allow custom kernels.

4.1.2 Preprocessing

The sequence of preprocessing steps per each part can be summarized as

1. generating a mesh from the CAD model using NETGEN [356];

2. computing the unsigned distance function using Havoc3D [183];

3. evaluating the PMC function [236] (to correct the distance signs) using (3.4.6) with an inverse-square kernel; and

4. evaluating the SDF descriptor [19] using (3.4.17) with a combined Gaussian and inverse-square kernel.

The last two steps can be implemented by approximating the integral in (3.4.3)—which subsumes (3.4.6) and (3.4.17) as special cases differing in the choice of the $\phi-$kernel—as a discrete Riemann sum over the triangles. In order to ensure numerical stability one must guarantee an upperbound on the error, which is not possible by assigning a lumped weight to each triangle in the sum. This is due to the inverse-square term in the kernel, resulting in large errors as the query point gets closer to a particular triangle. To overcome this difficulty, the algorithm carries out adaptive recursive subdivisions of the triangles with a spatial angle-based threshold, i.e., until the triangle is subdivided to small enough pieces each observed from the query point by a small spatial angle $\delta\gamma = \delta A_{\perp}/(4\pi\eta^2)$.

The following sequence of computations is carried out over a 3D uniform grid $G_m(S)$ of m nodes, where $\mathbf{p}_i \in G_m(S)$ $(1 \leq i \leq m)$ denotes an arbitrary grid node:

1. the distance queries from the n triangles to the m grid nodes, whose minimum over all n triangles for each grid node yields the unsigned distance function $|\xi_i| = |\xi(\mathbf{p}_i; S)|$ $(1 \leq i \leq m)$ (using Havoc3D [183]);

2. the PMC test to correct the distance sign for $\zeta-$mapping in (3.4.2) and obtain $\xi_i = \xi(\mathbf{p}_i; S)$ $(1 \leq i \leq m)$ which requires integration of the signed spatial angles over all n triangles as observed from the m grid nodes using (3.4.6); and

3. applying the $\phi-$kernel in (3.4.10) to obtain the SDF $\rho_i = \rho_\sigma(\mathbf{p}_i; S)$ $(1 \leq i \leq m)$ which also requires integration over all n triangles as observed from the m grid nodes using (3.4.17).

Let $\mathbf{q}_j \in \Delta_n(S)$ $(1 \leq j \leq n)$ denote a representative point (e.g., the mid-point) on a triangle $\delta_j \subset \Delta_n(S)$, whose unit normal is $\mathbf{n}_j \in \mathbb{R}^3$ and surface area is $\delta A_j > 0$. Then

(3.4.3)–which subsumes (3.4.6) and (3.4.17) as special cases differing in the choice of the $\phi-$kernel—can be approximated by the following discrete form:

$$\rho(\mathbf{p}_i; S) \approx \sum_{j=1}^{n} \phi\Big[\xi_i + \mathbf{i}\eta_{i,j}\Big] \cos\theta_{i,j}\ \delta A_j, \tag{4.1.1}$$

where $\xi_i = \xi(\mathbf{p}_i; S)$, $\eta_{i,j} = \eta(\mathbf{p}_i, \mathbf{q}_j)$, and $\cos\theta_{i,j} = \mathbf{v}_{i,j}{\cdot}\mathbf{n}_j$, in which $\mathbf{v}_{i,j} = (\mathbf{p}_i - \mathbf{q}_j)/\eta_{i,j}$. A similar discrete from can be obtained for the affinity gradient integral given in (3.4.42) through (3.4.44).

As previously mentioned, the approximation in (4.1.1) is reliable if the spatial angle by which the triangle δ_j is observed from $\mathbf{p}_i$ is small (i.e., $\cos\theta_{i,j}\ \delta A_j \ll \eta_{i,j}^2$) which is not necessarily true for grid nodes that are close to the ∂S surface, constituting only a small fraction of all grid nodes. For those points, the triangle can be recursively subdivided into smaller faces, until an upperbound criterion on the spatial angle of observation is reached. Assuming that the number of recursions is $O(1)$, computing (4.1.1) takes $O(n)$ basic steps. Therefore, the computation of the SDF over the entire grid $G_m(S)$ takes $O(mn)$ steps. The grid cell size should be small enough to capture the geometric features of the shape by the SDF, which implies a lowerbound on m. This can be sped up to $O(m'n)$ where $m' \ll m$ by using adaptively sampled query points, for instance over an octree $Q_{m'}(S)$ composed of m' nodes but with the same minimum cell size as that of $G_m(S)$.[3] For a mesh $\Delta_n(S) = \bigcup_{j=1}^{n} \delta_j$ that approximates the boundary ∂S with the complex of n faces δ_j $(1 \le j \le n)$ and $O(n)$ vertices and edges, and a uniformly sampled 3D grid of m query points $G_m(S)$ that contains the

[3]Both uniform grid- and octree-based sampling of the space is implemented into the program, but the former is preferred in subsequent steps that involve FFT computations in Section 4.1.4. The classical FFT algorithm [102] requires uniform sampling, while a nonuniform FFT [323] would be necessary for octree sampling, whose implementation is beyond the scope of this project.

bounding box of S, all steps take $O(mn)$ sequential time and $O(m+n)$ memory space.

Both PMC and SDF field computations are implemented in parallel on the multi-core CPU, assigning different chunks of $G_m(S)$ to different processors, using the Boost C++ libraries [355] for multithreading.[4] Significant speed-ups are obtaibed that scale almost linearly with the number of dedicated cores. In addition, the 3D grid structure maps properly to the single-instruction multiple-thread (SIMT) execution model of the modern many-core GPUs, allowing us to further speed up the process using NVIDIA®'s compute-unified device architecture (CUDA). The SDF field needs to be precomputed offline, *only once* per each rigid part or subassembly, hence can be done with high precision with little concern about the computation time.

4.1.3 A Cascade Method

Before presenting the FFT-based convolution in Section 4.1.4, let us first look at a naïve approach to computing the convolution integral in (3.5.1). Let the two assembly partners S_1 and S_2 be represented with triangular meshes $\Delta_{n_1}(S_1)$ and $\Delta_{n_2}(S_2)$ composed of n_1 and n_2 triangles, respectively. Assuming that the SDF fields for the individual objects are precomputed separately over $G_{m_1}(S_1)$ and $G_{m_2}(S_2)$ grids attached to each body, at every instance of the dynamic simulation with $T_1, T_2 \in \mathrm{SE}(3)$ the score integral in (3.5.1) can be discretized into

$$f_{\mathrm{SC}}(T_1, T_2; S_1, S_2) \approx \sum_{i=1}^{m} \rho_\sigma(T_1^{-1}\mathbf{p}_i; S_1)\rho_\sigma(T_2^{-1}\mathbf{p}_i; S_2)\ \delta V, \tag{4.1.2}$$

[4] In the earlier implementation reported in [18, 23] I used OpenMP for the task of CPU multi-threading. A significant improvement in performance was observed by re-implementing the same steps using the Boost C++ [355] Thread library in [22, 24].

where $\mathbf{p}_i \in G_m(T_1S_1 \cap T_2S_2)$ is a node on a grid sampled uniformly over the intersection of the moved objects, and $\delta V = vol(G_m)/m$ is the cell volume of this grid. The SDFs are interpolated from the precomputed values in (4.1.1). To save in interpolation time, the integration grid is picked as a subset of the smaller SDF grid, hence $m \leq \min\{m_1, m_2\}$ and computing (4.1.2) takes $O(m)$ basic steps. The score gradient in (3.5.8) and (3.5.9), needed for the guidance forces and torques in (3.7.2) and (3.7.3), respectively, can be discretized in a similar fashion. Alternatively, one could approximate the gradient using the finite difference method (FDM) by multiple computations of (4.1.2) after applying small translational and rotational variations, along each of the 3 coordinate axes one at a time, to the absolute transformations $T_1 = (R_1, \mathbf{t}_1)$ and $T_1 = (R_2, \mathbf{t}_2)$, i.e.,

$$\left\langle \frac{df_{\text{SC}}}{d\mathbf{t}_1}, \mathbf{e} \right\rangle \approx \frac{f_{\text{SC}}((R_1, \mathbf{t}_1 + \delta\mathbf{e}), T_2; S_1, S_2) - f_{\text{SC}}((R_1, \mathbf{t}_1 - \delta\mathbf{e}), T_2; S_1, S_2)}{2\delta}, \quad (4.1.3)$$

$$\left\langle \frac{df_{\text{SC}}}{d\mathbf{t}_2}, \mathbf{e} \right\rangle \approx \frac{f_{\text{SC}}(T_1, (R_2, \mathbf{t}_2 + \delta\mathbf{e}); S_1, S_2) - f_{\text{SC}}(T_1, (R_2, \mathbf{t}_2 - \delta\mathbf{e}); S_1, S_2)}{2\delta}, \quad (4.1.4)$$

$$\left\langle \frac{df_{\text{SC}}}{dR_1}, \mathbf{e} \right\rangle \approx \frac{f_{\text{SC}}((e^{+\Omega\theta}R_1, \mathbf{t}_1), T_2; S_1, S_2) - f_{\text{SC}}((e^{-\Omega\theta}R_1, \mathbf{t}_1), T_2; S_1, S_2)}{2\theta}, \quad (4.1.5)$$

$$\left\langle \frac{df_{\text{SC}}}{dR_2}, \mathbf{e} \right\rangle \approx \frac{f_{\text{SC}}(T_1, (e^{+\Omega\theta}R_2, \mathbf{t}_2); S_1, S_2) - f_{\text{SC}}(T_1, (e^{-\Omega\theta}R_2, \mathbf{t}_2); S_1, S_2)}{2\theta}, \quad (4.1.6)$$

where $\mathbf{e} \in \mathbb{R}^3$ is the dual vector of the skew-symmetric matrix $\Omega \in \mathfrak{so}(3)$ and the rotational variation $e^{\pm\Omega\theta} \in \text{SO}(3)$ is the exponential map [306] that converts the axis-angle representation (i.e., a right-handed rotation of $\pm\theta$ around $\mathbf{e}$) to its matrix representation.

The computation of (4.1.2) and its derivatives to obtain the forces and torques can be easily parallelized on the CPU and GPU. Although the performance scales almost linearly with the number of cores, the running times are not adequately small to keep

up with the 1 kHz haptic rendering loop. The simplest remedy is to precompute the geometric energy E_{G} in (3.7.1) and/or its derivatives in (3.7.2) and (3.7.3) over a sampling of relative transformations in SE(3), and interpolate the sample in real-time. This is not practical (both in terms of time and memory) for a 6D configuration space, even with the powerful computers available today. Fortunately, for most assembly scenarios the motion during the insertion phase is constrained to 1 or 2 DOF. For example, if the rotational space is limited to a finite number of permissible relative orientations, the field can be precomputed and stored over a 3D translational sampling in T(3) for each orientation (i.e., a section through the configuration space). However, this approach goes against the philosophy of avoiding the multiphase approach and manual specifications, from which we set off to pursue this method.

The alternative method presented in Section 4.1.4 uses GPU-accelerated FFTs to enable real-time computations.

4.1.4 FFT Computations

By sampling the physical and frequency domain data over the 3D grids $G_m(S)$ and $\hat{G}_m(S)$, respectively, the volume integrals in (3.6.2) and (3.6.3) that define the forward and inverse CFTs, respectively, can be approximated by the sums

$$\hat{f}_k \approx \sum_{i=1}^{m} f_i e^{-2\pi \mathbf{i}(\boldsymbol{\omega}_k \cdot \mathbf{p}_i)} \, \delta V, \quad 1 \leq k \leq m, \tag{4.1.7}$$

$$f_i \approx \sum_{k=1}^{m} \hat{f}_k e^{+2\pi \mathbf{i}(\boldsymbol{\omega}_k \cdot \mathbf{p}_i)} \, \delta V, \quad 1 \leq i \leq m, \tag{4.1.8}$$

where $\mathbf{p}_i \in G_m(S)$ and $\boldsymbol{\omega}_k \in \hat{G}_m(S)$ are uniformly sampled physical and frequency nodes at which the function values $f_i := f(\mathbf{p}_i)$ and $\hat{f}_k := \hat{f}(\boldsymbol{\omega}_k)$ are stored, respec-

tively. With the exception of a constant factor (depending on the chosen conventions), these sums define the discrete Fourier transform (DFT) whose cumulative computation for all grid nodes takes $O(m^2)$ basic operations using the cascade method. However, the same computation can be carried out in $O(m \log m)$ steps using the radix-2 FFT algorithm [102].

A key observation is that the frequency domain representations of the SDFs, namely $\hat{\rho}_1 = \mathscr{F}\{\rho_1\}$, $\hat{\rho}_2 = \mathscr{F}\{\rho_2\}$, and $\hat{\boldsymbol{\rho}}_2^* = \mathscr{F}\{\boldsymbol{\rho}_2^*\}$ depend on part geometries alone—and not on the instantaneous assembly configuration—hence can be precomputed offline prior to the virtual assembly session. For a pair of parts, computing the forward FFT in (4.1.7) to obtain $\hat{\rho}_{1,k}, \hat{\rho}_{1,k}, \hat{\boldsymbol{\rho}}_{2,k}^*$ $(1 \leq k \leq m)$ from $\rho_{1,i}, \rho_{2,i}, \boldsymbol{\rho}_{2,i}^*$ $(1 \leq i \leq m)$ takes $O(m \log m)$ per part.

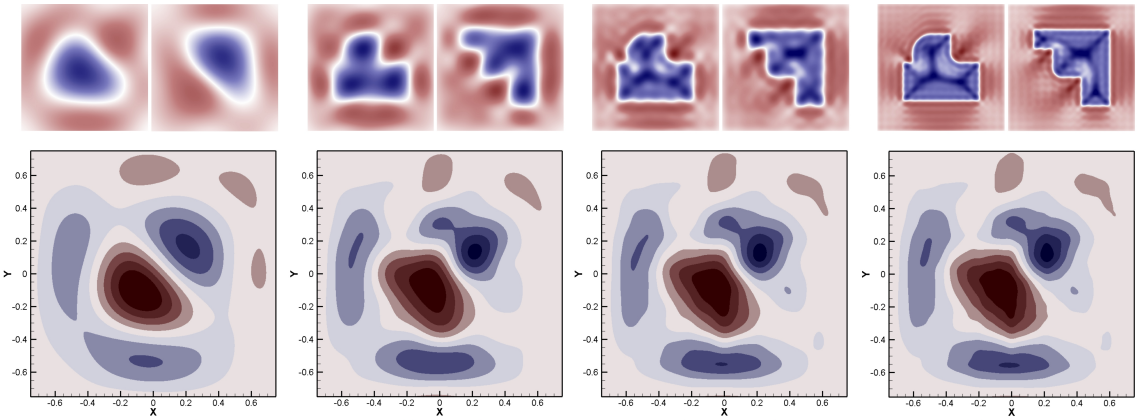
The subsequent computation of the convolutions in (3.5.7), (3.5.8), and (3.5.9) for geometric energy, force, and torque evaluations, respectively, in real-time for a particular relative orientation $R \in \mathrm{SO}(3)$ of the parts takes place entirely in the frequency domain. The sequence of operations is

1. interpolating $(\hat{\rho}_2 \circ R^{\mathrm{T}})$ and $(\hat{\boldsymbol{\rho}}_2^* \circ R^{\mathrm{T}})$ over a rotated grid $R^{\mathrm{T}}\hat{G}_m(S_2)$, followed by a reflection, which takes $O(m)$ basic trilinear interpolation steps;

2. a pointwise multiplication of the interpolated data with $\hat{\rho}_1$ data over $\hat{G}_m(S_1)$, which also takes $O(m)$ basic steps; and

3. an inverse FFT along with applying the proper coefficients—e.g., $2\pi\mathbf{i}$ in (3.6.11) and (3.6.12) for force/torque computations—which takes $O(m \log m)$ steps and yields the convolution results over a grid of translations.

Therefore, the total arithmetic complexity of the process (for a single rotation and



(a) $m' = 4^2$ (b) $m' = 8^2$ (c) $m' = 16^2$ (d) $m' = 32^2$

FIGURE 4.1.1: Frequency domain representation allows for a systematic means of successive approximation of the energy field.

all translations of interest) is $O(m \log m)$.[5]

The FFTW library [133, 134] is used for the CPU sequential implementation[6] and NVIDIA®'s cuFFT(W) is used for the GPU parallel implementation of the FFT, the running times of both to be presented in Section 5.2.2 for comparison.

4.1.5 Low-Pass Filtering

The additional important advantage of the analytic method is that one could decide to keep only a small subset of $m' \ll m$ frequency domain data nodes (i.e., the dominant modes) for computing the Fourier convolutions in (3.6.10), (3.6.11), and (3.6.12) in real-time, in a trade-off between the desired accuracy and available computational power. This results in a reduction of the real-time process complexity to $O(m' \log m')$ operations per frame, which is practically almost $O(1)$. In other words, the interpola-

[5]It is important to note that the fixation of orientation does not create a significant limitation to the application of this method, as most assembly activities can be planned with a finite number of permissible orientations during the constrained insertion.

[6]Unfortunately, FFTW does not provide a CPU-parallel implementation, which would be a nice addition to the performance results in Section 5.2.2 for comparison with those of the GPU.

tion (of the rotated and reflected data), the pointwise multiplication, and the inverse FFT steps are carried out for a significantly smaller sample of low-frequency grid nodes, a process referred to as 'low-pass filtering'. As we will demonstrate in Section 5.2.2, the pointwise multiplication step is the bottleneck and dictates an upperbound on the choice of m' in order to stay within a prespecified time allocated to guidance force and torque computations at each frame, hence to achieve the desired haptic frame rate of 1 kHz.[7]

It is worthwhile noting that the inverse FFT cumulatively computes the correlations for a range of m translations corresponding to a single rotation. This is a nonoptimal approach from a theoretical point of view, since we only need the results for a single configuration, i.e., the instantaneous relative translation and rotation of the objects $(R, \mathbf{t}) \in \mathrm{SE}(3)$ at the current simulation frame. Therefore, $O(m' \log m')$ of the FFT can be reduced to $O(m')$ of a cascade partial summation of (4.1.8) for a single translation index i. However, this does not yield a significant performance gain in practice, where the running time is governed by the notably slower pointwise multiplication step. Besides, parallel implementation of a cascade sum (especially on the GPU) for small m' does not necessarily outperform its FFT counterpart.

Figure 4.1.1 (a–d) illustrates the successive Fourier approximations for the 2D skeletal densities in Fig. 3.4.4 (b) with $\sigma = 0.5$. A grid size of $m = 512^2 = 262{,}144$ nodes is used, resulting in the same number of frequency domain amplitudes. In this case, the evolution of the score function is not substantial for $m' > 16^2 = 256$

[7] As detailed in Section 4.2.1, the geometric force and torque computations are carried out in a separate loop running on a separate thread than that of the haptic servo-loop running at 1 kHz, the two being coupled using a 'virtual coupling' described in Section 4.2.2. Hence the program is robust to discrepancies in the refresh rates (up to an order of magnitude). Nevertheless, the best user experience is achieved when the geometric force and torque values are also updated (at least) as frequently as 1 kHz.

(less than 0.1% of m). However, shapes with higher geometric detail require a larger number of modes to capture the smallest features. It is very important to note that the input model complexity—e.g., the number of triangles $n_{1,2}$ per mesh representation $\Delta_{n_{1,2}}(S_{1,2})$ of the solids $S_{1,2} \in \mathsf{S}$—is irrelevant here, and the suitable value of m' is determined from the desired fidelity with which the small geometric features are captured in the output energy field. This indifference to the syntactic representation complexity is a significant advantage of the presented method over most other collision detection and constraint management algorithms, whose running times depend on (and scale with) the representation size, resulting in a failure to handle large mesh sizes in real-time due to the high frame rate limitation.

4.2 Haptic Assembly System

I developed a simple graphics- and haptics-enabled software to test the geometric energy formulation given in Chapter 3 for mechanical assembly and protein docking test-cases (Fig. 4.2.1) using thew following hardware configurations:

- **Config.–L:** A Dell Precision T7500 workstation with an Intel® Xeon® E55520 CPU with 8 cores, clock rate of 2.67 GHz, and host memory of 12 GB. The system is equipped with dual NVIDIA® Quadro® NVS 420 graphics cards each with 16 CUDA cores (compute capability (CC) = 1.1) and device memory of 512 MB (used for graphics rendering for visualization).

- **Config.–H:** A Dell Precision T7600 workstation with an Intel® Xeon® E5-2687W CPU with 32 cores, clock rate of 3.10 GHz, and host memory of 64 GB. The system is equipped with two graphics cards: a NVIDIA® Quadro® K5000 GPU

FIGURE 4.2.1: A desktop graphics- and haptics-enabled system for mechanical assembly and protein docking at work. Two different haptic devices are used concurrently: Phantom® Omni® (left) and Virtuose6D™ 35-45 (right).

with 1,536 CUDA cores (CC = 3.0) and device memory of 4 GB (used for graphics rendering for visualization), and another NVIDIA® Tesla® K20c GPU with 2,496 CUDA cores (CC = 3.5) and device memory of 5 GB (used for preprocessing SDF and real-time energy computations).

The lower-end **Config.–L** is used for simpler rendering tasks that require minimal CPU or GPU power, while the higher-end **Config.–H** is used for more intensive computations, especially those that involve SDF precomputations and FFT convolutions on the GPU. The following two haptic devices are used:

- **Device–A:** A SensAble® Phantom® Omni® haptic device with 6 DOF input (single-point, 3D translation and 3D rotation) and 3 DOF output (single point, 3D force but no torque) is used. The device workspace contains a cubic volume

with dimensions 160 mm (width) × 120 mm (height) × 70 mm (depth); it has a nominal resolution of 0.055 mm, a maximum force capacity of 3.30 N, a maximum continuous (i.e., 24-hour) force capacity of 0.88 N.

- **Device–E:** A Haption Virtuose6D^{TM} 35-45 haptic device with 6 DOF input (single-point, 3D translation and 3D rotation) and 6 DOF output (single point, 3D force and 3D torque) is used. The device workspace contains a cubic volume with dimensions 450 mm (width) × 450 mm (height) × 450 mm (depth);[8] it has a nominal resolution of 0.020 mm, a maximum force capacity of 35.0 N, a maximum torque capacity of 3.1 Nm, a maximum continuous force capacity of 3.1 N, and a maximum continuous torque capacity of 1.0 N.

The small and affordable **Device–A** is used for initial tests and debugging purposes when there is a possibility of errors in the code that might result in excessive forces and torques, instability (e.g., 'buzzing'), and other destructive outcomes. The relatively expensive **Device–E** has a significantly larger workspace and stronger feedback, but is less forgiving when it comes to errors, hence is used only after the code has successfully passed extensive testing using **Device–A**.

4.2.1 Architecture

Figure 4.2.2 illustrates the architecture of the haptic assembly platform implemented in C++ to test the geometric energy formulation given in Chapter 3. The main application thread is forked into four threads on the CPU using Boost C++ libraries [355] to concurrently implement the following four asynchronous loops:

[8]The actual device workspace is significantly larger region with spherical boundaries contained in a cubic volume with dimensions 1,300 mm (width) × 1,080 mm (height) × 660 mm (depth).

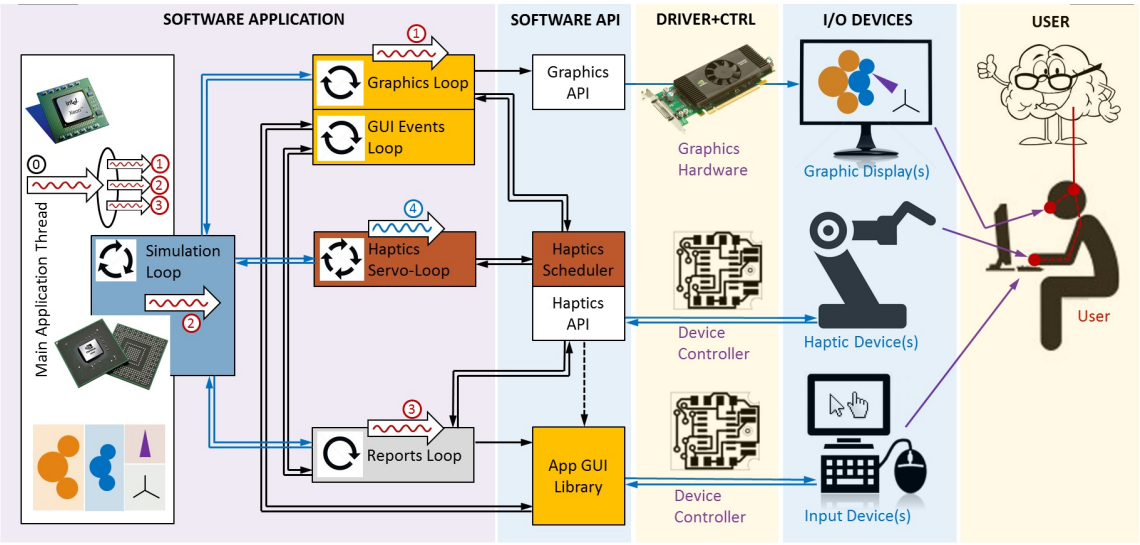


FIGURE 4.2.2: Architecture of the multithread/multiloop haptic assembly platform.

1. The 'graphics loop' runs at 30−60 Hz and is implemented using GLUT or Win32 API. At every frame, the geometric information such as the fixed B-rep data (e.g., triangular mesh coordinates), object frame transformation, camera frame transformation, and other rendering parameters (e.g., pertaining to lighting, shading, etc.) are sent into the OpenGL rendering pipeline for visualization. The graphical user interface (GUI) events such as mouse click, mouse motion, keyboard hit,and alike, which are used to perform basic graphics functions such as camera motions (e.g, pan, rotate, zoom, etc.) or rendering switches (e.g., wireframe on/off, shading on/off, etc.) are also implemented on the same thread.

2. The 'haptics loop' runs at 1 kHz and is implemented using OpenHaptics® API and VirtuoseAPI. If two (or more) devices are used concurrently, there must be as many haptic loops as there are devices running asynchronously. At every frame of each separate haptic loop, the positions and orientations of the haptic interface point(s) (HIP) are read from the device(s). Accordingly, the forces

and torques are computed directly from the relative kinematic configurations of the master and slave frames of the virtual coupling (detailed in Section 4.2.2) and are sent to the haptic device(s) for user feedback, and to the simulation thread (next item) for dynamic modeling.

3. The 'simulation loop' runs as fast as the computations allow (~ 100 Hz typically perceived acceptable). All geometric energy queries and physics-based computations are performed in this loop. At every frame, the simulation loop receives the most recently updated haptic forces and torques from the haptics loop, computes the geometric forces and torques (on the GPU) using the concepts developed in Chapter 3, adds the environmental (e.g., viscosity) effects, performs dynamic simulation by integrating Newton+Euler's equations of motion, and updates the configuration of the objects for the next frame.[9]

4. The 'report loop' runs at $1-10$ Hz and is responsible for measuring and reporting the desired performance data to the GUI for monitoring the assembly process.

The different threads share the kinematic variables—e.g., the 4×4 object frame transformation matrix (position and orientation), linear and angular velocity and acceleration vectors, etc.—as well as the dynamic variables—e.g., geometric, body, environment, and HIP force and torque vectors, etc.—for the moving rigid parts as well as constant parameters in a central data structure on the host memory. Each thread accesses the shared variables via standard mutual exclusion (mutex) techniques. To

[9]If the force and torque computations are intensive, the simulation loop itself can be divided into two concurrent loops; namely, a 'modeling loop' in which the geometric (and other) forces and torques are computed, and a 'dynamics loop' which is responsible only for integrating the equations of motion and obtaining the new configuration from the most recently updated force and torque values. This way, the dynamic simulation time-stepping does not have to suffer from delays imposed by the modeling computations and the integrations can be maintained at as high as $1-10$ kHz.

prevent the high frame rate loops (e.g., the haptic thread) to be delayed while waiting for the low frame rate loops (e.g., the graphics thread), care was taken not to lock a shared variable while doing computations on them by any thread. Instead, each thread keeps local copies of each variable, performs the time-consuming computations on those local copies, and makes necessary updates to the shared variable once the computations are finished using fast atomic read and/or write operations. To avoid inconsistencies, the privileges for updating certain variables are owned by certain threads only, while other threads are only allowed to access them as read-only variables. For instance,

- the HIP force and torque vectors are only updated by the haptic thread where their values are computed from the relative kinematic configurations of the master and slave frames of the virtual coupling (detailed in Section 4.2.2);
- the geometric force and torque vectors are only written by the simulation thread where their values are computed (by running kernels on the GPU); and
- the camera transformation is only updated by the graphics thread were it is updated in response to the mouse events on the GUI.

4.2.2 Virtual Coupling

In principle, there are at least two different methods by which the program can transfer the resultant forces and torques—including geometric, body, environment, and other forces and torques—to the haptic interface point (HIP) of the single-point haptic device with 6 DOF input and 3−6 DOF output, i.e., one that can be queried for a 4×4 transformation matrix (e.g., returned by OpenHaptics® API) or equiva-

lently a vector+quaternion 7−tuple (e.g., returned by VirtuoseAPI) for position and orientation of a frame at the HIP:

1. *Rigid Coupling*: at the instant when the designated button on the haptic device is pressed, the HIP frame at its instantaneous position and orientation is rigidly attached to (the geometric extension of) the object being grabbed. As long as the button is held down, the object is completely constrained to move with the HIP frame, i.e., undergo the same frame transformations. This is illustrated in Fig. 4.2.3 (a). This is only suitable when a single haptic device is present.

2. *Flexible Coupling*: at the instant when the designated button on the haptic device is pressed, the HIP frame at its instantaneous position and orientation is anchored on (the geometric extension of) the object being grabbed (i.e., the so-called 'slave' frame), while the HIP frame itself departs from it in the subsequent instants (i.e., the so-called 'master' frame). As long as the button is held down, the object—along with the slave frame rigidly attached to it—is still free to move around without any additional constraints. However, a spring-damper coupling (with both axial and torsional components) becomes effective between the master and slave frames, trying to make the two coincide. This is illustrated in Fig. 4.2.3 (b). This allows concurrent use of multiple haptic devices.

When rigid coupling is used, the position and orientation of the object being grabbed is dictated directly by the input from the haptic device, i.e., by the HIP frame transformations. The linear and angular velocity and acceleration vectors are thus obtained by differentiation (detailed in Section 4.2.3). Once the kinematic variables are computed, the resultant force and torque on the rigid body with configuration

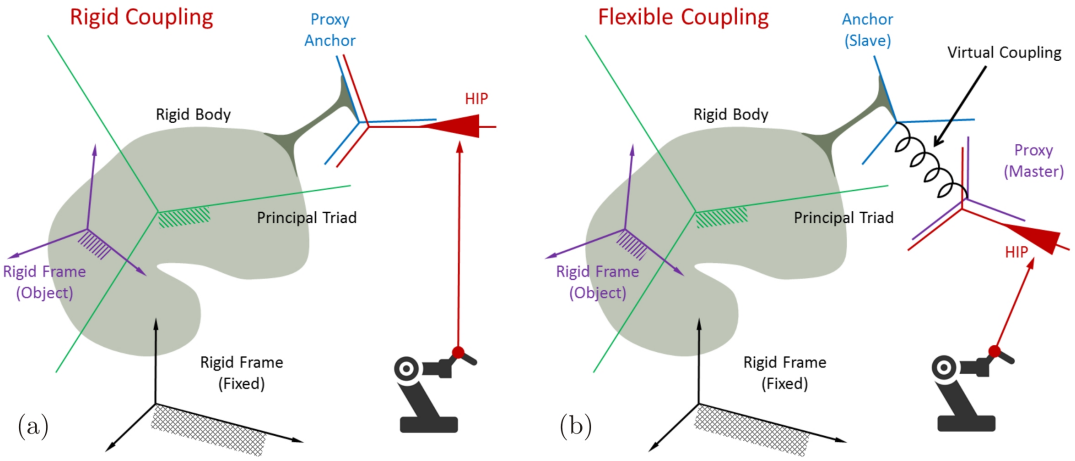


FIGURE 4.2.3: Virtual coupling techniques: (a) rigid coupling; and (b) flexible coupling.

$(R, \mathbf{t}) \in \mathrm{SE}(3)$ can be obtained using Newton+Euler's equations of motion [158]:

$$\mathbf{F} = \mathrm{M}\mathbf{a} \quad \text{and} \qquad \mathbf{T} = \mathbf{I}\boldsymbol{\alpha} + (\boldsymbol{\omega} \times \mathbf{I})\boldsymbol{\omega}, \tag{4.2.1}$$

where $\mathbf{F}, \mathbf{T} \in \mathbb{R}^3$ are the resultant force and torque on the body, respectively, M is the object's mass, $\mathbf{I}$ is the object's 3×3 inertia matrix, $\mathbf{a} = \frac{d^2\mathbf{t}}{dt^2} \in \mathbb{R}^3$ is the linear acceleration (of the center of mass), $\boldsymbol{\omega} \in \mathbb{R}^3$ is the angular velocity—i.e., the dual vector of the skew-symmetric matrix $\frac{dR}{dt}R^{\mathrm{T}}$—and $\boldsymbol{\alpha} \in \mathbb{R}^3$ is the angular acceleration—i.e., the dual vector of the skew-symmetric matrix $\frac{d^2R}{dt^2}R^{\mathrm{T}} + \frac{dR}{dt}\frac{dR^{\mathrm{T}}}{dt}$. Note that the second equation in (4.2.1) (i.e., Euler's equation in simplified form) is only valid if the angular velocity and acceleration are observed from the body-axis coordinate system centered at the center of mass and aligned with the principal axes of inertia, which makes $\mathbf{I}$ a diagonal matrix containing the 3 principal moments of inertia [158]. Once the resultant force and torque are known, the haptic force and torque that apply at

the HIP to the object are computed from a knowledge of all of the other effects as

$$\mathbf{F}_{\mathrm{H}} = \mathbf{F} - (\mathbf{F}_{\mathrm{G}} + \mathbf{F}_{\mathrm{B}} + \mathbf{F}_{\mathrm{E}} + \cdots), \tag{4.2.2}$$

$$\mathbf{T}_{\mathrm{H}} = \mathbf{T} - (\mathbf{T}_{\mathrm{G}} + \mathbf{T}_{\mathrm{B}} + \mathbf{T}_{\mathrm{E}} + \cdots), \tag{4.2.3}$$

where the geometric force and torque $\mathbf{F}_{\mathrm{G}}, \mathbf{T}_{\mathrm{G}} \in \mathbb{R}^3$ are computed from (3.7.2) and (3.7.3), respectively. Other effects may include body forces and torques $\mathbf{F}_{\mathrm{B}}, \mathbf{T}_{\mathrm{B}} \in \mathbb{R}^3$ (e.g, gravity), environment forces and torques $\mathbf{F}_{\mathrm{E}}, \mathbf{T}_{\mathrm{E}} \in \mathbb{R}^3$ (e.g., viscosity), etc. The haptic force and torque (in the inverse direction) are then returned to the haptic device as force feedback to the user. This is an *inverse* dynamics approach.

On the other hand, flexible coupling takes a *direct* dynamics approach by integrating (instead of differentiating) the kinematic variables. The master frame's motion is once again dictated by the HIP frame's motion, while the slave frame's motion is determined by the dynamic simulation on the object being grabbed. The kinematic variables of the two frames are maintained and updated at different rates in different asynchronous loops; namely, the master's configuration $(R_{\mathrm{M}}, \mathbf{t}_{\mathrm{M}}) \in \mathrm{SE}(3)$ is queried from the haptic device in the haptics loop at 1kHz while the slave's configuration $(R_{\mathrm{S}}, \mathbf{t}_{\mathrm{S}}) \in \mathrm{SE}(3)$ is updated in the simulation (or dynamics) loop as soon as the integration proceeds one time-step. Assuming a (linear and torsional) spring-damper virtual coupling between the two frames, the haptic force and torque that apply at the slave frame's anchor point to the object are obtained as

$$\mathbf{F}_{\mathrm{H}} = K_{\mathrm{F}}(\mathbf{t}_{\mathrm{S}} - \mathbf{t}_{\mathrm{M}}) + B_{\mathrm{F}}(\mathbf{v}_{\mathrm{S}} - \mathbf{v}_{\mathrm{M}}), \tag{4.2.4}$$

$$\mathbf{T}_{\mathrm{H}} = K_{\mathrm{T}}\boldsymbol{\theta}(R_{\mathrm{M}}^{\mathrm{T}} R_{\mathrm{S}}) + B_{\mathrm{T}}(\boldsymbol{\omega}_{\mathrm{S}} - \boldsymbol{\omega}_{\mathrm{M}}), \tag{4.2.5}$$

where $K_{\mathrm{F}}, K_{\mathrm{T}} \in \mathbb{R}^{+}$ are the linear and torsional spring (i.e., stiffness) constants while $B_{\mathrm{F}}, B_{\mathrm{T}} \in \mathbb{R}^{+}$ are the linear and torsional damper (i.e., viscosity) constants, respectively. The vector $\boldsymbol{\theta}(R) = \theta\mathbf{u}$ is the dual vector of the skew-symmetric matrix

$$\Omega := \log R = \frac{\theta}{2\sin\theta}\big(R - R^{\mathrm{T}}\big), \tag{4.2.6}$$

which can be interpretted as the axis-angle representation $(\mathbf{u}, \theta)$ of the rotation matrix $R \in \mathrm{SO}(3)$, i.e., $\mathbf{u} \in \mathbb{R}^3$ is the unit eigenvector that represents the rotation axis according to the right-hand rule, and $\theta \in [0, \pi)$ represents the rotation angle along the shorter geodesic, obtained as

$$\theta = \cos^{-1}\left[\frac{1}{2}\mathsf{tr}(R) - 1\right], \tag{4.2.7}$$

where $\mathsf{tr}(R)$ is the trace of the rotation matrix. The haptic force and torque are then added with other effects to obtain the resultant force and torque on the body:

$$\mathbf{F} = \mathbf{F}_{\mathrm{H}} + (\mathbf{F}_{\mathrm{G}} + \mathbf{F}_{\mathrm{B}} + \mathbf{F}_{\mathrm{E}} + \cdots), \tag{4.2.8}$$

$$\mathbf{T} = \mathbf{T}_{\mathrm{H}} + (\mathbf{T}_{\mathrm{G}} + \mathbf{T}_{\mathrm{B}} + \mathbf{T}_{\mathrm{E}} + \cdots), \tag{4.2.9}$$

which are then substituted in the Newton+Euler's equations of motion [158] to solve for the linear acceleration $\mathbf{a} \in \mathbb{R}^3$ and angular acceleration $\boldsymbol{\alpha} \in \mathbb{R}^3$ as

$$\mathbf{a} = \mathbf{F}/\mathrm{M} \quad \text{and} \qquad \boldsymbol{\alpha} = \mathbf{I}^{-1}\big[\mathbf{T} - (\boldsymbol{\omega} \times \mathbf{I})\boldsymbol{\omega}\big]. \tag{4.2.10}$$

The linear and angular acceleration vectors are then integrated twice (detailed in Section 4.2.3) to obtain the variations in the object's linear and angular velocities

and the changes in the object's position and orientation, respectively.

There are several benefits in using flexible coupling over using rigid coupling. First, it provides a simple means to separate the haptics loop, where the HIP force and torque are computed using simple and fast formulae in (4.2.2) and (4.2.3), respectively, from the simulation (or dynamics) loop that runs asynchronously with a different frame rate. The master and slave frames are thus updated at different rates independently without one delaying the other. Second, the existence of a virtual spring-damper coupling between the master and slave frames provides additional noise absorption from the HIP (i.e., master) frame to the object (i.e., slave) frame. Last but not least, the fact that the object is not kinematically constrained to the HIP frame makes it possible to use multiple haptic devices concurrently, each maintaining their own HIP and master-slave coupling at a separate loop running on a separate thread. When a single object is grabbed by multiple devices, each device's HIP is anchored at a different point where it drags the object through the application of a force/torque pair per device computed from (4.2.2) and (4.2.3).

4.2.3 Numerical Tools

This section overviews the finite difference method (FDM) for numerical differentiation and integration needed for direct and inverse dynamic simualtion presented in Section 4.2.2.

Let $(R_-, \mathbf{t}_-), (R_0, \mathbf{t}_0), (R_+, \mathbf{t}_+) \in \mathrm{SE}(3)$ be 3 different configurations of a virtual object in the scene at consecutive times (e.g., simulation frames) $t_-, t_0, t_+ \in \mathbb{R}$—assuming a small finite difference of $\delta := (t_+ - t_0) \approx (t_0 - t_-)$—respectively, the linear velocity $\mathbf{v} = \frac{d\mathbf{t}}{dt} \in \mathbb{R}^3$ and angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$—i.e., the dual vector of

the skew-symmetric matrix $\frac{dR}{dt}R^{\mathrm{T}}$—at t_0 can be approximated using 2-point forward-, backward-, or central-difference (FD/BD/CD) FDM as

$$\text{FD-FDM:} \qquad \mathbf{v}_0 = \Big(\frac{\mathbf{t}_+ - \mathbf{t}_0}{t_+ - t_0}\Big) + O(\delta), \qquad \boldsymbol{\omega}_0 = \Big(\frac{R_+ - R_0}{t_+ - t_0}\Big)R_0^{\mathrm{T}} + O(\delta), \tag{4.2.11}$$

$$\text{BD-FDM:} \qquad \mathbf{v}_0 = \Big(\frac{\mathbf{t}_0 - \mathbf{t}_-}{t_0 - t_-}\Big) + O(\delta), \qquad \boldsymbol{\omega}_0 = \Big(\frac{R_0 - R_-}{t_0 - t_-}\Big)R_0^{\mathrm{T}}, + O(\delta), \tag{4.2.12}$$

$$\text{CD-FDM:} \qquad \mathbf{v}_0 = \Big(\frac{\mathbf{t}_+ - \mathbf{t}_-}{t_+ - t_-}\Big) + O(\delta^2), \quad \boldsymbol{\omega}_0 = \Big(\frac{R_+ - R_-}{t_+ - t_-}\Big)R_0^{\mathrm{T}} + O(\delta^2), \tag{4.2.13}$$

The above formulae can be viewed as fitting straight lines to data points in order to approximate the 1st derivative. More accurate approximations are possible, e.g., using 3-point FDM approximations which are equivalent to fitting parabolas to the data points, e.g.,

$$\text{FD-FDM:} \qquad \mathbf{v}_0 = \Big(\frac{-\mathbf{t}_{++} + 4\mathbf{t}_+ - 3\mathbf{t}_0}{t_{++} - t_0}\Big) + O(\delta^2), \qquad \boldsymbol{\omega}_0 = \cdots, \tag{4.2.14}$$

$$\text{BD-FDM:} \qquad \mathbf{v}_0 = \Big(\frac{+3\mathbf{t}_0 - 4\mathbf{t}_- + \mathbf{t}_{--}}{t_0 - t_{--}}\Big) + O(\delta^2), \qquad \boldsymbol{\omega}_0 = \cdots, \tag{4.2.15}$$

and so on by fitting even higher-order polynomials to larger number of consecutive points in time. Similar equations are obtained for the linear acceleration $\mathbf{a} = \frac{d^2\mathbf{t}}{dt^2} \in \mathbb{R}^3$ and angular acceleration $\boldsymbol{\alpha} \in \mathbb{R}^3$—i.e., the dual vector of the skew-symmetric matrix $\frac{d^2R}{dt^2}R^{\mathrm{T}} + \frac{dR}{dt}\frac{dR^{\mathrm{T}}}{dt}$—at t_0 using 3-point FDM approximations as

$$\text{FD-FDM:} \qquad \mathbf{a}_0 = \Big(\frac{\mathbf{t}_{++} - 2\mathbf{t}_+ + \mathbf{t}_0}{(t_+ - t_0)^2}\Big) + O(\delta), \qquad \boldsymbol{\alpha}_0 = \cdots, \tag{4.2.16}$$

$$\text{BD-FDM:} \qquad \mathbf{a}_0 = \Big(\frac{\mathbf{t}_0 - 2\mathbf{t}_- + \mathbf{t}_{--}}{(t_+ - t_0)^2}\Big) + O(\delta), \qquad \boldsymbol{\alpha}_0 = \cdots, \tag{4.2.17}$$

$$\text{CD-FDM:} \qquad \mathbf{a}_0 = \Big(\frac{\mathbf{t}_+ - 2\mathbf{t}_0 + \mathbf{t}_-}{(t_+ - t_0)^2}\Big) + O(\delta^2), \qquad \boldsymbol{\alpha}_0 = \cdots, \tag{4.2.18}$$

The above formulae can be viewed as fitting parabolas to the data points to approximate the 2nd derivative, which can also be made more accurate by fitting higher-order polynomials to larger number of points.

For real-time simulations in the VE, one typically has access to the history of the kinematic data up to the current moment—i.e., data is available at all $t \leq t_0$ but not at $t > t_0$. Hence the BD-FDM and FD-FDM formulae appear to be the natural choice for differentiation and integration, respectively, in our VE application.[10]

An important caveat is that the kinematic variables queried from a haptic device (e.g., the position and orientation of the HIP frame) are inherently noisy, partly due to the device encoder errors and partly due to the authentic vibrations of the user's hand (i.e., hand 'jerking'). The noise is significantly amplified every time these variables are differentiated using 2-, 3-, (or generally few-) point FDM approximations. The haptic device APIs typically come with functions to directly query the linear and angular velocity and acceleration information using built-in noise filtering algorithms. However, my experience has shown that these filters, despite alleviating the noise problem, are not sufficient especially when it comes to the second differentiation. To resolve this problem, I use a larger number of points in the recent history of the HIP frame configurations—e.g., the last 100 points corresponding to a 0.1-second time interval queries in the 1 kHz haptic servo-loop—to which quadratic equations are fit (using the least-squares method) to the translation and rotation components whose coefficients are taken to approximate 'filtered' linear and angular velocity and acceleration vectors.

[10]There are many different ways in which the FDM approximations to differentiation can be solved to obtain numerical formulae for integration. A subset of these methods correspond to the Runge-Kutta (RK) family of numerical integration methods which can be found in any elementary numerical analysis textbook [99].

Chapter 5

Results and Discussion

5.1 Peg-in-Hole Examples

I first demonstrate the effectiveness of the proposed approach for simple classical peg-in-hole examples in Fig. 2.1.3, repeated here for convenience in Fig. 5.1.1.

Figure 5.1.2 (a) shows the three examples made of cylindrical pegs of circular, rectangular, and combined cross-sections, which were tested in assembly against their complementary holes. The geometric fit in all three cases is exact (i.e., zero clearance).

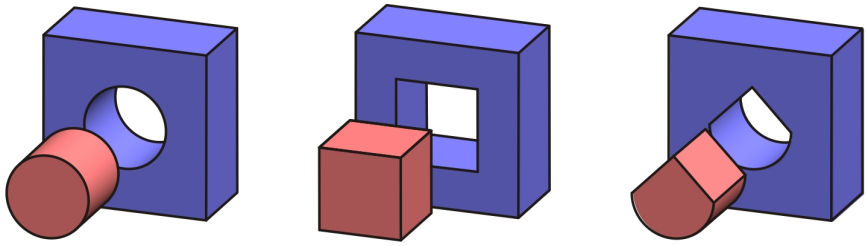
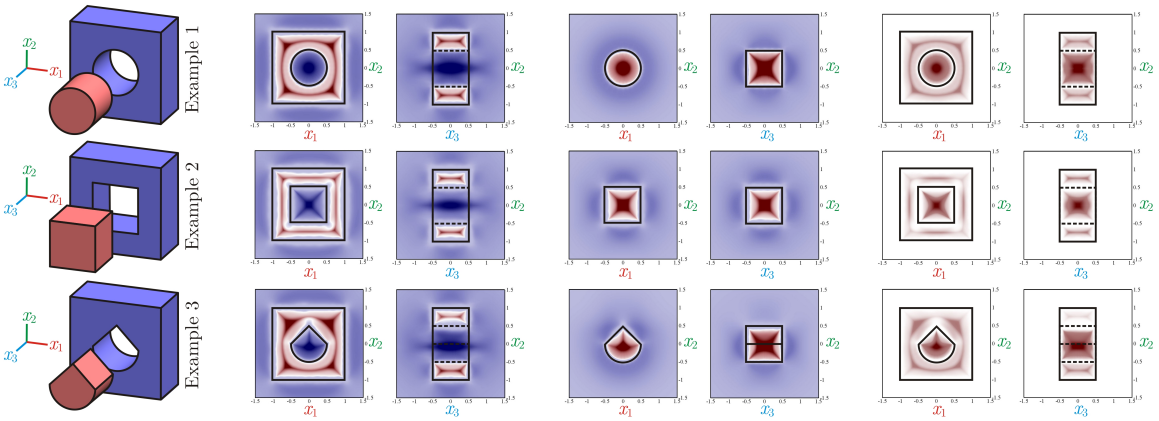


FIGURE 5.1.1: Examples of peg-in-hole assembly problems.



(a) Peg-in-hole examples (b) Hole SDF (front & left views) (c) Peg SDF (front & left views) (d) SDF overlap (front & left views)

FIGURE 5.1.2: Three peg-in-hole assemblies (a), their SDFs (imaginary-parts) (b, c), and their spatial overlap (real-part) (d).

5.1.1 Skeletal Overlaps

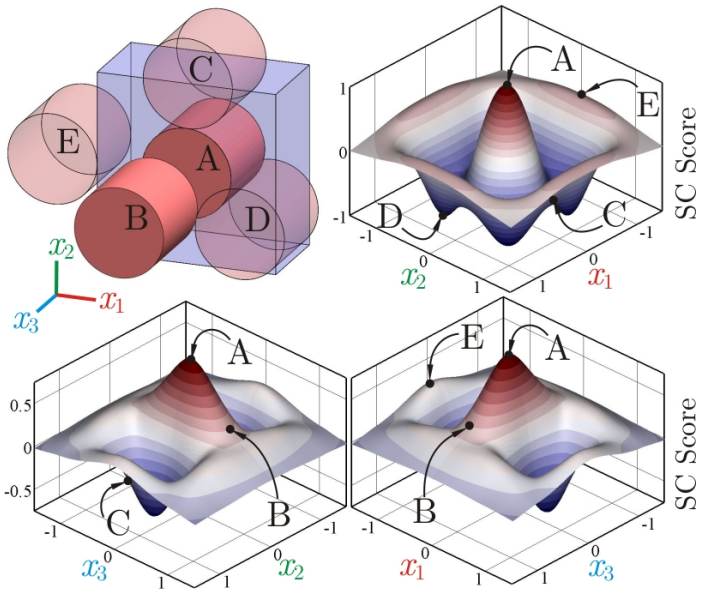
In Fig. 5.1.2 (b, c), the individual SDF maps of the parts are plotted only for their imaginary-parts. As expected, each part has the highest positive-imaginary SDF values at the proximity of the high-prong internal MA branches (the red spots) and the highest negative-imaginary SDF values at the proximity of the high-prong external MA branches (the blue spots). The latter are weaker in intensity due to the choice of $\mathfrak{p} = \lambda_2/\lambda_1 = 3.0$ in (3.4.12). It can be verified that the complementary features have similar SDF distributions on one part's interior and the other part's exterior, resulting in the 'hot spots' on the geometric energy density map given in Fig. 5.1.2 (d), which dominate the 'dark spots'.

In the simplest case of Example 1, as the user brings the peg closer to the opening of the hole to perform the assembly task, the geometric force-field attracts the peg into the hole and tries to align the high-density SDF regions, i.e., enforce co-axiallity of the two cylindrical faces. The circular symmetry of the cross-section results in a circular symmetry of the SDF, hence the force-field imposes no orientation pref-

erence around the axis of the hole, resulting in a partially constrained motion that resembles a cylindrical joint. However, in the case of Example 2, the cross-shaped skeletal form creates an additional orientation preference; hence the force-field tries to align the four corners of the two complementary objects. This results in a partially constrained motion that resembles a prismatic joint. In the case of Example 3, the shape descriptors appear as a combination of the two cases, with part of the geometry being indifferent to rotations around the hole, while another feature contributes energy terms to align the sharp corner.

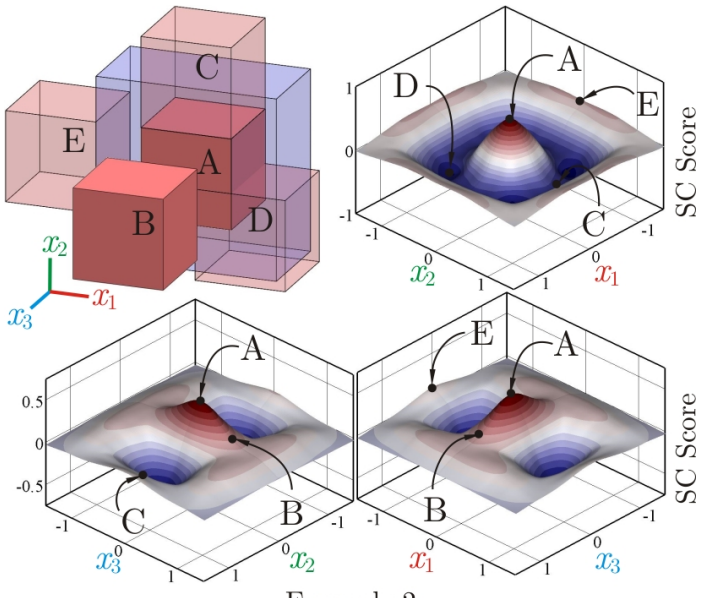
It is clear from the above discussion that our SDF descriptors serve as generic replacements for the abstract virtual fixtures [343, 344]—e.g., the cylindrical axes in Example 1 and diagonal planes in Example 2—resembling the explicit guiding axes and planes previously implemented in [389, 390] for haptic assembly guidance, which were limited to simple geometric constructs. The skeletal branches formed automatically in our development serve as abstractions of the functional surfaces (e.g., introduced in [200]) for arbitrarily complex shapes, in contrast to the ad hoc characterizations (e.g., in [198–202, 322]). As illustrated by Example 3, combinations of guiding mechanisms naturally appear with no theoretical limitation on the complexity of the assembly features. Furthermore, the force-field incorporates both collision response—as a repulsive force in the case of interpenetration—and assembly assistance—as an attractive force in the vicinity of the hole—in a single model that enforces geometric constraints. Hence the hybrid approach based on two separate phases for free motion and precise insertion [402] is integrated into a single model, eliminating the need to switch between the two using 'blending' algorithms.

Figures 5.1.3 through 5.1.5 plot the shape complementarity score variations (only the real-parts) for the 3 example pairs, respectively, due to the translational motion



Example 1

FIGURE 5.1.3: The shape complementarity score variations versus biaxial relative translation of the peg with respect to the hole for Example 1.



Example 2

FIGURE 5.1.4: The shape complementarity score variations versus biaxial relative translation of the peg with respect to the hole for Example 2.

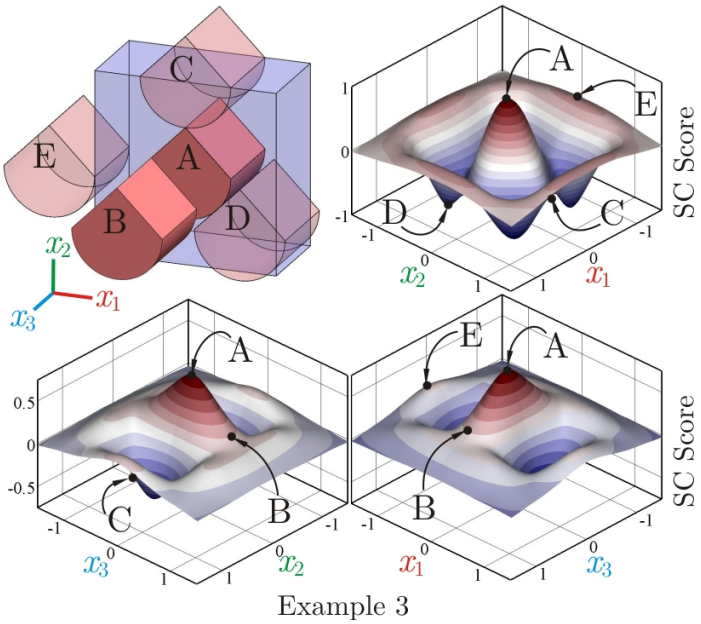


FIGURE 5.1.5: The shape complementarity score variations versus biaxial relative translation of the peg with respect to the hole for Example 3.

of the pegs relative to the holes along the 3 Cartesian axes. To enable visual illustration, the motion in each plot is restricted to a plane—i.e., changing only 2 out of 3 position coordinates at a time, of the peg with respect to the hole. It is clear that the shape complementarity score is maximal at configuration A—i.e., the zero translation, where the best fit occurs according to visual judgment—as expected from the definition. Other configurations are also sketched on the score profile, such as axis-aligned removal of the peg at B resulting in a decay of score from A to B along the x_3-axis, collision at C and D, and contact (but little shape complementarity) at E.

Figure 5.1.6 shows the corresponding geometric energy variations due to the same translational motions, this time moving the peg along one Cartesian axis at a time. These correspond to sections through the 2D plots in Fig. 5.1.3 through 5.1.5, except with a signed coefficient due to the definition in (3.7.1). Figure 5.1.7 shows a similar set of geometric energy variation diagrams, plotted versus the rotational motions

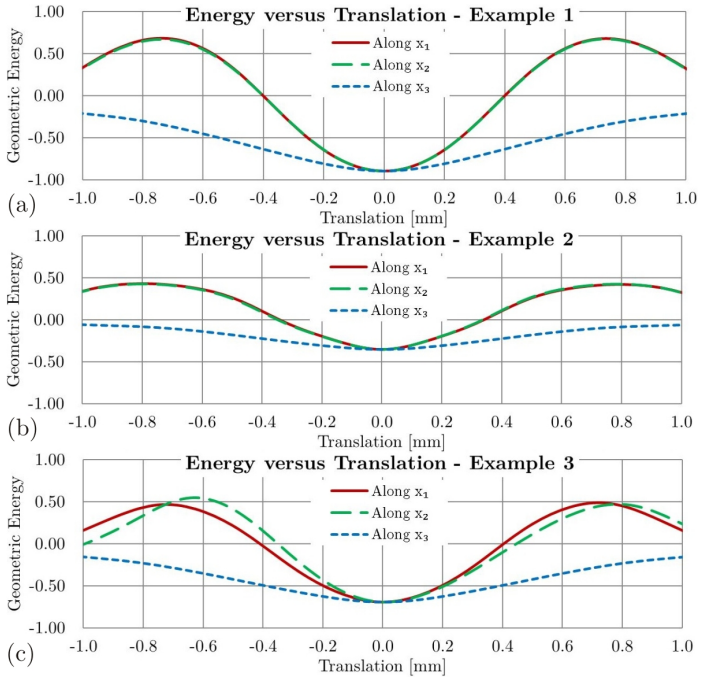


FIGURE 5.1.6: The geometric energy variations versus uniaxial relative translational motion of the peg with respect to the hole.

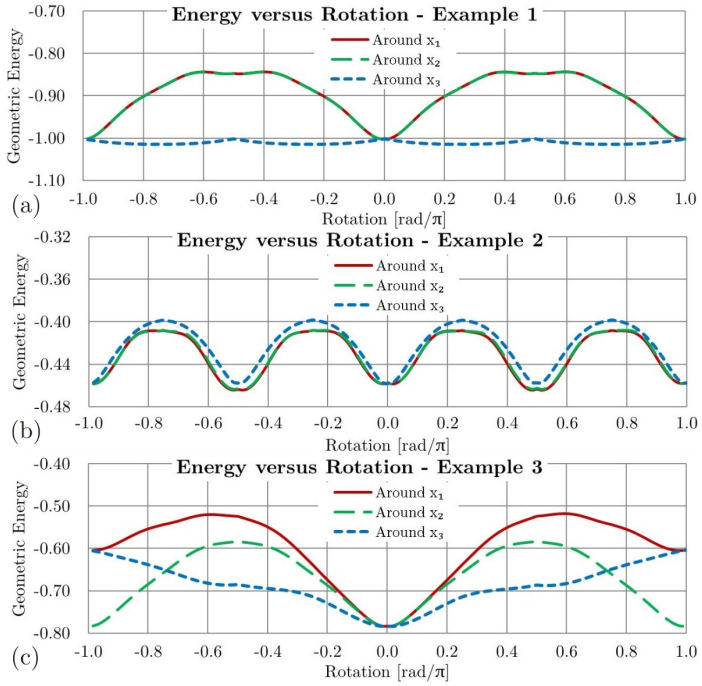


FIGURE 5.1.7: The geometric energy variations versus uniaxial relative rotational motion of the peg with respect to the hole.

around one Cartesian axis at a time. For both translational and rotational motions, there is an evident equivalence between x_1- and x_2-axes in Examples 1 and 2, as expected from the symmetrical shapes, which is not the case for Example 3 due to its different geometry. One can also notice the indifference of the circular cross-section to rotations around the axis of the hole in Example 1, and four equivalent configurations for the cubic peg with 90° phase difference in Example 2.

It is interesting to note that the size of the convex region of the geometric energy profile around the local minimum (characterizing the equilibrium point) can be conceptualized as the *diameter* of the geometric constraints, i.e., the degree of proximity necessary for the constraint to become activated for insertion. The second-order differential properties of the energy profile are directly related to the practical *stiffness* of the constraint enforcement in the VE. For instance, a Taylor series expansion of the energy function over the translational space T(3) (i.e., for fixed rotation) around the stable equilibrium configuration $\mathbf{t}_0 \in \mathrm{T}(3)$ has the form

$$E_{\mathrm{G}} = E_{\mathrm{G,min}} + (\mathbf{t} - \mathbf{t}_0) \cdot H(\mathbf{t}_0)(\mathbf{t} - \mathbf{t}_0) + O(\|\mathbf{t} - \mathbf{t}_0\|_2^3), \tag{5.1.1}$$

noting that $dE_{\mathrm{G}}/d\mathbf{t}(\mathbf{t}_0) = \mathbf{0}$ where the Hessian matrix $[H(\mathbf{t}_0)]_{3\times 3}$ carries the stiffness elements—i.e., tensile/compressive resistance in the diagonal elements, and shear resistance in the off-diagonal elements.[1] A similar expansion is possible over the tangent space $\mathfrak{so}(3)$ to SO(3) to obtain the rotational stiffness matrix. Both the diameter and stiffness can be adjusted by a proper setting of the thickness factor $\sigma > 0$ and coefficients $\lambda_{1,2}$ in (3.4.10).

[1]Note than no constraint can be rigidly and strictly satisfied due to the electromechanical restrictions at the haptic device level. Even the DOF-limiting equality constraints are typically enforced by rendering resistance forces using a spring-damper model that penalizes the violation, whose stiffness is upperbounded due to the servo-loop rate of 1 kHz [316].

5.1.2 Haptic Experiments

Finally, I report on a few pilot experiments to *feel* the applicability of the technique in real haptic-assisted assembly applications. Two simple experiments are carried out on **Config.–L** using **Device–A:** SensAble® Phantom® Omni®, namely:

1. *Collision Test*: the test is conducted by simply pushing the peg against the walls of the hole in random directions, in an attempt to disturb it from the proper fit configuration and penetrate into the walls of the object with the hole. The user tries to do this with complete control and steadiness, approximately once every two seconds.

2. *Snap Test*: the user carelessly moves the peg around the entrance of the hole, with random and uncontrolled (but gentle) impacts with the end-effector, positioning the peg in proximity of the proper fit configuration approximately once every second. The force field is expected to react immediately and snap the peg into the proper position.

The tests are carried out only in 3 translational DOF with force feedback. The force magnitude is plotted versus time in Fig. 5.1.8 for the first experiment. The performance is satisfactory, with accurate geometric alignment up to observable precision, smooth and continuous repulsive force feedback resisting penetration in all directions, and a smaller attractive force resisting the peg leaving the hole along the axis. The results of the second experiment are plotted in Fig. 5.1.9. In this case, the response was effective in snapping the object into position with very rare occasions of undesirable vibration or 'buzzing'. The haptic servo-loop rate was maintained within the acceptable range of 1.0 ± 0.3 kHz at all times.

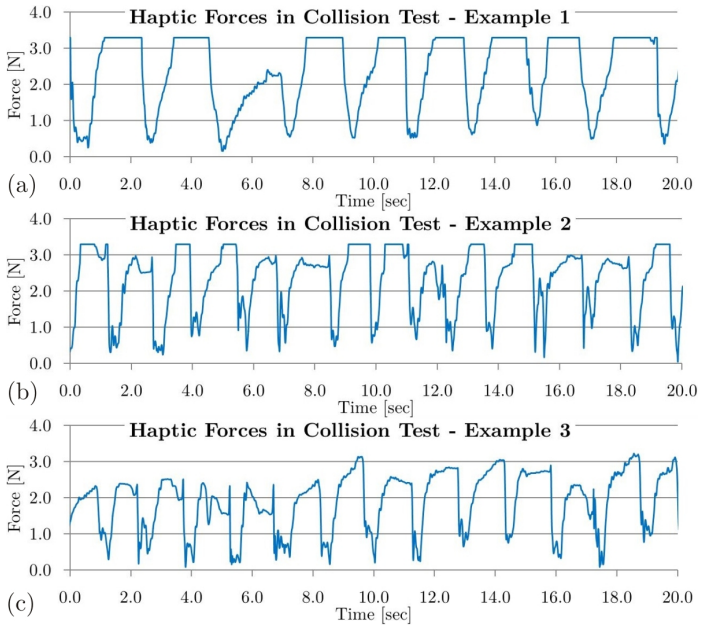


FIGURE 5.1.8: Haptic force feedback versus time for the collision test applied to the peg-in-hole examples in Fig. 5.1.2 using **Device–A:** SensAble® Phantom® Omni®.

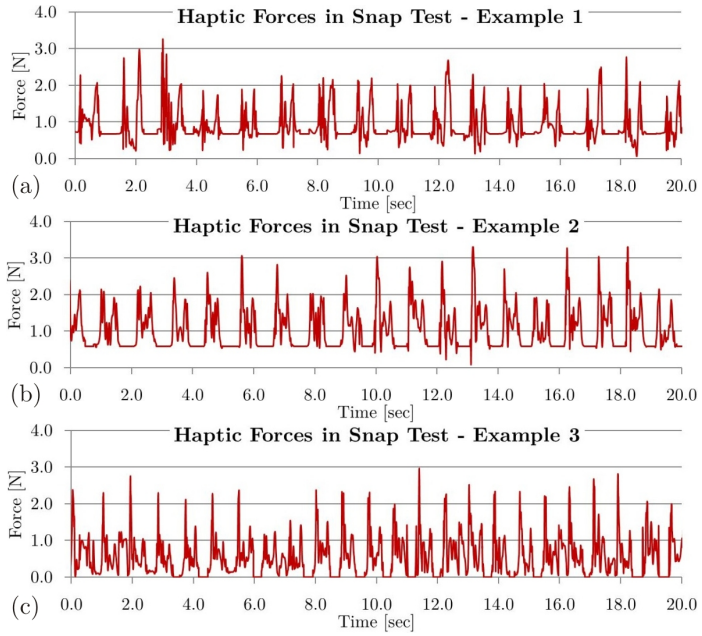


FIGURE 5.1.9: Haptic force feedback versus time for the snap test applied to the peg-in-hole examples in Fig. 5.1.2 using **Device–A:** SensAble® Phantom® Omni®.

The experimentation with both translational and rotational motions with force and torque feedback requires a haptic device with 6 DOF input and 6 DOF output (e.g., **Device–E:** Haption Virtuose6D^{TM} 35-45). The experience is quite different with **Device–E** even when the torque feedback is disabled, simply because of its capability to render forces of more than 10× larger magnitude within a significantly larger workspace compared to those of **Device–A**, making it harder to control the pase of assembly to repeat the two periodic data collections reported for **Device–A** in Figs. 5.1.8 and 5.1.9 for **Device–E** using a similar desktop setup.[2] Nonetheless, the results reported here are sufficient to demonstrate the effectiveness of the developed computational framework. A more robust implementation of the method and its further adjustments in order to cater to a wider range of device specifications and capabilities is, by itself, another demanding (and important) project that I choose to postpone to future studies.

5.2 A Realistic Example

The examples in Section 5.1 provided important insight into the properties of SDF and its effectiveness as an automated generalization of virtual fixtures for arbitrary geometry. However, the numerically intensive cascade computation of the integral in (4.1.2) requires substantial preprocessing time. Here I use the FFT convolution technique, using optimized CPU- and GPU-accelerated implementations. To demonstrate the practicality of the method, I test it on the pair of 3D assembly parts shown in Fig. 5.2.1. The solids in this example are made of semialgebraic r-sets with only

[2] I speculate that although **Device–A** is easier (and safer) to work with in a desktop VE (e.g., sitting in front of a monitor), the latter can provide more effective control and more efficient assembly when used in a more immersive VE (e.g., standing up in front of large stereoscopic projection screens).

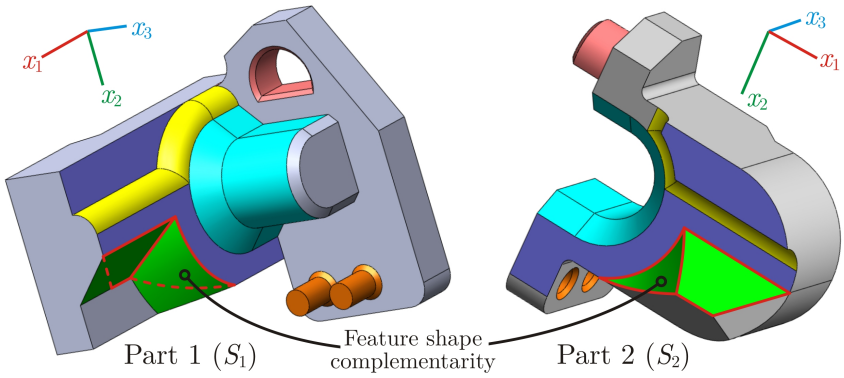


FIGURE 5.2.1: A non-trivial, zero-clearance assembly pair.

planar, cylindrical, spherical, and toroidal surface patches, which forms a small subset of the general semialgebraic class—i.e., solids bounded by polynomial surfaces of arbitrary degrees—and even more general semianalytic class covered by our formulation. Nevertheless, an automatic identification of the correspondence between the mating features (depicted with different colors) is not trivial from an algorithmic perspective—e.g., recognition and matching of the partially complementary features connected by a curve. Furthermore, there are 3 pairs of pegs and holes with zero clearance, making this example sufficiently challenging.

5.2.1 Skeletal Overlaps

The running times for the offline steps made of precomputing the part SDFs in the physical domain and their forward FFT into the frequency domain are plotted in Figs. 5.2.2 and 5.2.3.

The performance of the CPU- and GPU-parallel SDF computation for parts in Fig. 5.2.1 using **Config.–H** is presented in Fig. 5.2.2 (a, b) for different mesh and sample sizes to demonstrate the parallel computability and scalability of the SDF

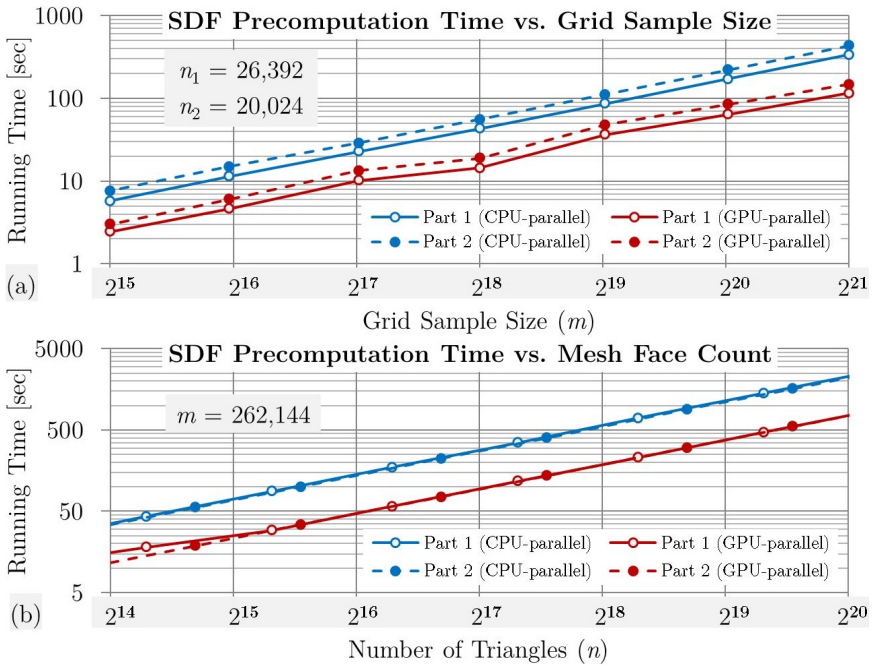


FIGURE 5.2.2: CPU vs. GPU performances for SDF computation.

computation process. The NVIDIA® Tesla® K20c GPU (2,496 CUDA cores, 5GB device memory) outperforms the Intel® Xeon® E5-2687W CPU (32 cores, 3.10 GHz clock-rate, 64GB host memory) by average speed-up factors of $2.5{-}3.0\times$.[3] For this particular example, a sample size of $m = 128^3 = 2{,}097{,}152$ and $n_{1,2} \approx 2 \times 10^5$ is adequate to capture the geometric details within the SDF field, which takes about 2–3 minutes per part to precompute the SDF offline. The resulting SDFs of the parts in Fig. 5.2.1 (using $\sigma := 0.5$ and $\mathfrak{p} = \lambda_2/\lambda_1 := 3$) are plotted for their imaginary parts in Fig. 5.2.5 (c). The correspondence between high density regions—e.g., along the axes of cylindrical features or along the bisectors of corners of the boundary—which are analytic generalizations of virtual fixtures, is apparent.

[3]This is less than expected, because the GPU implementation suffers from extra overhead due to data transfer between host and device memories, and is suboptimal in performing conditional instructions. Much better speed-ups up to two orders of magnitude are obtained in the convolution step, as depicted next.

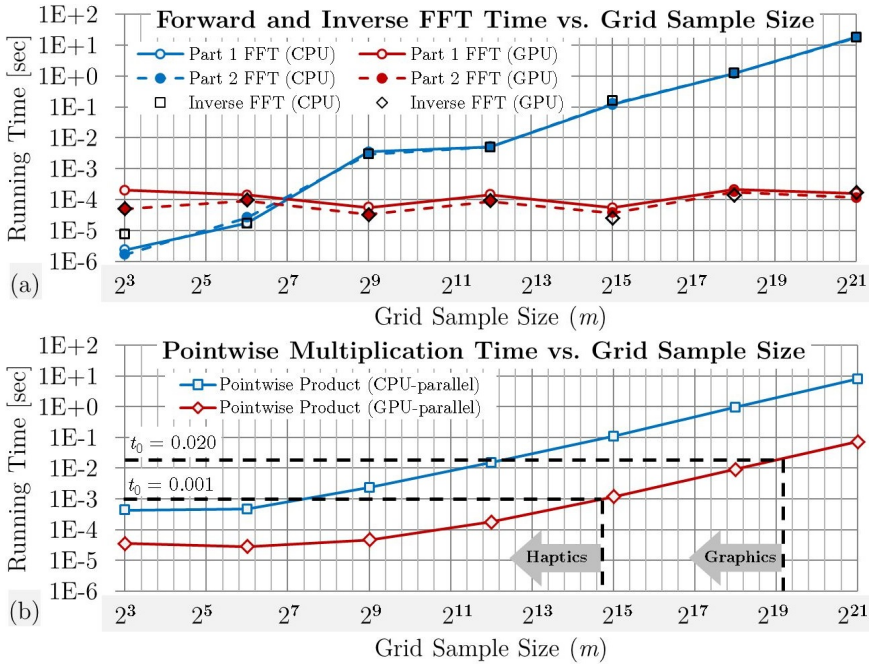


FIGURE 5.2.3: CPU vs. GPU performances for FFT convolution.

Figure 5.2.3 (a) presents the performance of the forward FFTs versus the sample size to map the part SDFs into the frequency domain on the CPU (using FFTW [133, 134]) and the GPU (using cuFFT(W)).[4] It is important to note that the forward FFT of part SDFs is also an offline preprocessing step, which takes negligible time compared to the previous steps (typically in the sub-millisecond range).

5.2.2 FFT Performance

The combination of Fourier amplitude product and inverse FFT cumulatively produces the geometric energy response for all relevant transformational configurations, which can take up to 0.1 second for large sample sizes, as depicted in Fig. 5.2.3 (b).

[4]Unfortunately, FFTW does not provide a CPU-parallel implementation of FFT, hence the CPU results reported in Fig. 5.2.3 (a) are sequential, unlike the other three plots in Figs. 5.2.2 and 5.2.3 in which the CPU results are in parallel, enabling a relatively fairer comparison with the GPU results (also in parallel, inherently, in all four plots).

However, during haptic assembly one ideally has less than 1 millisecond, but one also needs to evaluate only a single configuration at any instant of time during the virtual assembly; namely, the one corresponding to the instantaneous pose of the objects.

It is not possible in practice to precompute the energy field for a 6D grid of all possible motions. However, for most assembly scenarios the motion during the insertion phase is constrained to 1 or 2 DOF. For example, if the rotational space is limited to a finite number of permissible relative orientations, the inverse FFT for 3D translational motion can be precomputed and stored for each orientation, and queried rapidly during motion. This approach allows for computing the guidance force feedback to full accuracy, but goes against the philosophy of avoiding the multi-phase approach and manual specifications, from which we set off to pursue this method.

The major benefit of working in the frequency domain is the systematic means it provides to trade off the accuracy of physical domain representation with computation time. This provides a chance for real-time computation of translational convolution for arbitrary rotations given at any instant of the assembly simulation. If t_0 is the amount of dedicated computation time available at each frame—e.g., ideally $t_0 \leq 0.020$ seconds for 50 Hz graphic rendering, and $t_0 \leq 0.001$ seconds for 1 kHz haptic feedback—one can always choose the maximal sample size $m_0 \ll m$ whose processing time is approximately t_0. Hence real-time computing is contingent upon keeping only $m' \leq m_0$ dominant modes of the SDFs in the frequency domain to approximate the convolution function (see Fig. 4.1.1).

Figure 5.2.3 presents the running times of the convolution step, composed of both CPU- and GPU-parallel pointwise multiplication of the Fourier transformed SDFs (plotted in panel (b)), followed by an inverse FFT to obtain the geometric energy field in the physical domain (plotted in panel (a)). Here the GPU-acceleration of both steps

makes a crucial difference, as the inverse FFT time on the GPU stays significantly below the 1 millisecond threshold even for sample sizes as large as $m \sim 10^6 - 10^7$. The results clearly indicate that *the pointwise multiplication step is the bottleneck*, whose running time determines the upperbound m_0 on the size of truncated Fourier expansions. For this step, CPU parallelization speeds up the multiplication process by about $25\times$ over the sequential implementation, and as depicted in Fig. 5.2.3 the GPU implementation enhances it by an additional factor of about $100\times$—hence an overall speed-up as much as $2,500\times$. The difference grows with the sample size, which indicates the scalability of the process.[5]

An interesting subject to explore for future research is the possibility of using octree sampling (instead of using a uniform grid) and applying nonuinform FFT algorithms (NFFT) [323] (also available on the GPU [242]). The NFFT is less efficient than FFT, but octree sampling allows for using smaller sample sizes more densely populated around high-density regions of the SDF, in turn allowing a sharper Gaussian component in the $\phi-$kernel defined in (3.4.10) of Section 3.4.4 (i.e., smaller thickness parameter σ). This also allows for denser population of Fourier amplitudes around the dominant frequencies which facilitates more accurate frequency domain interpolation to sample $R^{\mathrm{T}}\boldsymbol{\omega}$ in (3.6.10) through (3.6.12) of Section 3.6.2 for arbitrary rotations in real-time. Another possible strategy to improve the running time (and increase the m_0 threshold) is to use simplified real-valued rather than complex-valued $\phi-$kernels

[5]Another strength of our method is its scalability with the available computational power in terms of the *accuracy* of the energy function, whose value largely depends on arbitrary decisions in the first place, rather than the *performance* of the haptic response cycle, which cannot be compromised. To put this into perspective, note that with most other available collision detection and constraint management tools for virtual assembly, the hardware imposes a restriction on the model complexity—e.g., limiting the triangle count $n \leq n_0$, simplifying the geometry to convex hulls or spherical unions, etc. Thus for $n > n_0$ the haptic frame rate deteriorates resulting in unsatisfactory performance, making large models completely intractable. In contrast, our method does not impose any such restriction, and keeps the performance satisfactory at all costs by systematic approximation.

to enable leveraging Hermitian symmetry property in Fourier transforms [226].

Figures 5.2.4 and 5.2.5 show the effects of successive FFT filtering described in Section 4.1.5 for different numbers of retained dominant modes $m' \leq m$ on the SDF (only imaginary part plotted on top rows). It also plots the score function on the bottom row, over a 2D section corresponding to a biaxial relative translation along the x_1x_2-plane through the 6D convolution. As more frequency domain data is kept, the geometric details—e.g., pertaining to the small pairs of cylindrical pegs and holes depicted by their axis lines in panel (a)—start to emerge in the SDF shortly after $m'/m \approx 0.2\%$ in panel (d). An important observation is that the maximum score (i.e., minimum energy) configuration (denoted by B) does not change much even with very few number of frequencies in panels (b) and (c). However, the slopes and curvatures of the energy profile characterizing the forces/torques and the stiffness of combined physical and geometric constraints do change significantly. For example, for the uniaxial motion from A to B, filtering with $m'/m < 1\%$ results in a relaxed collision response and geometric guidance along BA with 'soft' snapping at B, as a result of faded geometric details. By increasing the precision with $m'/m \geq 1\%$, a brisker response is imposed by larger transverse slopes along the AB trajectory with 'hard' snapping at B due to sharper curvature. However, the changes are insignificant after $m'/m > 2\%$ with $\sigma = 0.5$, which enables speed-ups of two orders of magnitude by disposing of 98% of the frequency data. One needs to use smaller σ factors to capture more geometric details thus more meaningful frequencies, and larger $\mathfrak{p} = \lambda_2/\lambda_1$ to impart stronger collision response compared to collision-free geometric guidance.

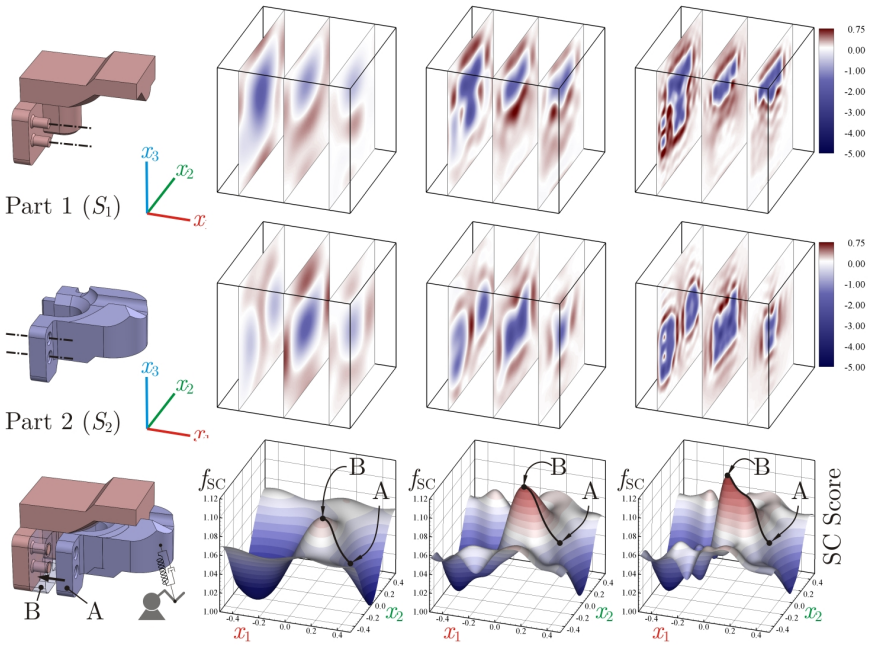


FIGURE 5.2.4: The effect of FFT filtering on part SDFs (top) and score variations versus biaxial relative translation (bottom) (cont'd to Fig. 5.2.5).

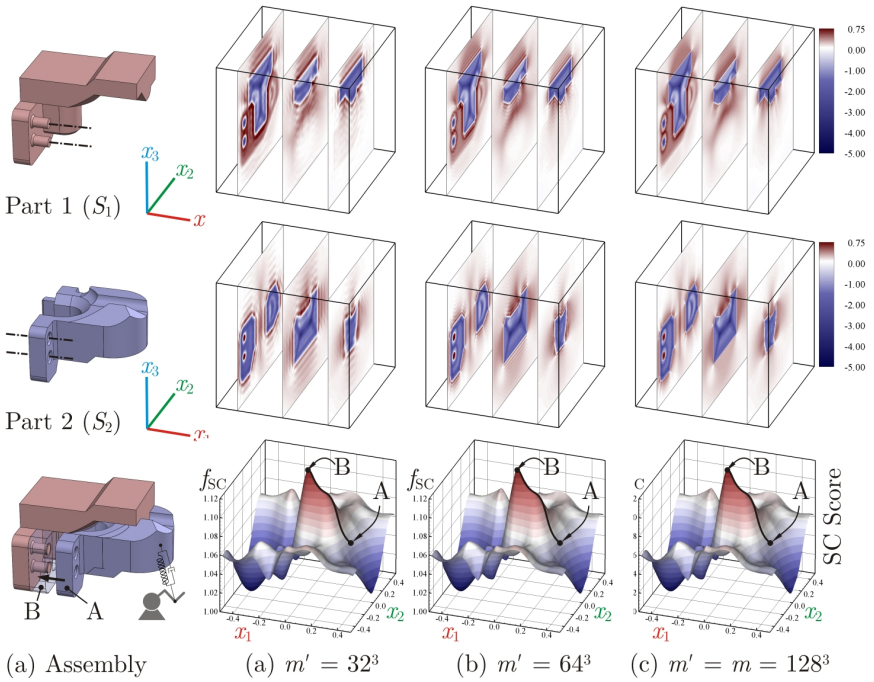


FIGURE 5.2.5: The effect of FFT filtering on part SDFs (top) and score variations versus biaxial relative translation (bottom) (cont'd from Fig. 5.2.4).

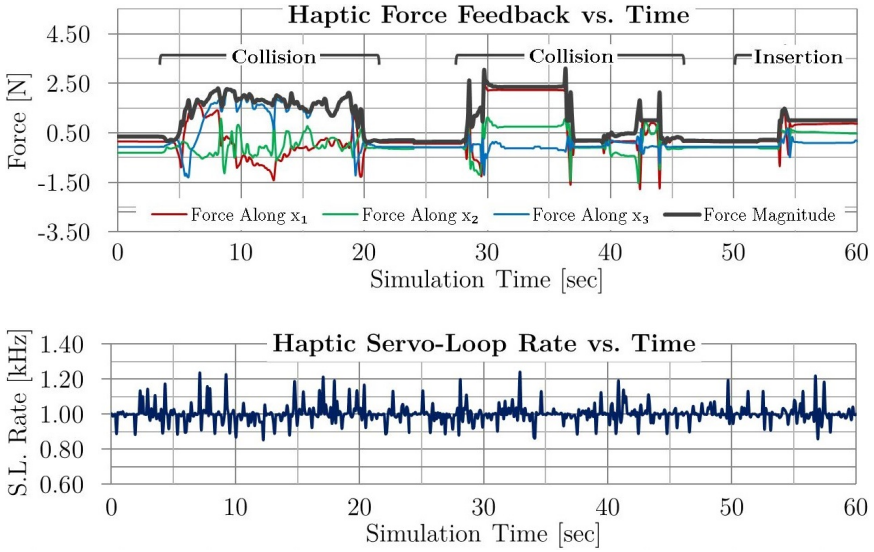


FIGURE 5.2.6: Performance of a haptic assembly simulation.

5.2.3 Haptic Experiments

I test the method to assemble the pair of parts in Fig. 5.2.1, keeping Part 1 stationary and manipulating Part 2 on **Config.–L** using **Device–A:** SensAble® Phantom® Omni®. Because the device is not capable of torque feedback, I choose to restrict the motion to the translational space, due to the observation that force response is insufficient to create a satisfactory user experience when dealing with rotational constraints. The SDFs is precomputed with $\sigma = 0.5$ for fine triangular meshes of $n_1 = 422{,}272$ and $n_2 = 320{,}384$, and a grid sample size of $m = 128^3 = 2{,}097{,}152$. The confinement of motion to translation only allows for exact precomputation of the convolution. However, after transferring to the Fourier domain, only $m' = 16^3 = 4{,}096$ (less than 0.2% of m) dominant modes of the SDFs is retained and remaining modes are zero-padded.

Figure 5.2.6 is the time plot of the haptic performance over a short assembly session, where the user explores colliding the objects from different sides and finally

assembles them into the obvious configuration. The haptic performance is consistent, always maintaining the servo-loop rate of $1.00^{+0.24}_{-0.15}$ kHz not only during free motion, but also during collision and final insertion.

Once again, the effectiveness of the method for virtual prototyping and assembly at an industrial scale must be further validated using a haptic device with 6 DOF input and 6 DOF output (e.g., **Device–E:** Haption Virtuose6D$^{\text{TM}}$ 35-45) with both force (of more than 10× larger magnitude) and torque feedback operating in a significantly larger workspace and augmented with more immersive tools (e.g., stereoscopic projection screens), which is postponed to future studies.

5.3 Other Applications

Mechanical assembly, despite being an important application of the concepts developed in Chapter 3 and implemented in Chapter 4, is not the only one. In fact, there are numerous studies and systems using alternative methods with notable success for haptic-assisted mechanical assembly and disassembly as reviewed in Chapter 2—many of which use CAD semantics to automatically identify the mating constraints for assembly and disassembly tasks. The assumption that the design history and functional significance of individual geometric features are known makes the task of VE development a lot easier. This is not the case when dealing with parts at an early stage of product development (e.g., conceptual design) or when the design semantics are simply not available (e.g., reverse engineering). In those situations, it is crucial to have a model that can work with the geometric information of the objects in a single snapshot alone without turning to auxiliary design documentation.

Another important application of the method is when one deals with mechanical systems that are *not* designed in the first place—e.g., biological systems that have evolved to the current state with no access to explicit information about the functional reasons for the existence of individual geometric features. A prominent example of such systems are cellular nano-machinery such as protein complexes.

Proteins are large biomolecules that are responsible for a vast array of biological functions inside the cell, and appear in the form of enzymes, antibodies, motor proteins, transport proteins, etc. [243]. The proteins typically function through binding with other proteins (or macromolecules such as DNA or RNA) which strongly depends on their 3D structure. In particular, it is a well-known fact that shape complementarity (Section 3.5) plays a significant role in the proper binding of protein molecules [56,243,317]. Therefore, the computer-aided prediction of the proper binding configurations (referred to as 'protein docking') is the key to understanding many biological processes in the cell. This knowledge is crucial toward the ultimate goal of modeling proper function or malfunction at molecular and cellular level (e.g., deadly diseases such as cancer, Alzheimer's, Parkinson's, etc.) and is central to a variety of bioengineering applications including 'protein engineering' [87,401]. Similar to the assembly planning and verification problems (Section 2.4.1), there are two fundamental approaches to the protein docking problem:

1. The first approach relies on an automatic prediction of the 6D relative configuration of a pair of protein domains or monomers (typically modeled as rigid bodies) in complex by leveraging the computational power of the machine. The correlation paradigm (Section 3.3) supplemented with FFT-based search algorithms have been developed to find the best fit configuration, also referred to

as the 'docking pose', e.g., in [11, 74, 121, 136, 225, 239, 240, 273] using affinity functions that can be viewed as special cases of the generic formulation in this thesis (Section 3.4.5). An extensive review of these and other methods for protein docking is given by Ritchie [338]. The main challenge common to all of these approaches is the high dimensionality of the search landscape which makes it difficult to find the optimum pose in a time-efficient manner.

2. The second approach relies on the cognitive capacity of the humans to explore the different binding pockets of a protein by trial-and-error, which can benefit from graphics- and haptics-assistance. This allows for a quick rejection of a large subset of the 6D configuration space and rapid exploration of the feasible poses using the user's understanding of spatial relations and sense of proper fit/contact. Haptic feedback, in particular, has been applied to different molecular simulation and training applications [110, 351, 377, 429]. However, to the best of my knowledge, FFT-based convolution methods have not been utilized for interactive simulation in spite of their popularity for offline optimization.

An important application of interactive protein docking is in rational drug design. A large class of drugs are small molecules or medium-sized proteins that bind to the active site of enzymes and inhibit the (mal)function of cellular processes. The early-stage development of a 'lead compound' as a scaffold before adding biochemical functional groups is governed by shape complementarity [243]. For example, Fig. 5.3.1 shows the HIV protease, an enzyme that plays a key role in the progression of the HIV disease by cleaving inactive viral polypeptides in the cell. A commercial drug called Saquinavir is a small molecule that binds to the active site of the HIV protease and inhibits substrate binding. As shown in Fig. 5.3.1 (c), shape complementarity

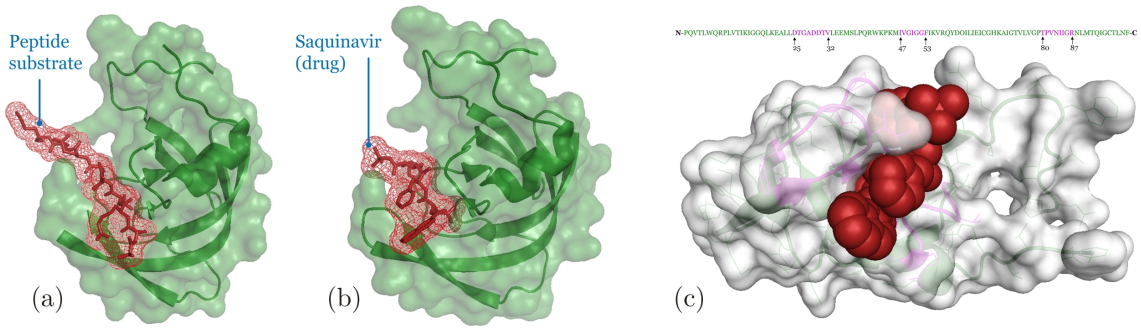


FIGURE 5.3.1: A monomeric unit of HIV protease whose active site is occupied by a peptide substrate (PDB code 1F7A) (a) or an inhibitor called Saquinavir (PDB code 1FB7)(b). As shown in (c), shape complementarity is the key to complex formation.

plays a key role in the competitiveness of the drug with the natural substrate, which in turn is crucial to the effectiveness of the drug.

The active site of the HIV protease—designated by color purple from the rest of the molecule in Fig. 5.3.1 (c)—is shown in Fig. 5.3.2 (a) in complex with Saquinavir. As depicted by the section views in Fig. 5.3.2 (b, c), the proper contact/fit configuration between the two molecules is not trivial to determine by visual inspection. Unlike mating features in mechanical assembly problems, the binding features in protein docking are harder to spot because of more complex geometry with less obvious functional reasoning. As a result, the geometric energy model—from which haptic force and torque feedback is generated—can be notably beneficial in addition to visual cues to explore and evaluate different docking poses.

Figure 5.3.3 shows the manual docking practice using 3D printed plastic models of the molecules shown in Fig. 5.3.2. Figures 5.3.4 and 5.3.5 show the simulated docking process using **Device–A** and **Device–E**, respectively, operated by the same software implementation presented in Chapter 4.

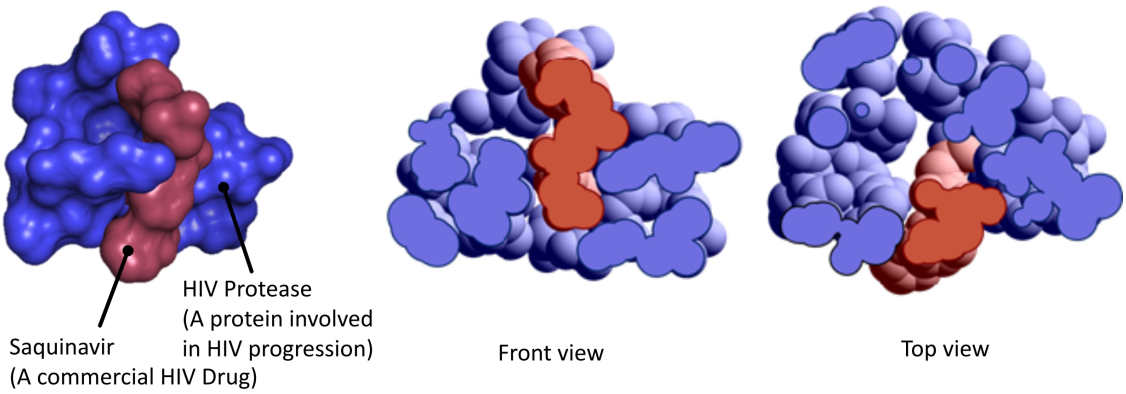


FIGURE 5.3.2: Active site of HIV protease in complex with Saquinavir.

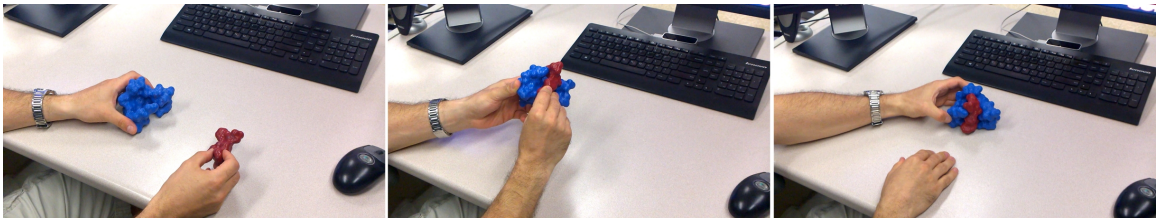


FIGURE 5.3.3: Manual docking of the 3D printed plastic models of the HIV protease active site and Saquinavir (Fig. 5.3.2).

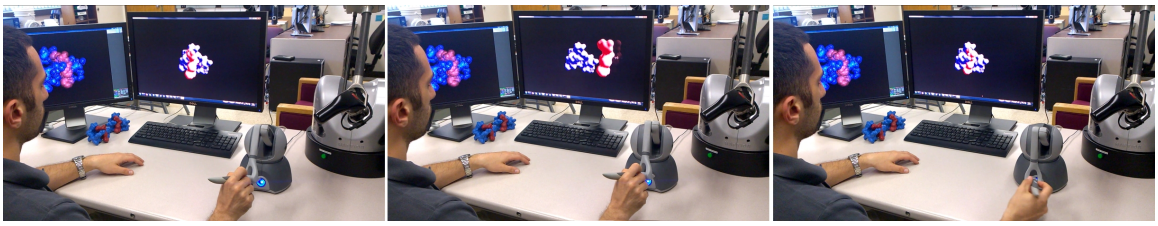


FIGURE 5.3.4: Haptic-assisted simulated docking of the computer models of the HIV protease active site and Saquinavir (Fig. 5.3.2) using SensAble® Phantom® Omni®.

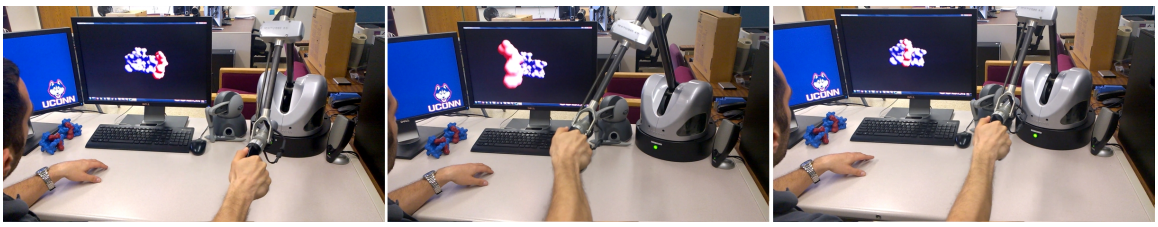


FIGURE 5.3.5: Haptic-assisted simulated docking of the computer models of the HIV protease active site and Saquinavir (Fig. 5.3.2) using Haption Virtuose6D^{TM} 35-45.

Conclusions

Haptic-enabled assembly planning has been restrained for a long time from achieving its full potential, due to the challenges presented by the competing objectives of handling high geometric complexity while maintaining a response rate of 1 kHz.

Lately, the dominant direction in this area has been aligned with a hybrid approach [316,402], separating the simulation into a 'free motion' phase, using unilateral (i.e., inequality) 'physical constraints' originated from collision detection; and a 'fine insertion' phase, using bilateral (i.e., equality) 'mating constraints' (e.g., of geometric and/or kinematic types) introduced artificially to limit the DOF. While the former fail to produce dynamically stable guidance for low-clearance insertion, the latter are either dependent on a priori manual specifications by the user, or are limited to simple semialgebraic features (e.g., planar, cylindrical, spherical, or conical) that can be identified automatically from CAD semantics using heuristic algorithms. The identification of the switch criteria between the two phases, on the one hand, and modeling the insertion constraints for different contact surface features, on the other hand, remained an open problem for objects of arbitrary shape.

In this thesis, I proposed a novel paradigm that unifies the two modes into a single interaction, by introducing a 'geometric energy' field that models collision response

and geometric guidance for arbitrarily complex shapes. This energy function expands the collision penalty function [270] into one that not only penalizes the configurations with interpenetration—hence produces the repulsive collision response—but also rewards the configurations with high *shape complementarity*—hence produces the attractive guidance forces. This was accomplished by formulating the energy function as a cross-correlation of new descriptors of shape, to which I referred as the skeletal density functions (SDF). The SDF interactions can be conceptualized as generic replacements for ad hoc virtual fixtures [343, 344] or simplistic mating constraints, and apply to objects of arbitrary shape. I showed that this approach automatically ensures a continuous transition between collision response in free movement to insertion guidance in low-clearance or precise-fit assembly, avoiding the two-phase approach along with its several drawbacks—including the failure to prevent collision events outside the insertion site, and the need for blending the force feedback during the switch. The unified paradigm provides a promising alternative direction for solving virtual assembly problems, in general, and for haptic rendering, in particular.

Although the generality of the method inevitably imparts additional computational complexity, I demonstrated that the guidance forces and torques can be efficiently computed in the Fourier domain, where the convolution converts to pointwise product of SDF amplitudes. The SDF shape descriptors and their Fourier expansions can be computed in a preprocessing step that takes a few minutes per part for reasonable mesh and sample sizes. For real-time computations, a very small subset of the dominant frequency domain data can be used to compute the pointwise multiplication followed by an inverse FFT. The results confirm that such low-pass filtering of the SDF information, together with the computational power offered by the modern GPUs, enable fast evaluation of the haptic forces and torques within the available 1

millisecond time frame during haptic assembly, with little compromise in accuracy.

Unlike the existing approaches to collision detection or constraint management which are restricted by topological complexity (e.g., connectivity and number of holes), geometric complexity (e.g., convexity and type of surfaces), or syntactic complexity (e.g., number of triangles or voxels), the proposed method does not impose any such restriction. Instead, it allows for a systematic trade-off between the desired fidelity and computational efficiency regardless of the input size or complexity.

The outcome of this research is a powerful paradigm that streamlines haptic assembly using spectral analysis of shape descriptors in the Fourier domain, and opens up new promising theoretical and computational directions for VR researchers and haptics software developers. Nevertheless, there are still several important problems that need to be addressed to realize the full potential of this method:

- The uniform sampling required by the classical uniform FFT algorithm [102] is not memory-efficient when large models with high levels are detail are rasterized. A promising new direction in this aspect is provided in [20, 21] using ideas from protein docking [11] and tools such as the nonuniform FFT algorithm [323].

- Although the translational (i.e., commutative) component of the SDF convolution is handled efficiently by the Fourier methods, the rotational (i.e., noncommutative) component still requires sampling and interpolation. An interesting subject to look at in this area is the noncommutative harmonic analysis [86].

- The preliminary validation results presented in this thesis are not nearly as strong as their counterparts presented in the literature for the currently popular approaches. Before this method can be adopted into physics simulation engines (PSE), a more robust implementation along with extensive testing is imperative.

Bibliography

[1] ASME Y14.5-2009: Dimensioning and tolerancing: Engineering drawing and related documentation practices. An international standard, American Society of Mechaninical Engineers (ASME), 2009.

[2] R. J. Adams and B. Hannaford. Stable haptic interaction with virtual environments. *IEEE Transactions on Robotics and Automation*, 15(3):465–474, 1999.

[3] R. J. Adams and B. Hannaford. Control law design for haptic interfaces to virtual reality. *IEEE Transactions on Control Systems Technology*, 10(1):3–13, 2002.

[4] I. Aguinaga, D. Borro, and L. Matey. Parallel RRT-based path planning for selective disassembly planning. *The International Journal of Advanced Manufacturing Technology*, 36(11):1221–1233, 2007.

[5] C. Alexander. *Notes on the Synthesis of Form*. Harvard University Press, 1964.

[6] S. Andrews, J. Mora, J. Lang, and W. S. Lee. Hapticast: A physically-based 3D game with haptic feedback. In *Proceedings of FuturePlay*, 2006.

[7] M. Anitescu, F. A. Potra, and D. E. Stewart. Time-stepping for three-dimensional rigid body dynamics. *Computer Methods in Applied Mechanics and Engineering*, 177(3-4):183–197, 1999.

[8] D. Attali, J. D. Boissonnat, and H. Edelsbrunner. Stability and computation of medial axes: A state-of-the-art report. *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, pages 109–125, 2009.

[9] Multiple Authors. *Towards a Machine with Interactive Skills*, pages 31–38. Understanding Computer-Computer Images. Time-Life Books, 1986.

[10] C. L. Bajaj, B. Bauer, R. Bettadapura, and A. Vollrath. Nonuniform Fourier transforms for rigid-body and multidimensional rotational correlations. *SIAM Journal on Scientific Computing*, 35(4):B821–B845, 2013.

[11] C. L. Bajaj, R. Chowdhury, and V. Siddahanavalli. F2Dock: Fast Fourier protein-protein docking. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 8(1):45–58, 2011.

[12] M. Balasubramaniam, S. Ho, S. Sarma, and Y. Adachi. Generation of collision-free 5-axis tool paths using a haptic surface. *Computer-Aided Design*, 34(4):267–279, 2002.

[13] D. Baraff. Fast contact force computation for nonpenetrating rigid bodies. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, pages 23–34, New York, NY, USA, 1994.

[14] D. Baraff. Interactive simulation of solid rigid bodies. *Journal of Computer Graphics and Applications*, 15(3):63–75, 1995.

[15] J. Barbič and D. James. Time-critical distributed contact for 6-DOF haptic rendering of adaptively sampled reduced deformable models. In *Proceedings of the ACM SIGGRAPH'2007/Eurographics Symposium on Computer Animation*, pages 171–180, Aire-la-Ville, Switzerland, Switzerland, 2007.

[16] J. Barbič and D. L. James. Six-DoF haptic rendering of contact between geometrically complex reduced deformable models. *IEEE Transactions on Haptics*, 1(1):39–52, 2008.

[17] M. Behandish. Lie groups, Lie algebras, and correlation gradients: A tutorial. Technical report, University of Connecticut, 2015.

[18] M. Behandish and H. T. Ilieş. Peg-in-hole revisited: A generic force model for haptic assembly. In *Proceedings of the 2014 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE'2014)*, number DETC2014-35290, pages V01BT02A041:1–10. American Society of Mechanical Engineers (ASME), 2014.

[19] M. Behandish and H. T. Ilieş. Shape complementarity analysis for objects of arbitrary shape. Technical Report CDL-TR-14-01, University of Connecticut, 2014.

[20] M. Behandish and H. T. Ilieş. Analytic methods for geometric modeling via spherical decomposition. *Computer-Aided Design*, 70:100–115, 2015. 2015 SIAM/ACM Symposium on Solid and Physical Modling (GD/SPM'2015).

[21] M. Behandish and H. T. Ilieş. Analytic methods for geometric modeling via spherical decomposition. In *2015 SIAM/ACM Symposium on Solid and Physical Modling (GD/SPM'2015)*, volume 70, pages 100–115, 2015.

[22] M. Behandish and H. T. Ilieş. Haptic assembly using skeletal densities and Fourier transforms. In *Proceedings of the 2015 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE'2015)*, number DETC2015-47923, pages V01BT02A060:1–11. American Society of Mechanical Engineers (ASME), 2015.

[23] M. Behandish and H. T. Ilieş. Peg-in-hole revisited: A generic force model for haptic assembly. *Journal of Computing and Information Science in Engineering (JCISE)*, 15(4):041004:1–11, 2015.

[24] M. Behandish and H. T. Ilieş. Haptic assembly using skeletal densities and Fourier transforms. *Journal of Computing and Information Science in Engineering (JCISE)*, 16(2):021002:1–11, 2016.

[25] Jan Bender, Kenny Erleben, and Jeff Trinkle. Interactive simulation of rigid body dynamics in computer graphics. *Computer Graphics Forum*, 33(1):246–270, 2014.

[26] G. R. Bennett. The application of virtual prototyping in the development of complex aerospace products. *Aircraft Engineering and Aerospace Technology*, 69(1):19–25, 1997.

[27] G. van den Bergen. Efficient collision detection of complex deformable models using AABB trees. *Journal of Graphics Tools*, 2(4):1–13, 1997.

[28] A. Bernard and A. Fischer. New trends in rapid product development. *CIRP Annals - Manufacturing Technology*, 51(2):635–652, 2002.

[29] B. T. Bethea, A. M. Okamura, M. Kitagawa, T. P. Fitton, S. M. Cattaneo, V. L. Gott, W. A. Baumgartner, and D. D. Yuh. Application of haptic feedback to robotic surgery. *Journal of Laparoendoscopic and Advanced Surgical Techniques*, 14(3):191–195, 2004.

[30] A. Bhatti, Y. B. Khoo, D. Creighton, J. Anticev, S. Nahavandi, and M. Zhou. Haptically enabled interactive virtual reality prototype for general assembly. In *Proceedings of the 2008 World Automation Congress (WAC'2008)*, pages 1–6, 2008.

[31] A. Bhatti, S. Nahavandi, Y. B. Khoo, D. Creighton, J. Anticev, and M. Zhou. Haptically enable interactive virtual assembly training system development and evaluation. In *Proceedings of the 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (SIMTECT'2009)*, pages 1–6. SIMTECT, 2009.

[32] V. Boor, M. H. Overmars, and A. F. van der Stappen. The Gaussian sampling strategy for probabilistic roadmap planners. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, volume 2, pages 1018–1023. IEEE, 1999.

[33] G. Boothroyd and P. Dewhurst. *Product Design for Assembly*. Boothroyd Dewhurst Incorporated, 1987.

[34] M. Bordegoni. Deliverable D1 hardware and software technology and cognitive ergonomics update. Technical Report at KAEMaRT group, Politecnico di Milano, 2004.

[35] M. Bordegoni, G. Caruso, F. Ferrise, and . Giraudo. A mixed environment for ergonomics tests: Tuning of stereo viewing parameters. In *Proceedings of the 5th Eurographics Italian Chapter Conference*. The Eurographics Association, 2007.

[36] M. Bordegoni, G. Colombo, and Formentini. Haptic technologies for the conceptual and validation phases of product design. *Computers and Graphics*, 30(3):377–390, 2006.

[37] M. Bordegoni and U. Cugini. A conceptual design tool based on innovative haptic interfaces. In *Proceedings of the 2006 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE'2006)*, pages 897–904. American Society of Mechanical Engineers, 2006.

[38] M. Bordegoni and U. Cugini. Haptic modeling in the conceptual phases of product design. *Virtual Reality*, 9(2-3):192–202, 2006.

[39] M. Bordegoni and U. Cugini. The role of haptic technology in the development of aesthetic driven products. *Journal of Computing and Information Science in Engineering (JCISE)*, 8(4):041006, 2008.

[40] M. Bordegoni and U. Cugini. A CAD system based on haptic modelling for conceptual design. *International Journal of Product Development*, 11(1-2):25–46, 2010.

[41] M. Bordegoni, U. Cugini, P. Belluco, and M. Aliverti. Evaluation of a haptic-based interaction system for virtual manual assembly. In Randall Shumaker, editor, *Virtual and Mixed Reality*, volume 5622 of *Lecture Notes in Computer Science*, pages 303–312. Springer Berlin Heidelberg, 2009.

[42] M. Bordegoni, U. Cugini, G. Caruso, and S. Polistina. Mixed prototyping for rapid design review of information appliances. In *Proceedings of IDMME'2008 Virtual Concept*, Beijing, China, 2008.

[43] M. Bordegoni, U. Cugini, G. Caruso, and S. Polistina. Mixed prototyping for product assessment: A reference framework. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 3(3):177–187, 2009.

[44] M. Bordegoni, U. Cugini, G. Caruso, and S. Polistina. The role of mixed prototyping in product design assessment. In *Proceedings of the 2nd International Conference on Research into Design (ICoRD'2009)*, Beijing, China, 2009.

[45] M. Bordegoni, U. Cugini, and F. Ferrise. Analysis of designers manual skills for the development of a tool for aesthetic shapes evaluation. In *Proceedings of IDMME'2010 Virtual Concept*, 2010.

[46] M. Bordegoni, U. Cugini, and F. Ferrise. Requirements for an enactive tool to support skilled designers in aesthetic surfaces definition. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 6(2):83–91, 2012.

[47] M. Bordegoni, U. Giraudo, G. Caruso, and F. Ferrise. Ergonomic interactive testing in a mixed-reality environment. In R. Shumaker, editor, *Virtual Reality*, volume 4563 of *Lecture Notes in Computer Science*, pages 431–440. Springer Berlin Heidelberg, 2007.

[48] D. Borro, J. Savall, A. Amundarain, J. J. Gil, A. Garcia-Alonso, and L. Matey. A large haptic device for aircraft engine maintainability. *Computer Graphics and Applications*, 24(6):70–74, 2004.

[49] P. Bosscher and I. Ebert-Uphoff. Digital clay: architecture designs for shape-generating mechanisms. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA'2003)*, volume 1, pages 834–841, 2003.

[50] P. Bourdot, T. Convard, F. Picon, M. Ammi, D. Touraine, and J. M. Vezien. VR-CAD integration: Multimodal immersive interaction and advanced haptic paradigms for implicit edition of CAD models. *Computer-Aided Design*, 42(5):445–461, 2010.

[51] P. Bourdot, M. Krus, and R. Gherbi. Cooperation between reactive 3D objects and a multimodal X window kernel for CAD. In H. Bunt, R. J. Beun, and T. Borghuis, editors, *Multimodal Human-Computer Communication*, volume 1374 of *Lecture Notes in Computer Science*, pages 188–212. Springer Berlin Heidelberg, 1998.

[52] F. Boussuge, J. C. Léon, S. Hahmann, and L. Fine. An analysis of DMU transformation requirements for structural assembly simulations. In *Proceedings of the 8th International Conference on Engineering Computational Technology*, 2012.

[53] G. Bouyer, P. Bourdot, and M. Ammi. Supervision of task-oriented multimodal rendering for VR applications. In *Proceedings of the 13th Eurographics Conference on Virtual Environments (EGVE'2007)*, pages 93–100. Eurographics Association, 2007.

[54] D. Bowman, D. Johnson, and L. Hodges. Testbed evaluation of virtual environment interaction techniques. *Presence*, 10(1):75–95, 2001.

[55] G. Bradshaw and C. O'Sullivan. Adaptive medial-axis approximation for sphere-tree construction. *ACM Transactions on Graphics (TOG)*, 23(1):1–26, 2004.

[56] C. Branden and J. Tooze. *Introduction to Protein Structure*, volume 2. Garland Science, 2nd edition edition, 1999.

[57] R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, pages 594–603, New York, NY, USA, 2002.

[58] R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. *ACM Transactions on Graphics (TOG)*, 21(3):594–603, 2002.

[59] J. Broeren, A. L. Bellner, M. Fogelberg, O. Göransson, D. Goude, B. Johansson, P. Larsson, K. Pettersson, and R. Rydmark. Exploration of computer games in rehabilitation for brain damage. In *Proceedings of the 7th International Conference on Disability, Virtual Reality and Associated Technologies*, 2008.

[60] J. Broeren, M. Rydmark, and K. S. Sunnerhagen. Virtual reality and haptics as a training device for movement rehabilitation after stroke: A single-case study. *Archives of Physical Medicine and Rehabilitation*, 85(8):1247–1250, 2004.

[61] H. J. Bullinger, R. Breining, and W. Bauer. Virtual prototyping–state of the art in product design. In *26th International Conference on Computers and Industrial Engineering*, pages 103–107, 1999.

[62] G. Burdea and P. Coiffet. *Virtual Reality Technology*. Wiley-IEEE Press, 2nd edition edition, 2003.

[63] G. C. Burdea, G. C. Burdea, and C. Burdea. *Force and Touch Feedback for Virtual Reality*. Wiley New York, 1996.

[64] W. Carlson. A critical history of computer graphics and animation. Online course, Ohio State University.

[65] G. Caruso and P. Belluco. Robotic arm for car dashboard layout assessment in mixed reality environment. In *Proceedings of the 2010 IEEE International Conference on Robot and Human Interactive Communication (RO-MAN'2010)*, pages 62–68, 2010.

[66] G. Caruso, S. Polistina, and M. Bordegoni. Collaborative mixed-reality environment to support the industrial product development. In *Proceedings af the 2011 ASME World Conference on Innovative Virtual Reality*, pages 207–215. American Society of Mechanical Engineers, 2011.

[67] G. Caruso, S. Polistina, M. Bordegoni, and M. Aliverti. Collaborative mixed-reality platform for the design assessment of cars interior. In R. Shumaker, editor, *Virtual and Mixed Reality - Systems and Applications*, volume 6774 of *Lecture Notes in Computer Science*, pages 299–308. Springer Berlin Heidelberg, 2011.

[68] L. S. H. Chan and K. S. Choi. Integrating PhysX and OpenHaptics: Efficient force feedback generation using physics engine and haptic devices. In *Proceedings of the 2009 Joint Conferences on Pervasive Computing (JCPC'2009),*, pages 853–858, 2009.

[69] J. W. Chang, W. Wang, and M. S. Kim. Efficient collision detection using a dual OBB-sphere bounding volume hierarchy. *Computer-Aided Design*, 42(1):50–57, 2010.

[70] Y. S. Chang, K. T. McDonnell, and H. Qin. A new solid subdivision scheme based on box splines. In *Proceedings of the Seventh ACM Symposium on Solid Modeling and Applications (SMA'2002)*, pages 226–233, 2002.

[71] F. Chazal and A. Lieutier. Stability and homotopy of a subset of the medial axis. In *Proceedings of the 9th ACM Symposium on Solid Modeling and Applications (SM'2004)*, pages 243–248. Eurographics Association, 2004.

[72] F. Chazal and A. Lieutier. The $\lambda-$medial axis. *Graphical Models*, 67(4):304–331, 2005.

[73] F. Chazal and R. Soufflet. Stability and finiteness properties of medial axis and skeleton. *Journal of Dynamical and Control Systems*, 10(2):149–170, 2004.

[74] R. Chen, L. Li, and Z. Weng. ZDOCK: An initial-stage protein-docking algorithm. *Proteins: Structure, Function, and Bioinformatics*, 52(1):80–87, 2003.

[75] R. Chen and Z. Weng. A novel shape complementarity scoring function for protein-protein docking. *Proteins: Structure, Function, and Bioinformatics*, 51(3):397–408, 2003.

[76] Y. H. Chen, L. L. Lian, and X. J. He. Haptic rendering of three-dimensional heterogeneous fine surface features. *Computer-Aided Design and Applications*, 5(1-4):1–16, 2008.

[77] Y. H. Chen, Y. Z. Wang, and Z. Y. Yang. Towards a haptic virtual coordinate measuring machine. *International Journal of Machine Tools and Manufacture*, 44(10):1009–1017, 2004.

[78] Y. H. Chen and Z. Y. Yang. Haptic rendering based on spatial run-length encoding. *Robotics and Computer-Integrated Manufacturing*, 20(3):237–246, 2004.

[79] Y. H. Chen and Z. Y. Yang. Haptic modeling as an integrating tool for product development. *The International Journal of Advanced Manufacturing Technology*, 33(7-8):635–642, 2007.

[80] Y. H. Chen, Z. Y. Yang, and L. Lian. Haptic modeling in rapid product development. *Computer-Aided Design and Applications*, 1(1-4):577–584, 2004.

[81] Y. H. Chen, Z. Y. Yang, and L. Lian. On the development of a haptic system for rapid product development. *Computer-Aided Design*, 37(5):559–569, 2005.

[82] Y. H. Chen, Z. Y. Yang, and y. z. Wang. Haptic modeling for a virtual coordinate measuring machine. *International Journal of Production Research*, 43(9):1861–1878, 2005.

[83] C. Cheng-Jun, W. Yun-feng, and L. Niu. Research on interaction for virtual assembly system with force feedback. In *Proceedings of the 2010 International Conference on Information and Computing (ICIC'2010)*, volume 2, pages 147–150, 2010.

[84] Chen Cheng-Jun, A.Y.C. Nee, S.K. Ong, Zhou Yi-qi, and Qu Bin. An improved haptic interface for virtual assembly operations. In *Second Workshop on Digital Media and its Application in Museum Heritages*, pages 410–415, Dec 2007.

[85] D. G. Cheshire, M. A. Evans, and C. J. Dean. Haptic modeling - an alternative industrial design methodology? In *Proceedings of Eurohaptics 2001*, volume 2001, pages 124–128, 2001.

[86] G. S. Chirikjian and A. B. Kyatkin. *Engineering Applications of Noncommutative Harmonic Analysis: with Emphasis on Rotation and Motion Groups*. CRC Press, 1st edition, 2010.

[87] Gregory S Chirikjian, Kazem Kazerounian, and Constantinos Mavroidis. Analysis and design of protein based nanodevices: Challenges and opportunities in mechanical design. *Journal of Mechanical Design*, 127(4):695–698, 2005.

[88] H. I. S. Choi and H. Z. Tan. Effect of update rate on perceived instability of virtual haptic texture. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2004)*, volume 4, pages 3577–3582, 2004.

[89] H. I. S. Choi and H. Z. Tan. Discrimination of virtual haptic textures rendered with different update rates. In *Proceedings of the 2005 Joint Conference on World Haptics, the Eurohaptics 2005, and the Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 114–119, 2005.

[90] Christiand. Assembly simulations in virtual environments with optimized haptic path and sequence. *Robotics and Computer-Integrated Manufacturing*, 27(2):306–317, 2011.

[91] Christiand and J. Yoon. A novel optimal assembly algorithm for the haptic interface application of a virtual maintenance system. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA'2008)*, pages 3612–3617, 2008.

[92] Christiand, J. Yoon, M. Auralius, and Wonpil Y. An enhanced haptic assembly simulation system for the efficiency of assembly tasks. In *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2009)*, pages 1241–1246, 2009.

[93] J. D. Cohen, M. C. Lin, D. Manocha, and M. Ponamgi. I-COLLIDE: An interactive and exact collision detection system for large-scale environments. In *Proceedings of the 1995 Symposium on Interactive 3D Graphics (I3D'1995)*, New York, NY, USA, 1995.

[94] J. E. Colgate, M. C. Stanley, and J. M. Brown. Issues in the haptic display of tool use. In *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 140–145, 1995.

[95] G. Colombo, F. De Angelis, and L. Formentini. A mixed virtual reality-haptic approach to carry out ergonomics test on virtual prototypes. In *Proceedings of TMCE'2006*, pages 18–22, 2006.

[96] M. B. Colton and J. Hollerbach. Automated modeling of nonlinear mechanisms for virtual prototyping. In *ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 1349–1355. American Society of Mechanical Engineers, 2004.

[97] M. B. Colton and J. M. Hollerbach. Identification of nonlinear passive devices for haptic simulations. In *Proceedings on the 1st Joint Eurohaptics Conference 2005, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems 2005, and World Haptics 2005*, pages 363–368, 2005.

[98] P G. Comba. A procedure for detecting intersections of three-dimensional objects. *The ACM Journal*, 15(3):354–366, 1968.

[99] S. D. Conte and C. W. D. Boor. *Elementary Numerical Analysis: An Algorithmic Approach.* McGraw-Hill Higher Education, 1980.

[100] T. Convard and P. Bourdot. History-based reactive objects for immersive CAD. In *Proceedings of the 9th ACM Symposium on Solid Modeling and Applications*, pages 291–296. Eurographics Association, 2004.

[101] T. Convard, P. Bourdot, and J. M. Vezien. Managing deformable objects in cluster rendering. In V. S. Sunderam, G. D. van Albada, P. M. A. Sloot, and J. J. Dongarra, editors, *Computational Science (ICCS'2005)*, volume 3515 of *Lecture Notes in Computer Science*, pages 290–297. Springer Berlin Heidelberg, 2005.

[102] Tukey J. Cooley, J. W. An algorithm for the machine calculation of complex Fourier series. *Mathematics of CSomputation*, 19(90):297–301, 1965.

[103] A. S. Coutee and Bras. Collision detection for virtual objects in a haptic assembly and disassembly simulation environment. In *Proceedings of the 2002 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE'2002)*, pages 11–20, 2002.

[104] A. S. Coutee, S. D. McDermott, and B. Bras. A haptic assembly and disassembly simulation environment and associated computational load optimization techniques. *Journal of Computing and Information Science in Engineering*, 1(2):113–122, 2001.

[105] U. Cugini and M. Bordegoni. Touch and design: Novel haptic interfaces for the generation of high quality surfaces for industrial design. *The Visual Computer*, 23(4):233–246, 2007.

[106] B. Curto, V. Moreno, and F. J. Blanco. A general method for C-space evaluation and its application to articulated robots. *IEEE Transactions on Robotics and Automation*, 18(1):24–31, 2002.

[107] F. Dachille IX, H. Qin, and A. E. Kaufman. A novel haptics-based interface and sculpting system for physics-based geometric design. *Computer-Aided Design*, 33(5):403–420, 2001.

[108] F. Dachille IX, H. Qin, A. E. Kaufman, and J. El-Sana. Haptic sculpting of dynamic surfaces. In *Proceedings of the 1999 Symposium on Interactive 3D Graphics (I3D'1999)*, pages 103–110. ACM, 1999.

[109] F. Dai, W. Felger, T. Frühauf, M. Göbel, D. Reiners, and G. Zachmann. Virtual prototyping examples for automotive industries. volume 96, 1996.

[110] B. Daunay and S. Daunay. Stable six degrees of freedom haptic feedback for flexible ligand-protein docking. *Computer-Aided Design*, 41(12):886–895, 2009.

[111] B. R. de Araujo, T. Guerreiro, M. J. Fonseca, J. A. Jorge, J. M. Pereira, M. Bordegoni, F. Ferrise, M. Covarrubias, and M. Antolini. An haptic-based immersive environment for shape analysis and modelling. *Journal of Real-Time Image Processing*, 5(2):73–90, 2010.

[112] T. Deviprasad and Kesavadas. Virtual prototyping of assembly components using process modeling. *Journal of Manufacturing Systems*, 22(1):16–27, 2003.

[113] J. Dingliana and C. O'Sullivan. Graceful degradation of collision handling in physically based animation. *Computer Graphics Forum*, 19(3):239–248, 2000.

[114] Y. Duan, J. Hua, and H. Qin. HapticFlow: PDE-based mesh editing with haptics: Research articles. *Computer Animation Virtual Worlds*, 15(3-4):193–200, 2004.

[115] C. Duriez, C. Andriot, and A. Kheddar. A multithreaded approach for deformable/rigid contacts with haptic feedback. In *Proceedings of the 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 272–279, 2004.

[116] C. Duriez, C. Andriot, and A. Kheddar. Signorini's contact model for deformable objects in haptic simulations. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2004)*, volume 4, 2004.

[117] C. Duriez, F. Dubois, A. Kheddar, and C. Andriot. Realistic haptic rendering of interacting deformable objects in virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 12(1):36–47, 2006.

[118] A. Edalat and A. Lieutier. Foundation of a computable solid modelling. *Theoretical Computer Science*, 284(2):319–345, 2002.

[119] S. A. Ehmann and M. C. Lin. Accelerated proximity queries between convex polyhedra by multi-level Voronoi marching. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2000)*, volume 3, 2000.

[120] S. A. Ehmann and M. C. Lin. Accurate and fast proximity queries between polyhedra using convex surface decomposition. *Computer Graphics Forum*, 20(3):500–511, 2001.

[121] M. Eisenstein and E. Katchalski-Katzir. On proteins, grids, correlations, and docking. *Comptes Rendus Biologies*, 327(5):409–420, 2004.

[122] A. El Saddik. *Haptics Technologies: Bringing Touch to Multimedia*. Springer, 1st edition, 2011.

[123] M. Evans, D. Wallace, D. Cheshire, and B. Sener. An evaluation of haptic feedback modelling during industrial design practice. *Design Studies*, 26(5):487–508, 2005.

[124] M. A. Evans, D. G. Cheshire, and C. J. Dean. An investigation into the use of haptic modelling during industrial design activity. Technical report, Loughborough University, 2000.

[125] D. Faas and J. M. Vance. BREP identification during voxel-based collision detection for haptic manual assembly. In *Proceedings of the 2011 ASME World Conference on Innovative Virtual Reality*, pages 145–153. American Society of Mechanical Engineers, 2011.

[126] K. J. Fernandes, V. H. Raja, and J. Eyre. Immersive learning system for manufacturing industries. *Computers in Industry*, 51(1):31–40, 2003.

[127] A. Luiz S. Ferreira, Leonardo C. de M., E. Esteves C. de Miranda, and S. G. Sakamoto. A survey of interactive systems based on brain-computer interfaces. *SBC Journal on Interactive Systems*, 4(1):3–13, 2013.

[128] F. Ferrise, M. Bordegoni, and J. Lizaranzu. Product design review application based on a vision-sound-haptic interface. In S. Nordahl, R.and Serafin, F. Fontana, and S. Brewster, editors, *Haptic and Audio Interaction Design*, volume 6306 of *Lecture Notes in Computer Science*, pages 169–178. Springer Berlin Heidelberg, 2010.

[129] R. Fetecau, J. Marsden, M. Ortiz, and M. West. Nonsmooth Lagrangian mechanics and variational collision integrators. *SIAM Journal on Applied Dynamical Systems*, 2(3):381–416, 2003.

[130] S. Fisher and M. Lin. Deformed distance fields for simulation of non-penetrating flexible bodies. In *Computer Animation and Simulation*, pages 99–111. Springer Vienna, 2001.

[131] W. B. Fisher, E. E. Hartquist, A. A. G. Requicha, and H. B. Voelcker. The PADL-1.0/n processor: Overview & system documentation. Production Automation Project, System Document No. 01 (PADL/SD-01), University of Rochester, 1977.

[132] M. Frey, D. E. Johnson, and J. M. Hollerbach. Full-arm haptics in an accessibility task. In *Proceedings of the 2008 Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 405–412, 2008.

[133] M. Frigo and S. G. Johnson. FFTW: An adaptive software architecture for the FFT. In *Proceedings of THE 1998 IEEE International Conference on Acoustics Speech and Signal Processing*, volume 3, pages 1381–1384, 1998.

[134] M. Frigo and S. G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE, Special issue on Program Generation, Optimization, and Platform Adaptation*, 93(2):216–231, 2005.

[135] A. Fuhrmann, G. Sobotka, and C. Groß. Distance fields for rapid collision detection in physically based modeling. In *Proceedings of the GraphiCon'2003*, pages 58–65, 2003.

[136] H. A. Gabb, R. M. Jackson, M. J. E. Sternberg, et al. Modelling protein docking using shape complementarity, electrostatics and biochemical information. *Journal of Molecular Biology*, 272(1):106–120, 1997.

[137] Z. Gao, Y. Chen, and L. Lian. Haptic aided soft-touch multi-material product design. In *Proceedings of 2006 IEEE International Conference on Virtual Environments, Human-Computer Interfaces, and Measurement Systems*, pages 36–41, 2006.

[138] Z. Gao and I. Gibson. Haptics-based 3D B-spline curve sketch using multi-resolution and two-handed techniques. *Proceedings of EuroHaptics 2004*, 2004.

[139] Z. Gao and I. Gibson. Haptic B-spline surface sculpting with a shaped tool of implicit surface. *Computer-Aided Design and Applications*, 2(1-4):263–272, 2005.

[140] Z. Gao and I. Gibson. A novel haptic interface and its application in CAD. *Virtual Modelling and Rapid Manufacturing*, pages 225–232, 2005.

[141] Z. Gao and I. Gibson. Haptic sculpting of multi-resolution B-spline surfaces with shaped tools. *Computer-Aided Design*, 38(6):661–676, 2006.

[142] Z. Gao and I. Gibson. A 6-DOF haptic interface and its applications in CAD. *International Journal of Computer Applications in Technology*, 30(3):163–170, 2007.

[143] S. Garbaya and U. Zaldivar-Colado. Modeling dynamic behavior of parts in virtual assembly environment. In *Proceedings of the 2009 ASME-AFM World Conference on Innovative Virtual Reality*, pages 89–99, 2009.

[144] J. Gargus, B. Kim, J. R. Rossignac, and Christopher D. Shaw. Finger sculpting with digital clay. GVU Technical Report No. GIT-GVU-02-22, 2002.

[145] N. Gavish, T. Gutiérrez, S. Webel, J. Rodríguez, M. Peveri, U. Bockholt, and F. Tecchia. Evaluating virtual reality and augmented reality training for industrial maintenance and assembly tasks. *Interactive Learning Environments*, 23(6):778–798, 2015.

[146] N. Gavish, T. Gutiérrez, S. Webel, J. Rodriguez, and F. Tecchia. Design guidelines for the development of virtual reality and augmented reality training systems for maintenance and assembly tasks. In *Proceedings of the International Conference (SKILLS'2010), BIO Web of Conferences*, volume 1, page 00029. EDP Sciences, 2011.

[147] R. Geraerts and M. H. Overmars. *Algorithmic Foundations of Robotics V*, chapter A Comparative Study of Probabilistic Roadmap Planners, pages 43–57. Springer Berlin Heidelberg, 2004.

[148] I. Gibson, Z. Gao, and I. Campbell. A comparative study of virtual prototyping and physical prototyping. *International Journal of Manufacturing Technology and Management*, 6(6):503–522, 2004.

[149] E. G. Gilbert and D. W. Johnson. Distance functions and their application to robot path planning in the presence of obstacles. *Journal of Robotics and Automation*, 1(1):21–30, 1985.

[150] A. Gomes de Sa and G. Zachmann. Virtual reality as a tool for verification of assembly and maintenance processes. *Computers and Graphics*, 23(3):389–403, 1999.

[151] G. Gonzalez-Badillo, H. I. Medellin-Castillo, , T. Lim, J. M. Ritchie, R. C. W. Sung, and S. Garbaya. A new methodology to evaluate the performance of physics simulation engines in haptic virtual assembly. *Assembly Automation*, 34(2):128–140, 2014.

[152] G. Gonzalez-Badillo, H. I. Medellin-Castillo, and T. Lim. Development of a haptic virtual reality system for assembly planning and evaluation. *Procedia Technology*, 7(0):265–272, 2013. 3rd Iberoamerican Conference on Electronics Engineering and Computer Science (CIIECC'2013).

[153] G. Gonzalez-Badillo, H. I. Medellin-Castillo, T. Lim, and V. E. E. López. Physics engines evaluation based on model representation analysis. In *Proceedings of the 2014 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE'2014)*, 2014.

[154] G. Gonzalez-Badillo, H. I. Medellin-Castillo, T. Lim, J. M. Ritchie, and S. Garbaya. The development of a physics and constraint-based haptic virtual assembly system. *Assembly Automation*, 34(1):41–55, 2014.

[155] S. Gottschalk, M. C. Lin, and D. Manocha. OBBTree: A hierarchical structure for rapid interference detection. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 171–180, New York, NY, USA, 1996.

[156] N. K. Govindaraju, D. Knott, N. Jain, I. Kabul, R. Tamstorf, R. Gayle, M. C. Lin, and D. Manocha. Interactive collision detection between deformable models using chromatic decomposition. *ACM Transactions on Graphics*, 24(3):991–999, 2005.

[157] N. K. Govindaraju, S. Redon, M. C. Lin, and D. Manocha. CULLIDE: Interactive collision detection between complex models in large environments using graphics hardware. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware (HWWS'2003)*, pages 25–32, 2003.

[158] D. T. Greenwood. *Principles of Dynamics*. Prentice-Hall Englewood Cliffs, 2nd edition edition, 1988.

[159] A. Gregory, M. C. Lin, S. Gottschalk, and R. Taylor. H-COLLIDE: A framework for fast and accurate collision detection for haptic interaction. In *Proceedings of the 1999 IEEE Virtual Reality Conference (VR'1999)*, pages 38–45, 1999.

[160] A. Gregory, M. C. Lin, S. Gottschalk, and R. Taylor. Fast and accurate collision detection for haptic interaction using a three degree-of-freedom force-feedback device. In *Proceedings of the ACM SIGGRAPH'2005 Courses*, 2005.

[161] X. Guo, J. Hua, and H. Qin. Touch-based haptics for interactive editing on point set surfaces. *IEEE Computer Graphics and Applications*, 24(6):31–39, 2004.

[162] A. Gupta and M. K. O'Malley. Design of a haptic arm exoskeleton for training and rehabilitation. *IEEE/ASME Transactions on Mechatronics*, 11(3):280–289, 2006.

[163] R. Gupta, T. Sheridan, and D. Whitney. Experiments using multimodal virtual environments in design for assembly analysis. *Presence: Teleoperators and Virtual Environments*, 6(3):318–338, 1997.

[164] R. Gupta, D. Whitney, and Zeltzer. Prototyping and design for assembly analysis using multimodal virtual environments. *Computer-Aided Design*, 29(8):585–597, 1997.

[165] R. Gupta and Zeltzer. Prototyping and design for assembly analysis using multimodal virtual environments. In *Proccedings of the 1995 ASME Computers in Engineering Conference and the Engineering Database Symposium*, 1995.

[166] T. Gutiérrez, N. Gavish, S. Webel, and F. Tcchia. *Multimodal Virtual Environments*, chapter Training Platforms for Industrial Maintenance and Assembly, pages 227–239. CRC Press, Taylor & Francis, 2012.

[167] T. Gutiérrez, J. Rodriguez, Y. Velaz, S. Casado, A. Suescun, and E. J. Sanchez. IMA-VR: A multimodal virtual training system for skills transfer in industrial maintenance and assembly tasks. In *Proceedings of the 2010 IEEE International Symposium on Robot and Human Interactive Communication*, pages 428–433, 2010.

[168] B. Hannaford and J. H. Ryu. Time-domain passivity control of haptic interfaces. *IEEE Transactions on Robotics and Automation*, 18(1):1–10, 2002.

[169] P. T. Harker and J.-S. Pang. Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications. *Mathematical Programming*, 48(1-3):161–220, 1990.

[170] S. Hasegawa and N. Fujii. Real-time rigid body simulation based on volumetric penalty method. In *Proceedings of the 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 326–332, March 2003.

[171] S. Hasegawa and M. Sato. Real-time rigid body simulation for haptic interactions based on contact volume of polygonal objects. *Computer Graphics Forum*, 23(3):529–538, 2004.

[172] S. Hassan and J. Yoon. Haptic based optimized path planning approach to virtual maintenance assembly/disassembly (MAD). In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2010)*, pages 1310–1315, 2010.

[173] S. Hassan and J. Yoon. Intensified haptic guidance for a 3D virtual maintenance environment optimized by an intelligent assembly planning algorithm. In *Proceedings of the 11th IEEE International Conference on Control, Automation, and Systems (ICCAS'2011)*, pages 676–679, 2011.

[174] S. Hassan and J. Yoon. Virtual maintenance system with a two-staged ant colony optimization algorithm. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA'2011)*, pages 931–936, 2011.

[175] S. Hassan and J. Yoon. Haptic assisted aircraft optimal assembly path planning scheme based on swarming and artificial potential field approach. *Advanced Engineering Software*, 69:18–25, 2014.

[176] V. Hayward and B. Armstrong. A new computational model of friction applied to haptic rendering. In *Experimental Robotics VI*, pages 403–412. Springer-Verlag, 2000.

[177] V. Hayward, O. R. Astley, M. C. Hernandez, D. Grant, and G. Robles-De-La-Torre. Haptic interfaces and devices. *Sensor Review*, 24(1):16–29, 2004.

[178] X. J. He and Y. H. Chen. A haptics-guided hole-filling system based on triangular mesh. *Computer-Aided Design and Applications*, 3(6):711–718, 2006.

[179] X. J. He and K. Choi. Stable haptic rendering for physics engines using inter-process communication andremote virtual coupling. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 4(1), 2013.

[180] D. Hecht, M. Reiner, and G. Halevy. Multimodal virtual environments: Response times, attention, and presence. *Presence: Teleoperators and Virtual Environments*, 15(5):515–523, 2006.

[181] Gentaro Hirota, Susan Fisher, and Ming Lin. Simulation of non-penetrating elastic bodies using distance fields. Technical report, University of North Carolina at Chapel Hill, 2000.

[182] C. H. Ho. *Computer Haptics: Rendering Rechniques for Force-Feedback in Virtual Environments*. Ph.D. dissertation, Massachusetts Institute of Technology (MIT), 2000.

[183] K. E. Hoff III, T. Culver, J. Keyser, M. Lin, and D. Manocha. Fast computation of generalized Voronoi diagrams using graphics hardware. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pages 277–286, New York, NY, USA, 1999.

[184] K. E. Hoff III, A. Zaferakis, M. C. Lin, and D. Manocha. Fast and simple 2D geometric proximity queries using graphics hardware. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics (I3D'2001)*, pages 145–148. ACM, 2001.

[185] K. E. Hoff III, A. Zaferakis, M. C. Lin, and D. Manocha. Fast 3D geometric proximity queries between rigid and deformable models using graphics hardware acceleration. UNC-CH Technical Report TR02-004, 2002.

[186] J. M. Hollerbach, E. C. Cohen, W. B. Thompson, R. Freier, D. Johnson, A. Nahvi, D. Nelson, T. V. Thompson II, and S. C. Jacobsen. Haptic interfacing for virtual prototyping of mechanical CAD designs. In *Proceedings of the 1997 ASME Design for Manufacturing Symposium*, 1997.

[187] J. M. Hollerbach, E. C. Cohen, W. B. Thompson, and S. C. Jacobsen. Rapid virtual prototyping of mechanical assemblies. In *Proceedings of the NSF Design and Manufacturing Grantees Conference*, 1996.

[188] B. M. Howard and J. M. Vance. Desktop haptic virtual assembly using physically based modelling. *Virtual Reality*, 11(4):207–215, 2007.

[189] K. Hrbacek and T. Jech. *Introduction to Set Theory*. Chapman & Hall/CRC Pure and Applied Mathematics. CRC Press, 1999.

[190] Y. Hu, D. L. Wu, X. Fan, and X. J. Zhen. Grid-enabled collaborative virtual assembly environment. *Assembly Automation*, 30(4):352–364, 2010.

[191] J. Hua and H. Qin. Dynamic implicit solids with constraints for haptic sculpting. In *Proceedings of the 2002 Shape Modeling International Conference*, pages 119–129, 2002.

[192] J. Hua and H. Qin. Haptics-based volumetric modeling using dynamic spline-based implicit functions. In *Proceedings of the 2002 IEEE Symposium on Volume Visualization and Graphics (VVS'2002)*. IEEE, 2002.

[193] J. Hua and H. Qin. Haptics-based dynamic implicit solid modeling. *IEEE Transactions on Visualization and Computer Graphics*, 10(5):574–586, 2004.

[194] P. M. Hubbard. Interactive collision detection. In *Proceedings of the 1993 IEEE Symposium on Research Frontiers in Virtual Reality*, pages 24–31, 1993.

[195] P. M. Hubbard. Collision detection for interactive graphics applications. *IEEE Transactions on Visualization and Computer Graphics*, 1(3):218–230, 1995.

[196] P. M. Hubbard. Approximating polyhedra with spheres for time-critical collision detection. *ACM Transac*, 15(3):179–210, 1996.

[197] T. C. Hudson, M. C. Lin, J. Cohen, S. Gottschalk, and D. Manocha. V-COLLIDE: Accelerated collision detection for VRML. In *Proceedings of the 2nd Symposium on Virtual Reality Modeling Language*, New York, NY, USA, 1997.

[198] R. Iacob, P. Mitrouchev, and J. C. Léon. A simulation framework for assembly/disassembly process modeling. In *Proceedings of the 2007 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE'2007)*, number DETC2007-34804, pages 1017–1027. American Society of Mechanical Engineers (ASME), 2007.

[199] R. Iacob, P. Mitrouchev, and J. C. Leon. Contact identification for assembly-disassembly simulation with a haptic device. *The Visual Computer*, 24(11):973–979, 2008.

[200] R. Iacob, P. Mitrouchev, and J. C. Leon. Assembly simulation incorporating component mobility modelling based on functional surfaces. *International Journal on Interactive Design and Manufacturing*, 5(2):119–132, 2011.

[201] R. Iacob and D. Popescu. Generation of disassembly directions based on component mobility. *Studies in Informatics and Control*, pages 1220–1766, 2013.

[202] R. Iacob, D. Popescu, and P. Mitrouchev. Assembly/disassembly analysis and modeling techniques: A review. *Strojniški vestnik-Journal of Mechanical Engineering*, 58(11):653–664, 2012.

[203] R. Iglesias, S. Carrillo, S. Casado, T. Gutiérrez, and J. I. Barbero. Virtual assembly simulation in a distributed haptic virtual environment. In *Proceedings of the International Conference on Advanced Design and Manufacture*, 2006.

[204] R. Iglesias, S. Casado, T. Gutiérrez, A. García-Alonso, K. M. Yap, W. Yu, and A. Marshall. A peer-to-peer architecture for collaborative haptic assembly. In *Proceedings of the 10th International Symposium on Distributed Simulation and Real-Time Applications (DS-RT'2006)*, pages 25–34, 2006.

[205] R. Iglesias, S. Casado, T. Gutiérrez, A. García-Alonso, W. Yu, and A. Marshall. Simultaneous remote haptic collaboration for assembling tasks. *Multimedia Systems*, 13(4):263–274, 2008.

[206] R. Iglesias, E. Prada, A. Uribe, A. García-Alonso, S. Casado, and T. Gutiérrez. Assembly simulation on collaborative haptic virtual environments. In *Proceedings of the 15th International Conference in Central Europe on Computer Graphics*, 2007.

[207] G. Irving, J. Teran, and R. Fedkiw. Invertible finite elements for robust simulation of large deformation. In *Proceedings of the ACM SIGGRAPH'2004/Eurographics Symposium on Computer Animation*, pages 131–140, Aire-la-Ville, Switzerland, Switzerland, 2004.

[208] H. Iwata and H. Noma. Volume haptization. In *Proceedings of the 1993 IEEE 1993 Symposium on Research Frontiers in Virtual Reality*, pages 16–23, 1993.

[209] W. Jacobi. Halbleiterverstärker, 1952.

[210] D. L. James and D. K. Pai. BD-tree: Output-sensitive collision detection for reduced deformable models. *ACM Transactions on Graphics (TOG) and Proceedings of the ACM SIGGRAPH'2004*, 23(3):393–398, 2004.

[211] F. Janabi-Sharifi, V. Hayward, and C. S. J. Chen. Discrete-time adaptive windowing for velocity estimation. *IEEE Transactions on Control Systems Technology*, 8(6):1003–1009, 2000.

[212] S. Jayaram, H. I. Connacher, and K. W. Lyons. Virtual assembly using virtual reality techniques. *Computer-Aided Design*, 29(8):575–584, 1997.

[213] S. Jayaram, U. Jayaram, Y. J. Kim, C. DeChenne, K. Lyons, K. W. andLyons, and T. Mitsui. Industry case studies in the use of immersive virtual assembly. *Virtual Reality*, 11(4):217–228, 2007.

[214] S. Jayaram, U. Jayaram, H. Tirumali, and K. Lyons. A tool/part/human interaction model for assembly in virual environment. 2000.

[215] S. Jayaram, U. Jayaram, Y. Wang, H. Tirumali, K. Lyons, and P. Hart. VADE: A virtual assembly design environment. *Journal of Computer Graphics and Applications*, 19(6):44–50, 1999.

[216] S. Jayaram, J. . Vance, J. M. aVance, U. Jayaram, and H Srinivasan. Assessment of VR technology and its applications to engineering problems. *Journal of Computing and Information Science in Engineering*, 1(1):72–83, 2001.

[217] S. Jayaram, Y. Wang, U. Jayaram, K. Lyons, and P. Hart. A virtual assembly design environment. In *Proceedings of the IEEE Virtual Reality Annual International Symposium (VRAIS'1999)*, pages 172–179, 1999.

[218] U. Jayaram, S. Jayaram, C. DeChenne, Y. J. Kim, C. Palmer, and T. Mitsui. Case studies using immersive virtual assembly in industry. In *Proceedings of the 2004 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE'2004)*, pages 627–636, 2004.

[219] U. Jayaram, S. Jayaram, I. Shaikh, Y. J. Kim, and Palmer. Introducing quantitative analysis methods into virtual environments for real-time assembly and continuous ergonomic evaluations. *Computers in Industry*, 57(3):283–296, 2006.

[220] U. Jayaram, S. Jayaram, Y. Yang, and K. Lyons. CORBA-based collaboration in a virtual assembly design environment. In *Proceedings of the ASME Design Engineering Technical Conferences (DETC'2000)*, number DETC2000/CIE-14585. American Society of Mechanical Engineers (ASME), 2000.

[221] D. Jia, A. Bhatti, and S. Nahavandi. Design and evaluation of a haptically enable virtual environment for object assembly training. In *Proceedings of the 2009 IEEE International Workshop on Haptic Audio visual Environments and Games (HAVE'2009)*, pages 75–80, 2009.

[222] P. Jimenez, F. Thomas, and Torras. 3D collision detection: A survey. *Computers and Graphics*, 25(2):269–285, 2001.

[223] D. C. Johnson and J. M. Vance. The use of the Voxmap PointShell method of collision detection in virtual assembly methods planning. In *Proceedings of the 2001 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE'2001)*, 2001.

[224] H. Kang and A. Kak. Deforming virtual objects interactively in accordance with an elastic model. *Computer-Aided Design*, 28(4):251–262, 1996.

[225] E. Katchalski-Katzir, I. Shariv, M. Eisenstein, A. A Friesem, C. Aflalo, and I. A. Vakser. Molecular surface recognition: Determination of geometric fit between proteins and their ligands by correlation techniques. *Proceedings of the National Academy of Sciences*, 89(6):2195–2199, 1992.

[226] Y. Katznelson. *An Introduction to Harmonic Analysis*. Cambridge University Press, 3rd edition, 2004.

[227] L. Kavan and J. Žára. Fast collision detection for skeletally deformable models. In *Computer Graphics Forum*, volume 24, pages 363–372. Wiley Online Library, 2005.

[228] L. E. Kavraki. Computation of configuration-space obstacles using the fast Fourier transform. *IEEE Transactions on Robotics and Automation*, 11(3):408–413, 1995.

[229] L. E. Kavraki, M. N. Kolountzakis, and J. C. Latombe. Analysis of probabilistic roadmaps for path planning. *IEEE Transactions on Robotics and Automation*, 14(1):166–171, 1998.

[230] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

[231] C. E. Kim and J. M. Vance. Using VPS (Voxmap PointShell) as the basis for interaction in a virtual assembly environment. In *Proceedings of the 2003 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE'2003)*, pages 1153–1161, 2003.

[232] C. E. Kim and J. M. Vance. Collision detection and part interaction modeling to facilitate immersive virtual assembly methods. *Journal of Computing and Information Science in Engineering*, 4(2):83–90, 2004.

[233] C. E. Kim and J. M. Vance. Development of a networked haptic environment in VR to facilitate collaborative design using Voxmap PointShell (VPS) software. In *Proceedings of the 2004 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE'2004)*, number DETC2004-57648, pages 19–25. American Society of Mechanical Engineers (ASME), 2004.

[234] K. Kim and J. Park. Virtual bone drilling for dental implant surgery training. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology (VRST'2009)*, pages 91–94, New York, NY, USA, 2009.

[235] Y. J. Kim, M. C. Lin, and D. Manocha. DEEP: Dual-space expansion for estimating penetration depth between convex polytopes. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA'2002)*, volume 1, pages 921–926, 2002.

[236] F. Klein. A new approach to point membership classification in B-rep solids. In *Mathematics of Surfaces XIII*, pages 235–250. Springer Berlin Heidelberg, 2009.

[237] J. T. Klosowski, M. Held, J. S. B. Mitchell, H. Sowizral, and K. Zikan. Efficient collision detection using bounding volume hierarchies of k-DOPs. *IEEE Transactions on Acoustics Speech and Signal ProcessingVisualization and Computer Graphics*, 4(1):21–36, 1998.

[238] S. Kockara, T. Halic, K. Iqbal, C. Bayrak, and R. Rowe. Collision detection: A survey. In *Proceedings of the 2007 IEEE International Conference on Systems, Man and Cybernetics (ISIC'2007)*, pages 4046–4051, 2007.

[239] J. A. Kovacs, P. Chacon, Y. Cong, E. Metwally, and W. Wriggers. Fast rotational matching of rigid bodies by fast Fouriertransform acceleration of five degrees of freedom. *Acta Crystallographica Section D: Biological Crystallography*, 59(8):1371–1376, 2003.

[240] D. Kozakov, R. Brenke, S. R. Comeau, and S. Vajda. PIPER: An FFT-based protein docking program with pairwise potentials. *Proteins: Structure, Function, and Bioinformatics*, 65(2):392–406, 2006.

[241] D. Kubus, I. Weidauer, and F. M. Wahl. 1kHz is not enough; how to achieve higher update rates with a bilateral teleoperation system based on commercial hardware. In *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2009)*, pages 5107–5114, 2009.

[242] S. Kunis and S. Kunis. The nonequispaced FFT on graphics processing units. *Proceedings in Applied Mathematics and Mechanics (PAMM)*, 12(1):7–10, 2012.

[243] J. Kuriyan, B. Konforti, and D. Wemmer. *The Molecules of Life: Physical and Chemical Principles*. Garland Science, 2012.

[244] N. Kusumoto, T. Sohmura, S. Yamada, K. Wakabayashi, T. Nakamura, and H. Yatani. Application of virtual reality force feedback haptic device for oral implant surgery. *Clinical Oral Implants Research*, 17(6):708–713, 2006.

[245] N. Ladeveze, J. Y. Fourquet, and B. Puel. Interactive path planning for haptic assistance in assembly tasks. *Computers & Graphics*, 34(1):17–25, 2010.

[246] N. Ladeveze, J. Y. Fourquet, B. Puel, and M. Taix. Haptic assembly and disassembly task assistance using interactive path planning. In *Proceedings of the 2009 IEEE Virtual Reality Conference (VR'2009)*, pages 19–25, 2009.

[247] A. R. Lanfranco, A. E. Castellanos, J. P. Desai, and W. C. Meyers. Robotic surgery: A current perspective. *Annals of Surgery*, 239(1):14–21, 2004.

[248] E. Larsen, S. Gottschalk, M. C. Lin, and D. Manocha. Fast distance queries with rectangular swept sphere volumes. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA'2000)*, volume 4, pages 3719–3726, 2000.

[249] S. D. Laycock and A. M. Day. Recent developments and applications of haptic devices. *Computer Graphics Forum*, 22(2):117–132, 2003.

[250] D. Leithinger, S. Follmer, A. Olwal, and H. Ishii. Physical telepresence: Shape capture and display for embodied, computer-mediated remote collaboration. In *Proceedings of the 27th annual ACM symposium on User Interface Software and Technology*, pages 461–470, 2014.

[251] D. Leithinger, S. Follmer, A. Olwal, and H. Ishii. Shape displays: Spatial interaction with dynamic physical form. *IEEE Computer Graphics and Applications*, 35(5):5–11, 2015.

[252] M. C. Leu, H. A. ElMaraghy, A. Y. C. Nee, S. K. Ong, M. Lanzetta, M. Putz, W. Zhu, and A. Bernard. CAD model based virtual assembly simulation, planning and training. *CIRP Annals - Manufacturing Technology*, 62(2):799–822, 2013.

[253] A. Lieutier. Any open bounded subset of $\mathbb{R}^n$ has the same homotopy type as its medial axis. *Computer-Aided Design*, 36(11):1029–1046, 2004.

[254] T. Lim, M. Calis, J. M. Ritchie, J. R. Corney, R. G.e Dewar, and M. P. Desmulliez. A haptic assembly, machining and manufacturing system (HAMMS) approach. In *Proceedings of the 1st International Virtual Manufacturing Workshop (VirMan'2006)*, 2006.

[255] T. Lim, R. Dewar, M. Calis, J. M. Ritchie, J. R. Corney, and M. Desmulliez. A structural assessment of haptic-based assembly processes. In *Proccedings of the 1st International Virtual Manufacturing Workshop (VirMan'2006)*, volume 26, 2006.

[256] T. Lim, J. M. Ritchie, J. R. Corney, R. G. Dewar, K. Schmidt, and K. Bergsteiner. Assessment of a haptic virtual assembly system that uses physics-based interactions. In *Proceedings of the 2007 IEEE International Symposium on Assembly and Manufacturing (ISAM'2007)*, pages 147–153, 2007.

[257] T. Lim, J. M. Ritchie, R. Sung, Z. Kosmadoudi, Y. Liu, and A. G. Thin. *Haptic Virtual Reality Assembly - Moving Towards Real Engineering Applications*, chapter Advances in Haptics, pages 693–722. InTech, 2010.

[258] T. Lim, J.M. Ritchie, R.G. Dewar, J.R. Corney, P. Wilkinson, M. Calis, M. Desmulliez, and J.-J. Fang. Factors affecting user performance in haptic assembly. *Virtual Reality*, 11(4):241–252, 2007.

[259] M. Lin and S. Gottschalk. Collision detection between geometric models: A survey. In *Proceedings of the 1998 IMA Conference on Mathematics of Surfaces*, volume 1, pages 37–56, 1998.

[260] X. Liu. Designing B-spline surfaces with haptics based on energy minimization method. In *Proceedings of the 2012 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE'2012)*, pages 1509–1516. American Society of Mechanical Engineers, 2012.

[261] X. Liu. Filling N-sided holes using trimmed B-spline surfaces. In *Proceedings of the 2012 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE'2012)*, number DETC2012-70735, pages 611–618. American Society of Mechanical Engineers, 2012.

[262] X. Liu. Designing B-spline surfaces with haptics based on variational technique. *Computer-Aided Design and Applications*, 11(4):369–380, 2014.

[263] X. Liu, G. Dodds, B. K. Hinds, and J. McCartney. Virtual DesignWorks: Designing 3D CAD models via touch interaction. In *Proceedings of the 2003 ASME International Mechanical Engineering Congress and Exposition*, pages 39–45. American Society of Mechanical Engineers, 2003.

[264] X. Liu, G. Dodds, J. McCartney, and B. K. Hinds. Design and deformation of CAD surface models with haptics. In *Proceedings of the 2004 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE'2004)*, pages 883–890. American Society of Mechanical Engineers, 2004.

[265] X. Liu, G. Dodds, J. McCartney, and B. K. Hinds. Virtual DesignWorks–designing 3D CAD models via haptic interaction. *Computer-Aided Design*, 36(12):1129–1140, 2004.

[266] X. Liu, G. Dodds, J. McCartney, and B. K. Hinds. Manipulation of CAD surface models with haptics based on shape control functions. *Computer-Aided Design*, 37(14):1447–1458, 2005.

[267] Xiaodong Liu. Filling N-sided holes with trimmed B-spline surfaces based on energy-minimization method. *Journal of Computing and Information Science in Engineering (JCISE)*, 15(1):011001:1–9, 2015.

[268] P. Lötstedt. Numerical simulation of time-dependent contact and friction problems in rigid body mechanics. *SIAM Journal on Scientific and Statistical Computing*, 5(2):370–393, 1984.

[269] T. Lozano-Perez. Spatial planning: A configuration space approach. *IEEE Transactions on Computers*, C-32(2):108–120, 1983.

[270] M. Lysenko. Fourier collision detection. *International Journal of Robotics Research*, 32(4):483–503, 2013.

[271] M. Lysenko, S. Nelaturi, and V. Shapiro. Group morphology with convolution algebras. In *Proceedings of the 2010 ACM Symposium on Solid and Physical Modeling (SPM'2010)*, pages 11–22, New York, NY, USA, 2010.

[272] M. Lysenko, V. Shapiro, and Nelaturi. Non-commutative morphology: Shapes, filters, and convolutions. *Computer Aided Geometric Design*, 28(8):497–522, 2011.

[273] G. Macindoe, L. Mavridis, V. Venkatraman, M. D. Devignes, and D. W. Ritchie. HexServer: an FFT-based protein docking server powered by graphics processors. *Nucleic Acids Research*, 38(Web Server Issue):W445–W449, 2010.

[274] H. Maekawa and J. M. Hollerbach. Haptic display for object grasping and manipulating in virtual environment. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation (ICR'1998)*, volume 3, pages 2566–2573 vol.3, May 1998.

[275] L. Marcelino, N. Murray, and T. Fernando. A constraint manager to support virtual maintainability. *Computers and Graphics*, 27(1):19–26, 2003.

[276] D. W. Marhefka and D. E. Orin. Simulation of contact using a nonlinear damping model. In *Proceeding of the 1996 IEEE International Conference on Robotics and Automation*, volume 2, pages 1662–1668, 1996.

[277] T. H. Massie. *Initial Haptic Explorations with the Phantom: Virtual Touch through Point Interaction*. PhD thesis, Massachusetts Institute of Technology (MIT), 1996.

[278] L. McAtamney and E. N. Corlett. RULA: A survey method for the investigation of work-related upper limb disorders. *Applied Ergonomics*, 24(2):91–99, 1993.

[279] S. D. McDermott and B. Bras. Development of a haptically enabled dis/re-assembly simulation environment. In *Proceedings of the 1999 ASME Design Engineering Technical Conferences (DETC'1999)*, 1999.

[280] K. T. McDonnell. *Dynamic Subdivision-based Solid Modeling*. Research proficiency examination, State University of New York at Stony Brook, 2000.

[281] K. T. McDonnell, Y. S. Chang, and H. Qin. DigitalSculpture: a subdivision-based approach to interactive implicit surface modeling. *Graphical Models*, 67(4):34–369, 2005.

[282] K. T. McDonnell and H. Qin. Virtual clay: Haptics-based deformable solids of arbitrary topology. In F. Perales and E. R. Hancock, editors, *Articulated Motion and Deformable Objects*, volume 2492 of *Lecture Notes in Computer Science*, pages 1–20. Springer Berlin Heidelberg, 2002.

[283] K. T. McDonnell and H. Qin. A novel framework for physically based sculpting and animation of free-form solids. *The Visual Computer*, 23(4):285–296, 2007.

[284] K. T. McDonnell, H. Qin, and R. A. Wlodarczyk. Virtual clay: A real-time, haptics-based sculpting system. Technical report, State University of New York at Stony Brook, 2001.

[285] K. T. McDonnell, H. Qin, and R. A. Wlodarczyk. Virtual clay: A real-time sculpting system with haptic toolkits. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, pages 179–190, 2001.

[286] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy. Advances in voxel-based 6-DOF haptic rendering. In *Proceedings of the ACM SIGGRAPH'2005 Courses*, 2005.

[287] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy. Six degree-of-freedom haptic rendering using voxel sampling. In *Proceedings of the ACM SIGGRAPH'2005 Courses*, New York, NY, USA, 2005.

[288] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy. Voxel-based 6-DOF haptic rendering improvements. *Haptics-e*, 3(7), 2006.

[289] M. E. Merchant. Mechanics of the metal cutting process i. orthogonal cutting and a type 2 chip. *Journal of Applied Physics*, 16(5):267–275, 1945.

[290] B. Mirtich. V-Clip: Fast and robust polyhedral collision detection. *ACM Transactions on Graphics (TOG)*, 17(3):177–208, 1998.

[291] B. Mirtich and J. Canny. Impulse-based simulation of rigid bodies. In *Proceedings of the 1995 Symposium on Interactive 3D Graphics (I3D'1995)*, pages 181–217, New York, NY, USA, 1995.

[292] G. Moreau, P. Fuchs, and P. Stergiopoulos. Applications of virtual reality in the manufacturing industry: From design review to ergonomic studies. *Mécanique and Industries*, 5:171–179, 2004.

[293] J. J. Moreau. Standard inelastic shocks and the dynamics of unilateral constraints. In G. Del Piero and F. Maceri, editors, *Unilateral Problems in Structural Analysis*, volume 288, pages 173–221. Springer Vienna, 1985.

[294] D. Morris. Algorithms and data structures for haptic rendering: Curve constraints, distance maps, and data logging. Technical report, Standford University, 2006.

[295] D. Morris, N. Joshi, and K. Salisbury. Haptic Battle Pong: High-degree-of-freedom haptics in a multiplayer gaming environment. In *Experimental Gameplay Workshop (GDC)*, 2004.

[296] D. Morris, Hong Tan, F. Barbagli, T. Chang, and K. Salisbury. Haptic feedback enhances force skill learning. In *Proceedings of the 2007 Joint Conference on World Haptics, the Eurohaptics 2005, and the Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 21–26, March 2007.

[297] T. S. Mujber, T. Szecsi, and M. S. J. Hashmi. Virtual reality applications in manufacturing process simulation. *Journal of Materials Processing Technology*, 155156:1834–1838, 2004. Proceedings of the International Conference on Advances in Materials and Processing Technologies: Part 2.

[298] K. Murakami, R. Kiyama, T. Narumi, T. Tanikawa, and M. Hirose. Poster: A wearable augmented reality system with haptic feedback and its performance in virtual assembly tasks. In *Proccedings of the 2013 IEEE Symposium on 3D User Interfaces (3DUI'2013)*, pages 161–162, 2013.

[299] N. Murray and T. Fernando. An immersive assembly and maintenance simulation environment. In *International Conference on Advanced Research in Virtual and Rapid Prototyping*, 2003.

[300] N. Murray and T. Fernando. An immersive assembly and maintenance simulation environment. In *Proceedings of the 2004 IEEE International Symposium on Distributed Simulation and Real-Time Applications (DS-RT'2004)*, pages 159–166, October 2004.

[301] A. Nahvi, D. D. Nelson, J. M. Hollerbach, and D. E. Johnson. Haptic manipulation of virtual mechanisms from mechanical CAD designs. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation (ICR'1998)*, volume 1, pages 375–380, 1998.

[302] S. Nelaturi. *Configuration Modeling*. Ph.D. dissertation, University of Wisconsin-Madison, 2012.

[303] S. Nelaturi and V. Shapiro. Configuration products and quotients in geometric modeling. In *The 2009 SIAM/ACM Joint Conference on Solid and Physical Modeling (SPM'2009)*, 2009.

[304] S. Nelaturi and V. Shapiro. Configuration products and quotients in geometric modeling. *Computer-Aided Design*, 43(7):781–794, 2011. The 2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling.

[305] M. Nießner, C. Siegl, H. Schäfer, and C. T. Loop. Real-time collision detection for dynamic hardware tessellated objects. In *Eurographics (Short Papers)*, pages 33–36, 2013.

[306] M. O'Connor, V. Srinivasan, and A. Jones. Connected Lie and symmetry subgroups of the rigid motions: Foundations and classification. Research Report 20512, IBM, 1996.

[307] Staff of the Production Automation Project. An introduction to PADL. Production Automation Project, Technical Memo. No. 22 (TM-22), University of Rochester, 1974.

[308] T. Ohtani, N. Kusumoto, K. Wakabayashi, S. Yamada, T. Nakamura, Y. Kumazawa, H. Yatani, and Sohmura. Application of haptic device to implant dentistry–accuracy verification of drilling into a pig bone. *Dental Materials Journal*, 28(1):75–81, 2009.

[309] A. M. Okamura. Methods for haptic feedback in teleoperated robot-assisted surgery. *Industrial Robot: An International Journal*, 31(6):499–508, 2004.

[310] A. M. Okamura. Haptic feedback in robot-assisted minimally invasive surgery. *Current Opinion in Urology*, 19(1):102–107, 2009.

[311] P. Olsson, f. Nysjo, J. M. Hirsch, and I. Carlbom. Snap-to-Fit, a haptic 6 DOF alignment tool for virtual assembly. In *Proceedings of the 2013 World Haptics Conference (WHC'2013)*, pages 205–210, 2013.

[312] C. O'Sullivan and J. Dingliana. Real-time collision detection and response using sphere-trees. In *Spring Conference on Computer Graphics*, pages 83–92, 1999.

[313] I. J. Palmer and R. L. Grimsdale. Collision detection for animation using sphere-trees. *Computer Graphics Forum*, 14(2):105–116, 1995.

[314] C. Pan and S. Smith. Case study: The impact of assembly reorientations on assembly time. *International Journal of Production Research*, 44(21):4569–4585, 2006.

[315] M. Paterson. *The Senses of Touch: Haptics, Affects and Technologies*. Bloomsbury Academic, 2007.

[316] J. Perret, C. Kneschke, J. M. Vance, and G. Dumont. Interactive assembly simulation with haptic feedback. *Assembly Automation*, 33(3):214–220, 2013.

[317] G. A. Petsko and D. Ringe. *Protein Structure and Function*. New Science Press, 2004.

[318] G. Picinbono, H. Delingette, and N. Ayache. Non-linear anisotropic elasticity for real-time surgery simulation. *Graphical Models, Special Issue on SMI'2002*, 65(5):305–321, 2003.

[319] F. Picon, M. Ammi, and P. Bourdot. Case study of haptic methods for selection on CAD models. In *Proceedings of the 2008 IEEE Virtual Reality Conference (VR'2008)*, pages 209–212, 2008.

[320] F. Picon, M. Ammi, and P. Bourdot. Force model for CAD selection. In *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology (VRST'2008)*, pages 283–284, 2008.

[321] F. Picon, M. Ammi, and P.k Bourdot. Haptically-aided extrusion for object edition in CAD. In M. Ferre, editor, *Haptics: Perception, Devices and Scenarios*, volume 5024 of *Lecture Notes in Computer Science*, pages 736–741. Springer Berlin Heidelberg, 2008.

[322] D. Popescu, R. Iacob, F. Noel, C. Masclet, and T. Louis. Evaluation of a haptic environment for assembly task simulation. *Studies in Informatics and Control*, 24(3):329–338, 2015.

[323] D. Potts, G. Steidl, and M. Tasche. Fast Fourier transforms for nonequispaced data: A tutorial. In *Modern Sampling Theory*, pages 247–270. Springer-Verlag, 2001.

[324] S. Quinlan. Efficient distance computation between non-convex objects. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation (ICR'1994)*, volume 4, pages 3324–3329, 1994.

[325] S. Redon, A. Kheddar, and S. Coquillart. Fast continuous collision detection between rigid bodies. *Computer Graphics Forum*, 21(3):279–287, 2002.

[326] S. Redon, A. Kheddar, and S. Coquillart. Gauss' least constraints principle and rigid body simulations. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICR'2002)*, volume 1, pages 517–522, 2002.

[327] S. Redon, Y. J. Kim, M. C. Lin, and D. Manocha. Fast continuous collision detection for articulated models. In *Proceedings of the 9th ACM Symposium on Solid Modeling and Applications*, pages 145–156, 2004.

[328] S. Redon, M. C. Lin, D. Manocha, and Y. J. Kim. Fast continuous collision detection for articulated models. *Journal of Computing and Information Science in Engineering*, 5(2):126–137, 2005.

[329] M. Renouf, V. Acary, and G. Dumont. 3D frictional contact and impact multibody dynamics. a comparison of algorithms suitable for real-time applications. In *ECCOMAS Thematic Conference on Mutlibody Dynamics*, 2005.

[330] M. Renz, C. Preusche, M. Pötke, H. P. Kriegel, and G. Hirzinger. Stable haptic interaction with virtual environments using an adapted Voxmap-PointShell algorithm. In *Proceedings of the Eurohaptics 2001*, 2001.

[331] A. A. G. Requicha. Mathematical foundations of constuctive solid geometry: General topology of closed regular sets. Production Automation Project, Technical Memo. No. 27 (TM-27), University of Rochester, 1978.

[332] A. A. G. Requicha. Mathematical models of rigid solid objects. Production Automation Project, Technical Memo. No. 28 (TM-28), University of Rochester, 1980.

[333] A. A. G. Requicha. Representations for rigid solids: Theory, methods, and systems. *Journal of ACM Computational Surveys*, 12(4):437–464, 1980.

[334] A. A. G. Requicha. Representations of rigid solid objects. Production Automation Project, Technical Memo. No. 29 (TM-29), University of Rochester, 1980.

[335] A. A. G. Requicha, N. M. Samuel, and H. B. Voelcker. Part and assembly description languages–II. Production Automation Project, Technical Memo. No. 20a (TM-20a), University of Rochester, 1974.

[336] F. Reuleaux and E. S. Ferguson. *Kinematics of Machinery: Outlines of a Theory of Machines.* Dover Publications, 2012.

[337] A. Ricci. A constructive geometry for computer graphics. *The Computer Journal*, 16(2):157–160, 1973.

[338] D. Ritchie. Recent progress and future directions in protein-protein docking. *Current Protein and Peptide Science*, 9(1):1–15, 2008.

[339] J. M. Ritchie, T. Lim, H. Medellin, and R. S. Sung. A haptic based virtual assembly system for the generation of assembly process plans. In *Memorias Del XV Congreso Internacional Anual De La Somim*, pages 23–25, 2009.

[340] J. M. Ritchie, T. Lim, R. S. Sung, J. R. Corney, and H. Rea. The analysis of design and manufacturing tasks using haptic and immersive VR–some case studies. In D. Talaba and A. Amditis, editors, *Product Engineering*, pages 507–522. Springer Netherlands, 2008.

[341] J. M. Ritchie, T. Lim, R. S. Sung, and H. Medellin. Generation of assembly process plans and associated Gilbreth motion study data. In *Proceedings of the 2nd Virtual Manufacturing Workshop, EU Intuition VR Network Conference Intuition*, pages 6–8, 2008.

[342] J. B. T. M. Roerdink. Group morphology. *Pattern Recognition*, 33(6):877–895, 2000.

[343] L. B. Rosenberg. Virtual fixtures: Perceptual tools for telerobotic manipulation. In *Proceedings of the 1993 IEEE Virtual Reality Annual International Symposium*, pages 76–82, 1993.

[344] L. B. Rosenberg. *Virtual Fixtures*. PhD thesis, Stanford University, 1994.

[345] J. R. Rossignac, M. Allen, W. J. Book, A. Glezer, I. Ebert-Uphoff, C. Shaw, D. Rosen, S. Askins, J. Bai, P. Bosscher, J. Gargus, M. Kim, I. Llamas, A. Nguyen, G. Yuan, and H. Zhu. Finger sculpting with digital clay: 3D shape input and output through a computer-controlled real surface. In *Shape Modeling International, 2003*, pages 229–231. IEEE, 2003.

[346] E. Ruffaldi, D. Morris, F. Barbagli, K. Salisbury, and M. Bergamasco. Voxel-based haptic rendering using implicit sphere trees. In *Proceedings of the 16th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 319–325, 2008.

[347] M. Sagardia, T. Hulin, C. Preusche, and Hirzinger. Improvements of the Voxmap-PointShell algorithm–fast generation of haptic data-structures. In *53rd Internationales Wissenschaftliches Kolloquium Technische Universitt Ilmenau*, 2008.

[348] M. Sagardia, T. Stouraitis, and J. L. e Silva. A new fast and robust collision detection and force computation algorithm applied to the physics engine Bullet: Method, integration, and evaluation. In *Proceedings of the 2014 Conference and Exhibition of the European Association of Virtual and Augmented Reality (EuroVR'2014)*, 2014.

[349] S. E. Salcudean and T. D. Vlaar. On the emulation of stiff walls and static friction with a magnetically levitated input/output device. *Journal of Dynamic Systems, Measurement, and Control (JDSMC)*, 119(1):127–132, 1997.

[350] J. K. Salisbury and M. A. Srinivasan. Phantom-based haptic interaction with virtual objects. *IEEE Computer Graphics and Applications*, 17(5):6–10, 1997.

[351] G. Sankaranarayanan, S. Weghorst, M. Sanner, A. Gillet, and A. Olson. Role of haptics in teaching structural molecular biology. In *Proceedings of the 11th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 363–366, 2003.

[352] J. Sauer and E. Schömer. A constraint-based approach to rigid body dynamics for virtual reality applications. In *Proceedings of the 5th ACM Symposium on Virtual Reality Software and Technology (VRST'1998)*, pages 153–162, New York, NY, USA, 1998.

[353] J. Savall, D. Borro, J. J. Gil, and L. Matey. Description of a haptic system for virtual maintainability in aeronautics. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2887–2892, 2002.

[354] R. R. Schaller. Moore's law: Past, present and future. *IEEE Spectrum*, 34(6):52–59, 1997.

[355] B. Schling. *The Boost C++ Libraries*. XML Press, 2011.

[356] J. Schöberl. NETGEN: An advancing front 2D/3D-mesh generator based on abstract rules. *Computing and Visualization in Science*, 1(1):41–52, 1997.

[357] J. T. Schwartz and M. Sharir. On the "piano movers" problem i. the case of a two-dimensional rigid polygonal body moving amidst polygonal barriers. *Communications on Pure and Applied Mathematics*, 36(3):345–398, 1983.

[358] J. T. Schwartz and M. Sharir. On the "piano movers" problem. ii. general techniques for computing topological properties of real algebraic manifolds. *Advances in Applied Mathematics*, 4(3):298–351, 1983.

[359] J. T. Schwartz and M. Sharir. On the "piano movers" problem: Iii. coordinating the motion of several independent bodies: The special case of circular bodies moving amidst polygonal barriers. *The International Journal of Robotics Research*, 2(3):46–75, 1983.

[360] ELI Educational Series. 7 things you should know about haptics. Technical report, Educause Learning Initiative (ELI), September 2007.

[361] A. Seth, H. J. Su, and J. M. Vance. A desktop networked haptic VR interface for mechanical assembly. In *Proceedings of the 2005 ASME International Mechanical Engineering Congress and Exposition*, pages 173–180, 2005.

[362] A. Seth, H. J. Su, and J. M. Vance. SHARP: A system for haptic assembly and realistic prototyping. In *Proceedings of the 2006 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE'2006)*, pages 905–912, 2006.

[363] A. Seth, H. J. Su, and J. M. Vance. Development of a dual-handed haptic assembly system: SHARP. *Journal of Computing and Information Science in Engineering*, 8(4):1–8, 2008.

[364] A. Seth, J. M. Vance, and J. H. Oliver. Combining geometric constraints with physics modeling for virtual assembly using SHARP. In *Proceedings of the 2007 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE'2007)*, 2007.

[365] A. Seth, J. M. Vance, and J. H. Oliver. Combining dynamic modeling with geometric constraint management to support low clearance virtual manual assembly. *Journal of Mechanical Design*, 132(8):081002–1–7, 2010.

[366] A. Seth, J. M. Vance, and J. H. Oliver. Virtual reality for assembly methods prototyping: A review. *Virtual Reality*, 15(1):5–20, 2011.

[367] I. Shaikh, U. Jayaram, S. Jayaram, and C. Palmer. Participatory ergonomics using VR integrated with analysis tools. In *Proceedings of the 2004 Winter Simulation Conference*, volume 2, pages 1746–1754, 2004.

[368] I. Shaikh, Y. J. Kim, S. Jayaram, U. Jayaram, and H. Choi. Integration of immersive environment and RULA for real-time study of workplace related musculoskeletal disorders in the upper limb. In *Proceedings of the 2003 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE'2003)*, pages 1163–1171. American Society of Mechanical Engineers (ASME), 2003.

[369] C. Shukla, M. Vazquez, and Chen. F. F. Virtual manufacturing: An overview. *Computers & Industrial Engineering*, 31(1-2):79–82, 1996. Proceedings of the 19th International Conference on Computers and Industrial Engineering.

[370] J. Simard, M. Ammi, F. Picon, and P. Bourdot. Potential field approach for haptic selection. In *Proceedings of Graphics Interface (GI'2009)*, pages 203–206. Canadian Information Processing Society, 2009.

[371] T. Sohmura, N. Kusumoto, T. Otani, S. Yamada, K. Wakabayashi, and H. Yatani. CAD/CAM fabrication and clinical application of surgical template and bone model in oral implant surgery. *Clinical Oral Implants Research*, 20(1):87–93, 2009.

[372] M. A. Srinivasan. What is haptics? Technical report, Massachusetts Institute of Technology, 1995.

[373] M. A. Srinivasan and Basdogan. Haptics in virtual environments: Taxonomy, research status, and challenges. *Computers and Graphics*, 21(4):393–404, 1997.

[374] V. Srinivasan. Standardizing the specification, verification, and exchange of product geometry: Research, status and trends. *Computer-Aided Design*, 40(7):738–749, 2008. Current State and Future of Product Data Technologies (PDT).

[375] D. Stewart. Rigid-body dynamics with friction and impact. *SIAM Review*, 42(1):3–39, 2000.

[376] E. Stoll, M. Wilde, and C. Pong. Using virtual reality for human-assisted in-space robotic assembly. In *Proceedings of the 2009 World Congress on Engineering and Computer Science*, volume 2, 2009.

[377] J. E. Stone, J. Gullingsrud, and K. Schulten. A system for interactive molecular dynamics simulation. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics (I3D'2001)*, pages 191–194, New York, NY, USA, 2001.

[378] A. Sud, N. Govindaraju, R. Gayle, I. Kabul, and D. Manocha. Fast proximity computation among deformable models using discrete Voronoi diagrams. *ACM Transactions on Graphics*, 25(3):1144–1153, 2006.

[379] H. Sugarman, E. Dayan, A. Weisel-Eichler, and J. Tiran. The Jerusalem telerehabilitation system, a new low-cost, haptic rehabilitation approach. *CyberPsychology and Behavior*, 9(2):178–182, 2006.

[380] R. C. W. Sung, J. M. Ritchie, T. Lim, and H. Medellin. Assembly planning and motion study using virtual reality. In *Proceedings of the 2009 ASME-AFM World Conference on Innovative Virtual Reality*, pages 31–38, 2009.

[381] R. C. W. Sung, J. M. Ritchie, G. Robinson, P. Day, J. R. Corney, and H. J. Rea. The role of therbligs in automated design process mapping. In *Proceedings of the KIM Project Conference*, 2008.

[382] R. C. W. Sung, J. M. Ritchie, G. Robinson, P. N. Day, J. R. Corney, and T. Lim. Automated design process modelling and analysis using immersive virtual reality. *Computer-Aided Design*, 41(12):1082–1094, 2009.

[383] I. E. Sutherland. *Sketchpad: A Man-Machine Graphical Communication System*. Ph.D. dissertation, Massachusetts Institute of Technology (MIT), 1963.

[384] I. E. Sutherland. The ultimate display. *Multimedia: From Wagner to Virtual Reality*, 1965.

[385] L. B. Tang, Y. H. Chen, and X. J. He. Multi-material compliant mechanism design and haptic evaluation. *Virtual and Physical Prototyping*, 2(3):155–160, 2007.

[386] M. Tang, D. Manocha, and R. Tong. Fast continuous collision detection using deforming non-penetration filters. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, pages 7–13. ACM, 2010.

[387] L. Tching and G. Dumont. Haptic simulations based on non-smooth dynamics for rigid-bodies. In *Proceedings of the 15th ACM Symposium on Virtual Reality Software and Technology (VRST'2008)*, pages 87–90, New York, NY, USA, 2008.

[388] L. Tching and G. Dumont. Interactive simulation based on non-smooth contact dynamics: Application to haptic rigid-body simulations. In *Contecpt and Integrated Design and Manufacturing in Mechanical Engineering (C&IDMME'2008)*, pages 87–90, New York, NY, USA, 2008.

[389] L. Tching, G. Dumont, and J. Perret. Haptic assembly of CAD models using virtual constraint guidance. In *Proceedings of the 2010 ASME World Conference on Innovative Virtual Reality*, pages 11–19, 2010.

[390] L. Tching, G. Dumont, and J. Perret. Interactive simulation of CAD models assemblies using virtual constraint guidance. *International Journal on Interactive Design and Manufacturing*, 4(2):95–102, 2010.

[391] M. Teschner, S. Kimmerle, B. Heidelberger, G. Zachmann, L. Raghupathi, A. Fuhrmann, M. P. Cani, F. Faure, N. Magnenat-Thalmann, W. Strasser, and P. Volino. Collision detection for deformable objects. *Computer Graphics Forum*, 24(1):61–81, 2005.

[392] T. V. Thompson II and E. Cohen. Direct haptic rendering of complex trimmed NURBS models. In *Proceedings of the ACM SIGGRAPH'2005 Courses*, SIGGRAPH'2005, 2005.

[393] T. V. Thompson II, D. D. Nelson, E. Cohen, and J. Hollerbach. Maneuverable NURBS models within a haptic virtual environment. In *Proceedings of the 6th Annual Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, volume 61, pages 37–44, 1997.

[394] Thomas V. Thompson II, T. V., D. E. Johnson, and E. Cohen. Direct haptic rendering of sculptured models. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, I3D'1997, pages 167–176, 1997.

[395] R. B. Tilove. Set membership classification: A unified approach to geometric intersection problems. *IEEE Transactions on Computers*, 100(10):874–883, 1980.

[396] R. B. Tilove and A. A. G. Requicha. Closure of Boolean operations on geometric entities. *Computer-Aided Design*, 12(5):219–220, 1980.

[397] D. Touraine and P. Bourdot. VEserver: A manager for input and haptic multi-sensorial devices. In *Proceedings of the 10th IEEE International Workshop on Robot and Human Interactive Communication*, pages 2–7, 2001.

[398] D. Touraine, P. Bourdot, Y. Bellik, and L. Bolot. A framework to manage multimodal fusion of events for advanced interactions within virtual environments. In *Proceedings of the 2002 Workshop on Virtual Environments (EGVE'2002)*, pages 159–168. Eurographics Association, 2002.

[399] C. Trakunsaranakom, F. Noël, and P. Marin. Assessment of virtual reality environments for design activities. In G Zachmann, J. Perret, and A. Amditis, editors, *Conference and Exhibition of the European Association of Virtual and Augmented Reality (EURO-VR'2014)*, 2014.

[400] M. Y. Tsalamlal, N. Ouarti, and M. Ammi. Non-intrusive haptic interfaces: State-of-the art survey. In I. Oakley and S. Brewster, editors, *Haptic and Audio Interaction Design*, volume 7989 of *Lecture Notes in Computer Science*, pages 1–9. Springer Berlin Heidelberg, 2013.

[401] W. F. van Gunsteren. The role of computer simulation techniques in protein engineering. *Protein Engineering*, 2(1):5–13, 1988.

[402] J. M. Vance and G. Dumont. A conceptual framework to support natural interaction for virtual assembly tasks. In *Proceedings of the 2011 ASME World Conference on Innovative Virtual Reality*, pages 273–278, 2011.

[403] B. D. Varalakshmi, J. Thriveni, K. R. Venugopal, and L. M. Patnaik. Haptics: State of the art survey. *International Journal of Computer Science Issues*, 9(5):234–244, 2012.

[404] M. Veit, A. Capobianco, and D. Bechmann. Dynamic decomposition and integration of degrees of freedom for 3-D positioning. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology (VRST'2010)*, pages 131–134, New York, NY, USA, 2010.

[405] H. B. Voelcker and A. A. G. Requicha. Geometric modeling of mechanical parts and processes. *Computer*, 10(12):48–57, 1977.

[406] H. B. Voelcker and A. A. G. Requicha. Geometric modeling of mechanical parts and processes. Production Automation Project, Technical Memo. No. 23 (TM-23), University of Rochester, 1977.

[407] H. B. Voelcker and A. A. G. Requicha. Research in solid modeling at the university of rochester: 1972–1987. In L. Piegl, editor, *Fundamental Developments of Computer-Aided Geometric Modeling*, pages 203–254. Academic Press Ltd., London, UK, 1993.

[408] S. Volkov and J. M. Vance. Effectiveness of haptic sensation for the evaluation of virtual prototypes. *Journal of Computing and Information Science in Engineering*, 1(2):123–128, 2001.

[409] H. Wan, S. Gao, Q. Peng, G. Dai, and F. Zhang. MIVAS: A multi-modal immersive virtual assembly system. In *Proceedings of the 2004 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE'2004)*, pages 113–122, 2004.

[410] M. Wan and W. A. McNeely. Quasi-static approximation for 6 degrees-of-freedom haptic rendering. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, pages 257–262, Washington, DC, USA, 2003.

[411] G. G Wang. Definition and review of virtual prototyping. *Journal of Computing and Information Science in Engineering*, 2(3):232–236, 2002.

[412] Y. Wang, Y. Chen, Z. Nan, and Y. Hu. Ieee international conference on accessibility analysis for cmm inspection planning using haptic device. In *Robotics and Biomimetics (ROBIO'2006)*, pages 1239–1243, 2006.

[413] Y. Wang, S. Jayaram, and U. Jayaram. Physically based modeling in virtual assembly. *International Journal of Virtual Reality*, 5:1–14, 2001.

[414] Y. Wang, S. Jayaram, U. Jayaram, and K. Lyons. *Physically Based Modeling in Virtual Assembly*. PhD thesis, Washington State University, 1998.

[415] Y. Wang, U. Jayaram, S. Jayaram, and S. Imtiyaz. Methods and algorithms for constraint-based virtual assembly. *Virtual Reality*, 6(4):229–243, 2003.

[416] Y. Wang, W. Zhang, Y. Chen, D. Liu, and R. Zhang. Accessibility analysis for CMM inspection planning by means of haptic device and STL representation. In *IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurements Systems (VECIMS'2009)*, pages 174–178, 2009.

[417] S. Webel, U. Bockholt, T. Engelke, N. Gavish, M. Olbrich, and C. Preusche. An augmented reality training platform for assembly and maintenance skills. *Robotics and Autonomous Systems*, 61(4):398–403, 2013. Models and Technologies for Multi-modal Skill Training.

[418] S. Webel, U. Bockholt, T. Engelke, N. Gavish, and F. Tecchia. Design recommendations for augmented reality based training of maintenance skills. In *Recent Trends of Mobile Collaborative Augmented Reality Systems*, pages 69–82. Springer, 2011.

[419] S. Webel, U. Bockholt, T. Engelke, M. Peveri, M. Olbrich, and C. Preusche. Augmented reality training for assembly and maintenance skills. In *Proceedings of the International Conference (SKILLS'2011), BIO Web of Conferences*, volume 1, page 00097. EDP Sciences, 2011.

[420] P. Wegner. Why interaction is more powerful than algorithms. *Communications of*, 40(5):80–91, 1997.

[421] D. E. Weisberg. *The Engineering Design Revolution: The People, Companies and Computer Systems That Changed Forever the Practice of Engineering*. Cyon Research Corporation, 2008.

[422] R. Weller. Inner sphere trees. In *New Geometric Data Structures for Collision Detection and Haptics*, pages 113–144. Springer International Publishing, 2013.

[423] R. Weller and G. Zachmann. Inner sphere trees for proximity and penetration queries. In *Proceedings of the 2009 Robotics: Science and Systems Conference (RSS'2009)*, volume 2, 2009.

[424] R. Weller and G. Zachmann. Stable 6-DOF haptic rendering with inner sphere tees. In *Proceedings of the 2009 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE'2009)*, 2009.

[425] R. Weller and G. Zachmann. A unified approach for physically-based simulations and haptic rendering. In *Proceedings of the ACM SIGGRAPH'2009 Symposium on Video Games*, pages 151–159, New York, NY, USA, 2009.

[426] R. Weller and G. Zachmann. Inner sphere trees and their application to collision detection. In G. Brunnett, S. Coquillart, and G. Welch, editors, *Virtual Realities*, pages 181–201. Springer Vienna, 2011.

[427] J. G. W. Wildenbeest, D. A. Abbink, C. J. M. Heemskerk, F. C. T. van der Helm, and H. Boessenkool. The impact of haptic feedback quality on the performance of teleoperated assembly tasks. *IEEE Transactions on Haptics*, 6(2):242–252, 2013.

[428] A. Wilson, E. Larsen, D. Manocha, and M. C. Lin. Partitioning and handling massive models for interactive collision detection. In *Computer Graphics Forum*, volume 18, pages 319–330. Wiley Online Library, 1999.

[429] A. M. Wollacott and Jr. Merz. Haptic applications for molecular structure manipulation. *Journal of Molecular Graphics and Modelling*, 25(6):801–805, 2007.

[430] D. L. Wu, X. J. Zhen, X. Fan, Y. Hu, and H. Zhu. A virtual environment for complex products collaborative assembly operation simulation. *Journal of Intelligent Manufacturing*, 23(3):821–833, 2012.

[431] P. Xia, A. Lopes, and M. Restivo. Design and implementation of a haptic-based virtual assembly system. *Assembly Automation*, 31(4):369–384, 2011.

[432] P. J. Xia, A. M. Lopes, M. T. Restivo, and Y. X. Yao. A new type haptics-based virtual environment system for assembly training of complex products. *International Journal of Advanced Manufacturing Technology*, 58(1-4):379–396, 2012.

[433] P. j. Xia, A. Mendes Lopes, and M. T. Restivo. A review of virtual reality and haptics for product assembly: From rigid parts to soft cables. *Assembly Automation*, 33(2):157–164, 2013.

[434] Y. Yang, S. Jayaram, U. Jayaram, K. Lyons, and P. Hart. Representation of swept volumes in a parametric CAD system using trajectory information from virtual environments. In *Proceedings of the ASME Design Engineering Technical Conferences (DETC'1999)*, number DETC99/CIE-9109. American Society of Mechanical Engineers (ASME), 1999.

[435] Z. Y. Yang and Y. H. Chen. Haptic rendering of milling. In *Proceedings of the EuroHaptics*, 2003.

[436] Z. Y. Yang and Y. H. Chen. Reverse engineering based on virtual volume sculpting. *Computer-Aided Design and Applications*, 1(1-4):63–72, 2004.

[437] Z. Y. Yang and Y. H. Chen. A reverse engineering method based on haptic volume removing. *Computer-Aided Design*, 37(1):45–54, 2005.

[438] Z. Y. Yang and Y. Z Chen. Inspection path generation in haptic virtual CMM. *Computer-Aided Design and Applications*, 2(1-4):273–282, 2005.

[439] Z. Y. Yang, L. Lian, and Y. Chen. Haptic function evaluation of multi-material part sesign. *Computer-Aided Design*, 37(7):727–736, 2005.

[440] G. Zachmann. Rapid collision detection by dynamically aligned DOP-trees. In *Proceedings of the 1998 IEEE Virtual Reality Annual International Symposium*, pages 90–97, 1998.

[441] U. Zaldivar-Colado and S. Garbaya. Virtual assembly environment modelling. In *Proceedings of the 2009 ASME-AFM World Conference on Innovative Virtual Reality*, pages 157–163, 2009.

[442] X. Zhang, M. Lee, and Y. J. Kim. Interactive continuous collision detection for non-convex polyhedra. *The Visual Computer*, 22(9-11):749–760, 2006.

[443] X. Zhang, S. Redon, M. Lee, and Y. J. Kim. Continuous collision detection for articulated models using taylor models and temporal culling. In *Proceedings of SIGGRAPH'2007*, volume 26, 2007.

[444] X. Zhang, S. Redon, M. Lee, and Y. J. Kim. Continuous collision detection for articulated models using taylor models and temporal culling. *ACM Transactions on Graphics (TOG)*, 26(3-15), 2007.

[445] X. J. Zhen, D. L. Wu, X. Fan, and Y. Hu. Distributed parallel virtual assembly environment for automobile development. *Assembly Automation*, 29(3):279–289, 2009.

[446] X. J. Zhen, D. L. Wu, Yong Hu, and X-M Fan. A real-time simulation grid for collaborative virtual assembly of complex products. *International Journal of Computer Integrated Manufacturing*, 23(6):500–514, 2010.

[447] Z. Zhu, S. Gao, H. Wan, Y. Luo, and W. Yang. Grasp identification and multi-finger haptic feedback for virtual assembly. In *Proceedings of the 2004 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE'2004)*, number DETC2004-57718, pages 661–669. American Society of Mechanical Engineers (ASME), 2004.

[448] Y. Zhuang and Canny. Real-time simulation of physically realistic global deformation. Technical report, University of California at Berkeley, 1999.

[449] P. Zimmermann. Virtual reality aided design. a survey of the use of VR in automotive industry. In D. Talaba and A. Amditis, editors, *Product Engineering*, pages 277–296. Springer Netherlands, 2008.

[450] F. Zorriassatine, C. Wykes, R. Parkin, and N. Gindy. A survey of virtual prototyping techniques for mechanical product development. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 217(4):513–530, 2003.