INTERACTIVE 3D MODELING IN GRASSHOPPER USING LEAP MOTION AND FIREFLY: A CASE STUDY OF FINGERTIP-BASED PARAMETRIC MODELING

Hendro Trieddiantoro Putro[1], Suparno[2 *],

1.2. Department of Architecture, Faculty of Science and Technology, Universitas Teknologi Yogyakarta, Yogyakarta

*Correspondent Author: hendro.trieddiantoro@gmail.com

Date received: March 20, 2026 • Reviewer completion date: April 13 & 15, 2026 • Revision date: April 25, 2026 • Publication date: September 15, 2026.

DOI: 10.24167/joda.v6i1.16013

Abstract: *This study explores the integration of Leap Motion, Firefly, and Grasshopper to enable fingertip-based interactive 3D modeling in architectural design. A parametric modeling workflow was developed where fingertip movements captured by Leap Motion were translated into geometric parameters via Firefly, generating curves and pipe geometries almost simultaneously in the Rhinoceros environment. Experimental evaluation involved performing modeling tasks, including geometry creation and parametric form manipulation, while measuring task completion time, system response delay, and gesture recognition stability. Results show that curve and pipe geometry creation occurred with minimal delay, averaging 0.72 s and 0.93 s, respectively, indicating real-time responsiveness suitable for early-stage parametric modeling. Parametric form updates required slightly longer response times average 1.08 s due to continuous computation. Gesture recognition was generally stable, with minor jitter observed during rapid hand movements. Qualitative evaluation indicates that the fingertip-based interface is responsive and supports exploratory design interaction. Overall, the findings demonstrate the potential of gesture-based interaction for real-time parametric modeling, providing a natural and immersive design experience. The study highlights opportunities for combining such interfaces with augmented and virtual reality to further enhance interactive architectural design workflows.*

Keywords: *Interactive 3D Modeling, Leap Motion, Firefly, Grasshopper, Parametric Design, Digital Architecture.*

1. Introduction

In the current digital age, technical developments have significantly changed a number of industries, including architecture. The architectural design process has advanced with the creation of numerous software programs related to architecture, allowing for the development of new design innovations [1].

One of the increasingly emerging technological developments is the use of interactive three-dimensional (3D) modeling, which allows for more realistic, dynamic, and intuitive visual representations. This 3D modeling not only aids in the design process but also strengthens the communication of ideas between designers and stakeholders [2]. Furthermore, technological developments in architecture have led to the use of computers. Computer modeling accelerates the design process, enabling architects to create complex structures with high precision [3]. The use of technologies such as virtual reality and augmented reality allows stakeholders to visualize and interact with designs before construction, fostering collaboration and informed decision-making. Applications of building information modeling can also assist the full building lifecycle, decrease mistakes, and enhance project coordination [4]. Rhinoceros is a widely used 3D modeling platform in architectural design, providing a flexible environment for geometric modeling, as shown in Figure 1.

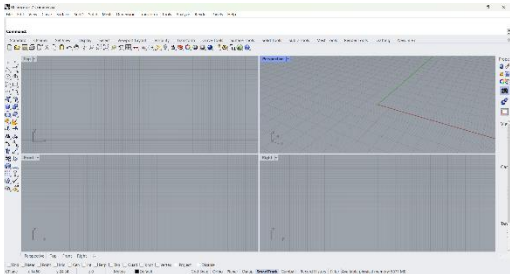

Figure 1. Rhinoceros User Interface

Within Rhinoceros, the Grasshopper plugin serves as a parametric design tool, enabling the generation and manipulation of geometry through visual programming, which allows for flexible and complex form exploration

Grasshopper is a visual programming plugin for Rhinoceros software that is widely used in parametric design. The application of Grasshopper is closely associated with generative design, which explores several design iterations using algorithms to foster creative and sustainable solutions [4] and facilitate data-driven design exploration [17].

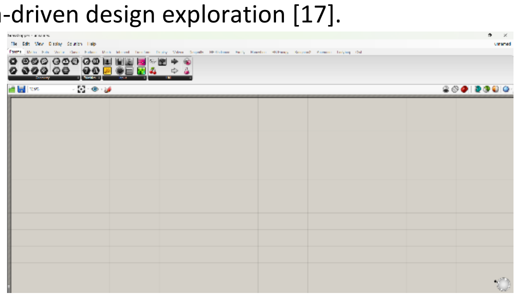

Figure 2. Grasshopper Parametric Visual Programming

The use of Grasshopper, a Rhinoceros software visual programming plugin, has proven crucial for enabling parametric modeling. Designers may produce intricate, algorithm-based designs quickly [5], experiment with complex geometries and shapes, fostering creativity through generative design methods [6] [7]. Tools like Grasshopper facilitate the exploration of various design alternatives, optimizing solutions based on specific parameter constraints [8].

Together, Grasshopper and Rhinoceros provide the ability to define precise parametric control of models, the ability to explore generative design workflows, and a platform for developing high-level programming logic all within an intuitive graphical interface [9]. Despite its capabilities, the adoption of Grasshopper in architectural design education and practice remains limited in certain contexts [10].

To extend the capabilities of Grasshopper, the Firefly plugin enables real-time integration of data from external hardware, such as sensors and controllers, allowing parametric models to respond dynamically to real-world inputs [11].

Figure 2. Firefly: Integration Data Real-Time

Firefly is a plugin for Grasshopper that enables real-time data integration from various hardware devices, including sensors and external controllers. Firefly was developed by Andy Payne and Jason Kelly Johnson in 2010. Firefly allows designers to connect physical environmental data with digital models, creating responsive and interactive models, allowing architects to adapt their models based on real-world conditions [11]. For example, Firefly is used as a robot controller that can be controlled with an iPhone via Open Sound Control (OSC) [12]. The use of Firefly opens opportunities for design exploration based on direct interaction between humans and digital models [13]. The use of Firefly is also found as part of the kinetic architecture learning process in architecture study programs, specifically the creation of 3D models of kinetic facade designs during the design development stage in the architectural design studio course [14]. It was further reported that students experienced a high level of comfort and enjoyment by combining hand movements with digital technology in learning kinetic architecture applications using interactive three-dimensional modeling for digital fabrication [15].

To complement the real-time interaction, devices like Leap Motion, designed to accurately detect hand movements, are further opening opportunities for touchless interaction in 3D modeling.

Figure 3. Leap Motion Hardware

Leap Motion is a hand gesture capture device that enables touchless interaction in 3D modeling. It is widely used in research related to sign language,

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

robotics, education, digital home assistance, virtual reality, augmented reality, and medicine [16]. With Leap Motion, designers can manipulate 3D models intuitive and immersive[8] [17] with hand movements in real-time [18] [19]. It has high accuracy in detecting hand gestures such as grasping in a virtual world [26].

According to Jailungka & Charoenseang [20] and Xiao & Peng [19], Leap Motion detects hand movements and converts them into instructions for controlling 3D models such as rotating, resizing, and moving objects. Leap Motion allows architects to create two-dimensional digital drawings through hand movements, enabling a more organic design process that mimics traditional drawing techniques [21]. Leap Motion can be integrated into collaborative environments, allowing multiple users to interact with 3D displays simultaneously, enriching the ideation process [22]. In addition, Leap Motion is claimed to have the best price-performance ratio for the Hand Gesture Recognition HGR system in Human Computer Interaction (HCI) research [16].

This combination of technologies has great potential to increase efficiency, creativity, and collaboration in the design process, leading to innovative approaches in architectural design [23]. Interactive 3D modeling techniques, including computer-aided design and virtual reality (VR), have become an integral part of modern architectural practice, enhancing creativity and problem-solving capabilities [24]. Several technologies support this modeling, including Firefly, Grasshopper, and Leap Motion, each of which plays a crucial role in realizing interactive experiences.

In education, Leap Motion is used as a learning tool for elementary school students to introduce the solar system [25] as well as in augmented reality applications for learning about bacteria [26]. In architecture, it can be used to manipulate 3D geometry in real time, supporting interactive and collaborative design processes [27]. Leap Motion's integration with augmented reality devices like Microsoft HoloLens enables intuitive 3D model prototyping. Users can manipulate holographic objects in real time, performing actions such as translation, rotation, and scaling, which are crucial for architectural and product design [20].

In 3D modeling, Leap Motion can capture human movement to manipulate control points on NURBS surfaces, enabling designers to create and modify 3D models with precision. The system supports both one-handed and two-handed interaction, providing flexibility in design resolution and creativity [27]. It allows designers to intuitively manipulate and adjust virtual models without collision detection. This can increase precision and efficiency in the design process, enabling users to create customized systems for optimal results in architectural and product design projects [28].

Based on the literature review above, the integration of Firefly, Grasshopper, and Leap Motion offers significant opportunities for interactive 3D modeling. This combination allows designers to incorporate real-time data from Leap Motion into the Grasshopper programming environment via Firefly. This allows designers to directly manipulate models through hand gestures, creating a more natural and efficient interaction system. This integration has been shown to improve visualization, user engagement, and efficiency of digital-based design processes or digital fabrication [29]. The combination of parametric design using Grasshopper and digital fabrication enables rapid prototyping, enabling architects to efficiently create complex geometries [30].

Although previous studies have explored gesture-based interaction using Leap Motion in virtual environments and educational applications, its implementation in parametric architectural modeling remains relatively limited. In particular, few studies have investigated the integration of Leap Motion with parametric design environments such as Grasshopper to support real-time geometric generation and manipulation. Furthermore, empirical evaluation of gesture-based modeling performance in architectural workflows has not been extensively examined.

This study addresses the following research questions, such as, how can Leap Motion be integrated with Firefly and Grasshopper to enable fingertip-based 3D modeling? Second, how effectively can fingertip motion be translated into geometric control parameters within the Grasshopper environment? Third, what are the advantages and limitations of Leap Motion interaction in architectural design workflows?

Therefore, this study aims to develop and experimentally evaluate an interactive parametric modeling workflow by integrating Leap Motion, Firefly, and Grasshopper within a real-time architectural design environment. In addition, this study provides a preliminary experimental evaluation of gesture-based interaction in architectural modeling workflows, highlighting both its advantages in intuitive spatial interaction and its technical limitations related to motion tracking stability. The findings provide a foundation for future research exploring immersive design interfaces that combine parametric modeling with emerging interaction technologies such as augmented reality and virtual reality. This contribution is significant as it demonstrates how gesture-based interaction can be effectively embedded within parametric design workflows, enabling more direct spatial interaction and potentially reshaping the way designers explore and manipulate geometry in the conceptual design phase.

2. Research Methods

This research adopts an experimental computational approach to develop and evaluate an interactive parametric modeling system, a method commonly applied in human–computer interaction and computational design research [31]. The study consists of two main stages: system development and experimental testing. The experiment was conducted through repeated trials performed by the author, focusing on system performance and interaction responsiveness. No formal user study involving multiple participants was carried out, as the objective of this study is to evaluate system feasibility rather than generalize user behavior.

In the first stage, a parametric modelling system was developed using Grasshopper within the Rhinoceros environment. The Firefly plugin was used to connect the Leap Motion device with the Grasshopper visual programming interface. Leap Motion captures hand gesture data, including fingertip positions, hand orientation, and finger vectors, enabling real-time hand tracking in interactive systems [32]. These motion data are transmitted through Firefly and processed in Grasshopper as parametric inputs. The input parameters are then used to generate geometric elements in the Rhinoceros modelling environment.

The parametric script processes the fingertip coordinates to create dynamic curves using interpolation methods. These curves are then transformed into three-dimensional pipe geometries using variable radius parameters. The resulting geometry is visualized in the Rhinoceros viewport. The experiment involved a parametric modeling task consisting of a continuous workflow, including curve generation using fingertip movement, pipe geometry creation based on the generated curve, and parametric manipulation by adjusting geometric parameters through gesture input. The tasks were predefined and standardized to ensure consistency across repeated trials, following common experimental practices in interaction studies [33]. Each trial followed the same interaction procedure and gesture sequence, although minor variations in hand movement naturally occurred during execution. For each trial, the author performed the same sequence of actions: initiating fingertip movement to generate a curve, allowing the system to create pipe geometry from the recorded trajectory, and subsequently modifying the geometric parameters through gesture input.

In the second stage, experimental testing was conducted to evaluate system performance and interaction responsiveness. The experiment involved performing the predefined modeling tasks using hand gestures detected by Leap Motion.

Figure 4. Device Layout and Experiment Positions for the Research Activity

The experimental setup is illustrated in Figure 5, which shows the device layout and testing positions, including (A) hand movement area, (B) Leap Motion sensor, (C) computer, and (D) recording camera. The system was tested on a desktop computer equipped with an Intel Core i7-3770 processor, 16 GB RAM, and an AMD Radeon RX 570 GPU. The Leap Motion controller was used as the hand-tracking device, integrated with Rhinoceros and Grasshopper through the Firefly plugin. This configuration represents a mid-range system, indicating that the proposed workflow

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

can be implemented without requiring high-end computational resources.

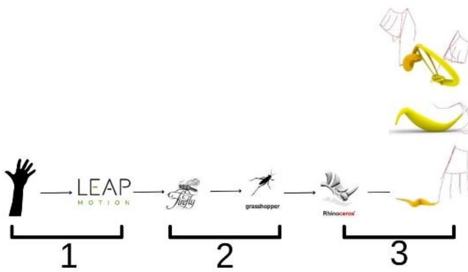

Figure 5. The input, integration, and 3D visualization

The input, integration, and 3D visualization processes are illustrated in the figure above. The following section explains how fingertip movements are translated into three-dimensional geometric representations in the Rhinoceros viewport.

1. **Fingertips Input:** Fingertip data are captured using the Leap Motion device, which serves as a bridge between physical hand gestures and digital input data
2. **Motion Data Integration:** The Firefly plugin is used to transmit and translate Leap Motion data into the Grasshopper environment. Firefly functions as an interface between the hardware and the parametric modeling platform.
3. **Parametric Programming:** Grasshopper processes the motion data into parametric inputs, enabling the generation and manipulation of geometric forms**.**
4. **3D Visualization:** The resulting geometry is visualized in the Rhinoceros viewport in real time.

The experiment was conducted through repeated trials to evaluate system performance and interaction responsiveness. A single parametric modeling workflow was performed using hand gestures captured by the Leap Motion device. This workflow consisted of curve generation, pipe geometry creation, and parametric manipulation as part of a continuous interaction process. Each trial followed the same interaction procedure and gesture sequence. The entire workflow was repeated five times under consistent conditions, resulting in five trials. The experiment was recorded using a camera to capture the interaction process between the user and the modeling system. The recorded video was subsequently used for frame-based analysis by identifying timestamps of gesture initiation and system response [34].

Performance measurement was conducted to evaluate system responsiveness and interaction quality. The system performance was assessed using three main indicators: task completion time, system response delay, and gesture recognition stability. Task completion time was defined as the duration from the initiation of gesture input to the completion of the modeling workflow. System response delay was measured through frame-based video analysis by identifying the time difference between the initiation of a gesture and the appearance of the corresponding geometry in the Rhinoceros viewport, following established approaches in human–computer interaction studies [35]. The video timecode was recorded in the format hh:mm:ss.xx, representing the temporal position of each event within the recording. For each task, two timestamps were identified: the gesture initiation time (T_gesture_start), defined as the moment when fingertip movement begins, and the geometry appearance time (T_geometry_appearance), defined as the moment when the generated geometry becomes visibly rendered. The response delay was calculated as the temporal difference between these events using the equation $\Delta t = T\_geometry\_appearance - T\_gesture\_start$. All timestamps were converted into seconds to ensure uniform calculation. This method follows established human–computer interaction evaluation practices that employ video-based temporal annotation when real-time system logging is unavailable. Gesture recognition stability was evaluated qualitatively based on the consistency of fingertip tracking, particularly by observing the presence of jitter or tracking interruptions during interaction. Minor jitter refers to small fluctuations that do not significantly affect the modeling process, while major jitter indicates instability that disrupts interaction continuity.

3. Discussion

Parametric Model Development

Several previous studies have demonstrated the development of applications with Leap Motion using various types of software, for example, game engine software like Unity 3D [25][26][36]. Digital art software like Corel Painter [21] and visual programming software for 3D modeling such as Grasshopper [12] [14]. In this study, parametric model development was conducted using Grasshopper, specifically the Firefly plug-in.

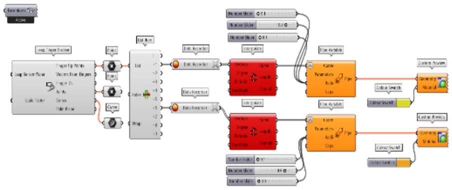

Figure 6. Parametric Script in Grasshopper

The image above is a representation of a parametric script in Grasshopper, which shows the integration of Leap Motion and Firefly in Grasshopper to generate geometry based on fingertips points data.

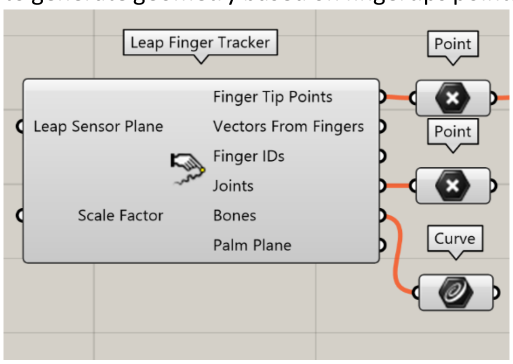

Figure 7. Input Leap Motion (Leap Finger Tracker)

The Leap Finger Tracker component retrieves data from the Leap Motion device, including fingertip points, finger vectors, and additional information such as finger IDs, joints, bones, and the palm plane, which describe the position and orientation of the hand. The primary output consists of fingertip coordinates used for geometry generation. In addition, joint data represent the positions of finger bone joints as points, while bone data are visualized as curved lines representing the structure of the fingers.

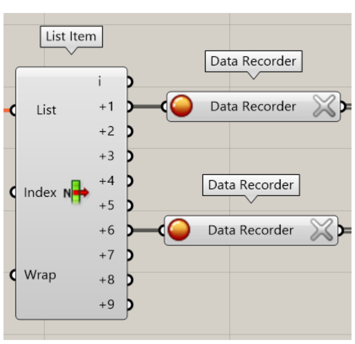

Figure 8. List Item and data recorder

The List Item component retrieves a specific point from the Leap Motion data list based on an index. The index can be set using a slider to determine which point or finger is used in the design. In this study, index values of +1 and +6 are used to represent the index finger of the right and left hands, respectively. The Data Recorder component stores the positional data of the selected fingertip over time, enabling real-time tracking of hand movements and the generation of smooth curves.

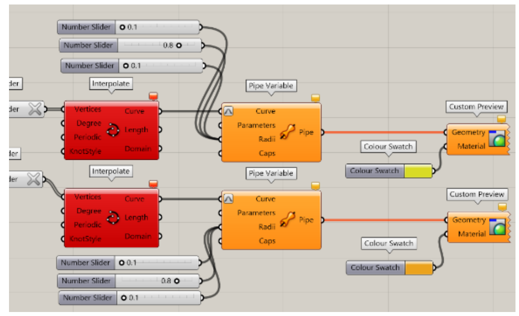

Figure 9. Interpolate and pipe variable

Point data from the Leap Motion sensor is used to generate an interpolated curve. The aim is to produce a smooth curve from previously recorded input point data. The Pipe Variable component is then used to generate geometry from the resulting curves, converting them into pipe-shaped forms. The pipe radius is controlled using a Number Slider, which allows dynamic adjustment of numerical values such as pipe radius and curve smoothness.

Next, the resulting geometry is displayed in the Rhinoceros viewport using two colors: yellow for the 3D model generated from the right-hand index finger, and orange for the left-hand index finger. This visualization is achieved using a Custom Preview component, which enables the display of geometry with customized materials or colors. This approach is useful for providing a clearer and more visually informative representation of parametric designs without requiring full rendering, as it focuses on real-time visual feedback of the generated geometry.

Experimental Setup and Testing

The following illustrates the experimental setup for interactive 3D modeling using Leap Motion and Firefly in Grasshopper, where user hand movements are captured through the Leap Motion sensor and translated into parametric input for real-time geometric generation. The system setup consists of five main components: (A) hand movement as the primary input source, (B) Leap Motion device for motion tracking, (C) computer screen for visual

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

feedback and interaction monitoring, (D) camera body used for recording the experimental process, and (E) the resulting 3D model generated within the parametric environment. This configuration enables real-time interaction between user gestures and digital modeling output, allowing direct evaluation of gesture-based parametric design performance.

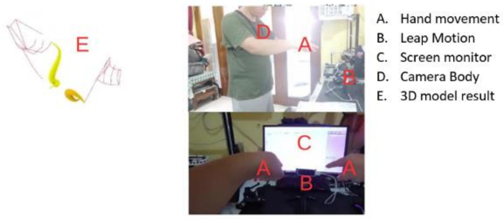

Figure 10. 3D modeling trial setup using leap motion

Figure 11 presents an example of a single-hand interaction outcome, where a continuous 3D geometry is generated from fingertip point and motion input. This highlights the potential of the system to support rapid form exploration and early-stage conceptual modeling without the need for traditional input devices.

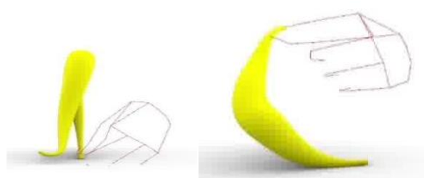

Figure 11. 3D modeling using leap motion with one hand

Test results indicate that the 3D modeling process shows improved user control and interaction fluidity, thereby enhancing users' creative experience during spatial exploration and form generation. The following figures present the outcomes of the 3D modeling test conducted using both hands simultaneously, demonstrating how dual-hand fingertip inputs enable more complex and continuous geometric manipulation in real time.

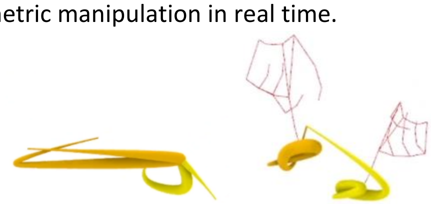

Figure 12. 3D modeling using leap motion with both hand

The performance of the interactive parametric modeling system was evaluated using task completion time, gesture recognition stability, interaction behavior, and system response delay. The average task completion time was approximately 2.5 seconds across all trials, reflecting the system's capability to support continuous real-time interaction during the modeling process. Task completion time was measured from the initial fingertip input captured by the Leap Motion sensor to the successful generation of 3D geometry in the Rhinoceros viewport. Due to the integration of curve interpolation and pipe geometry creation in the parametric script, these two operations occurred almost synchronously, resulting in shorter task completion times. Gesture recognition stability refers to the consistency and reliability of the Leap Motion sensor in detecting fingertip movements without interruption or jitter, while interaction behavior was assessed qualitatively based on intuitiveness, responsiveness, and ease of controlling parametric geometry with fingertip gestures. In addition, system response delay was quantitatively measured for each task to evaluate the real-time responsiveness of the system. Table 1 summarizes the delays recorded for curve generation, pipe geometry creation, and parametric updates across multiple trials. The results indicate that curve and pipe generation exhibit minimal delays, confirming that these operations can be performed almost simultaneously, while parametric updates require slightly longer response times due to continuous computation and geometric adjustments.

Table 1. System Response Delay for Interactive 3D Modeling Tasks

Trial	Curve Delay (s)	Pipe Delay (s)	Parametric Update Delay (s)
T1	0.72	0.95	1.10
T2	0.68	0.91	1.05
T3	0.75	0.97	1.08
T4	0.70	0.93	1.12
T5	0.73	0.90	1.07
Average	**0.72**	**0.93**	**1.08**

The measured response delays indicate that geometry creation (curve + pipe) occurs almost in real time, with an average combined delay of less than 1 second, which supports the previously reported short task completion times. Parametric updates require slightly more time, reflecting the computational load of updating complex geometric forms based on continuous gesture input. Overall, the system

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

demonstrates sufficient responsiveness for intuitive fingertip-based 3D modeling. In human–computer interaction studies, response times below one second are generally considered acceptable for maintaining real-time interaction [35].

The system enables real-time gesture-based parametric modeling with sub-second response times, supporting direct spatial interaction in early-stage design exploration. The integration of Leap Motion enables direct mapping between fingertip motion and geometric generation, reducing the cognitive gap between input and output compared to traditional mouse-based interfaces. The evaluation of interaction behavior indicates that the system enables intuitive control and supports more fluid design exploration, particularly in early-stage form generation. This finding aligns with previous research highlighting the potential of gesture-based interfaces for enhancing spatial interaction in digital design environments [37] [38]. Novel algorithms improve gesture accuracy, allowing for more effective interactions with 3D objects [39].

While the combination of Leap Motion and Firefly offers significant advantages, challenges such as hand jitter and occlusion may affect modeling precision, necessitating further refinement of gesture recognition technologies [40]. Previous studies have reported that Leap Motion-based systems may experience tracking instability and jitter under certain motion conditions [41] [42]. This is consistent with the observed minor jitter in the experimental trials, although it did not significantly affect task completion.

Furthermore, prior research also highlights that mastering parametric modeling tools such as Grasshopper requires significant learning effort, which may affect accessibility for novice users [13].

4. Conclusion

This study developed an interactive parametric modeling workflow by integrating Leap Motion, Firefly, and Grasshopper. The system enables designers to generate and manipulate three-dimensional parametric geometry using fingertip-based motion in real time. Experimental testing demonstrated that gesture-based interaction can support intuitive spatial modeling and creative form exploration.

However, several limitations were observed, including tracking instability, motion jitter, and occasional occlusion of hand gestures. These issues may affect modeling precision and require further improvement in gesture recognition algorithms and sensor calibration.

Future research may explore the integration of gesture-based parametric modeling with immersive technologies such as augmented reality and virtual reality to create fully interactive design environments for architectural exploration and collaborative design.

Acknowledgments

The authors would like to thank Universitas Teknologi Yogyakarta and SA8B Architectural Research Laboratory. The authors also acknowledge the use of AI-assisted tools for language refinement and proofreading of the manuscript. The scientific content, analysis, and conclusions remain the sole responsibility of the authors.

References

[1] R. A. Putra, "Peran Teknologi Digital dalam Perkembangan Dunia Perancangan Arsitektur," 2018. [Online]. Available: www.jurnal.ar-raniry.com/index.php/elkawnie

[2] K. M. Kensek and D. Noble, "Building Information Modeling."

[3] K. S. Kharchenko, T. M. Tovstyk, and D. M. Diakonov, "THE INFLUENCE OF THE DEVELOPMENT OF COMPUTER TECHNOLOGIES ON THE PROCESS OF ARCHITECTURAL DESIGN," *Ukrainian Journal of Civil Engineering and Architecture*, no. 2 (020), pp. 106–115, Jun. 2024, doi: 10.30838/J.BPSACEA.2312.260324.106.1049.

[4] N. L. Rane, S. P. Choudhary, and J. Rane, "Leading-Edge Technologies for Architectural Design: A Comprehensive Review," *International Journal of Architecture and Planning*, vol. 3, no. 2, pp. 12–48, Sep. 2023, doi: 10.51483/ijarp.3.2.2023.12-48.

[5] H. Pottmann, Y. Liu, J. Wallner, A. Bobenko, and W. Wang, "Geometry of multi-layer freeform structures for architecture," in *ACM SIGGRAPH 2007 papers*, New York, NY, USA: ACM, Jul. 2007, p. 65. doi: 10.1145/1275808.1276458.

[6] M. Helenowska-Peschke, "APPLYING GENERATIVE MODELLING TOOLS TO EXPLORE ARCHITECTURAL FORMS," *The Journal of Polish*

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

Society for Geometry and Engineering Graphics, vol. 23, pp. 43–49, 2012.
[7] A. S. Rawat and G. Tiwari, “Modern Generative Design Tools: Siemens NX’s Algorithmic Feature and Rhinoceros 3D’s Grasshopper,” 2023, pp. 275–284. doi: 10.1007/978-981-99-3033-3_24.
[8] shimaa ashour and S. gogo, “Boosting the Design Process Using a Proposed Methodology Based on Computational Design,” *ERJ. Engineering Research Journal*, vol. 0, no. 0, pp. 0–0, Mar. 2024, doi: 10.21608/erjm.2024.244713.1301.
[9] G. Akos and R. Parsons, “FOUNDATIONS THE GRASSHOPPER PRIMER THIRD EDITION,” 2015. [Online]. Available: http://modelab.is/
[10] R. Chairiyah, A. E. Yetti, and I. Pujiyanti, “The Grasshopper+Rhino for 3D Modelling in Indonesian’s Education of Biomimetic Architecture,” 2022. [Online]. Available: www.rhino3d.com
[11] A. Payne, “INTERACTIVE PROTOTYPING AN INTRODUCTION TO ELECTRONICS USING ARDUINO, GRASSHOPPER, AND FIREFLY,” 2013. [Online]. Available: http://www.fireflyexperiments.com.
[12] A. L. Cheng, “InTUItive Robot Teleoperation within Ambient Assisted Living Environments via Grasshopper, Firefly, and gHowl,” 2014. [Online]. Available: http://www.liftarchitects.com/
[13] M. A. Zboinska, “Boosting the Efficiency of Architectural Visual Scripts,” in *Modelling Behaviour*, Cham: Springer International Publishing, 2015, pp. 479–490. doi: 10.1007/978-3-319-24208-8_40.
[14] A. Agirbas, “The Evaluation of the Relationship between the Use of Multi-Software and the Students’ Attitude towards Computers and Technology in Undergraduate Architectural Design Studio Education,” 2019.
[15] S. Allam and S. Alacame, “Tailoring Arduino for Interactive Digital Fabrication: Mechanism, Algorithms, Cases,” in *2023 20th Learning and Technology Conference (L&T)*, IEEE, Jan. 2023, pp. 145–150. doi: 10.1109/LT58159.2023.10092318.
[16] N. Majdoub Bhiri, S. Ameur, I. Alouani, M. A. Mahjoub, and A. Ben Khalifa, “Hand gesture recognition with focus on leap motion: An overview, real world challenges and future directions,” *Expert Syst. Appl.*, vol. 226, p. 120125, Sep. 2023, doi: 10.1016/J.ESWA.2023.120125.
[17] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, “Analysis of the Accuracy and Robustness of the Leap Motion Controller,” *Sensors*, vol. 13, no. 5, pp. 6380–6393, May 2013, doi: 10.3390/s130506380.
[18] U. Hirschberg, A. Sayegh, M. Frühwirth, and S. Zedlacher, “3D Motion Tracking in Architecture Turning Movement into Form-Emerging Uses of a New Technology,” 2006. [Online]. Available: http://iam.tugraz.at/
[19] Y. ; Xiao and Peng, “A HAND GESTURE-BASED INTERFACE FOR DESIGN REVIEW USING LEAP MOTION CONTROLLER,” 2017.
[20] P. Jailungka and S. Charoenseang, “Intuitive 3D Model Prototyping with Leap Motion and Microsoft HoloLens,” 2018, pp. 269–284. doi: 10.1007/978-3-319-91250-9_21.
[21] A. Diacodimitri and F. Rebecchini, “Drawing with bare hands. A hand-gesture based drawing experience with motion sensors.,” in *Blucher Design Proceedings*, São Paulo: Editora Blucher, Apr. 2023, pp. 969–980. doi: 10.5151/sigradi2022-sigradi2022_274.
[22] R. Vivian, “Propositions for a Mid-Air Interactions System Using Leap-Motion for a Collaborative Omnidirectional Immersive Environment,” 2017, pp. 203–216. doi: 10.1007/978-3-319-58697-7_15.
[23] Ö. Atabeyoğlu and G. Bilge, “Temporal and Spatial Evolution of Architecture,” *Mimarlık Bilimleri ve Uygulamaları Dergisi (MBUD)*, vol. 9, no. 1, pp. 19–43, Jan. 2024, doi: 10.30785/mbud.1325250.
[24] S. P. Dash, “AN EXPLORATORY STUDY ON DESIGN PROCESS IN ARCHITECTURE: PERSPECTIVE OF CREATIVITY,” *Creativity Studies*, vol. 14, no. 2, pp. 346–361, Sep. 2021, doi: 10.3846/cs.2021.12989.
[25] K. Melfasoni, A. Trisnadoli, and I. Surya, “Implementasi Leap Motion Controller pada Aplikasi Pengenalan Tata Surya,” *Journal Research and Development (ITJRD)*, vol. 4, no. 1, 2019, doi: 10.25299/itjrd.2019.vol4(1).xxxx.

[26] B. S. D. Nugraha, D. W. Wibowo, and A. Ravandhika, "Penerapan Teknologi Leap Motion dengan Augmented Reality sebagai Media Pembelajaran Bakteri," *Jurnal Sistem dan Informatika (JSI)*, vol. 16, no. 2, pp. 68–76, Jun. 2022, doi: 10.30864/jsi.v16i2.405.

[27] M. Cohen, D. Regazzoni, and C. Vrubel, "A 3D Virtual Sketching System Using NURBS Surfaces and Leap Motion Controller," *Comput. Aided. Des. Appl.*, vol. 17, no. 1, pp. 167–177, May 2019, doi: 10.14733/cadaps.2020.167-177.

[28] S. Mu and A. Sourin, "Virtual Assembling Using Hand Tracking with Leap Motion Controller," in *2021 International Conference on Cyberworlds (CW)*, IEEE, Sep. 2021, pp. 121–124. doi: 10.1109/CW52790.2021.00026.

[29] K. Shumilov and Y. Guryeva, "Using the Grasshopper-Rhino-Archicad bundle for modeling complex architectural objects in BIM design," 2022, p. 020027. doi: 10.1063/5.0107609.

[30] J. Li, "Transforming architecture: The synergy of digital fabrication and parametric design," *Applied and Computational Engineering*, vol. 66, no. 1, pp. 243–248, May 2024, doi: 10.54254/2755-2721/66/20240968.

[31] Jenny. Preece, Yvonne. Rogers, and Helen. Sharp, *Interaction design : beyond human-computer interaction*. John Wiley & Sons Ltd, 2015.

[32] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the Accuracy and Robustness of the Leap Motion Controller," *Sensors*, vol. 13, no. 5, pp. 6380–6393, May 2013, doi: 10.3390/s130506380.

[33] Alan. Dix and Soren. Lauesen, *Human-computer interaction. Third edition*. [Prentice Hall], 2007.

[34] D. Abdlkarim *et al.*, "A methodological framework to assess the accuracy of virtual reality hand-tracking systems: A case study with the Meta Quest 2," *Behav. Res. Methods*, vol. 56, no. 2, pp. 1052–1063, Feb. 2023, doi: 10.3758/s13428-022-02051-8.

[35] R. Jota, A. Ng, P. Dietz, and D. Wigdor, "How fast is fast enough?," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA: ACM, Apr. 2013, pp. 2291–2300. doi: 10.1145/2470654.2481317.

[36] L. Hakim, "INTERAKSI 3D SENSOR LEAP MOTION UNTUK SIMULASI MENGGENGGAM BENDA VIRTUAL PLASTIS," INSTITUT TEKNOLOGI SEPULUH NOPEMBER, SURABAYA, 2020.

[37] V. Adhikarla, J. Sodnik, P. Szolgay, and G. Jakus, "Exploring Direct 3D Interaction for Full Horizontal Parallax Light Field Displays Using Leap Motion Controller," *Sensors*, vol. 15, no. 4, pp. 8642–8663, Apr. 2015, doi: 10.3390/s150408642.

[38] D. Bachmann, F. Weichert, and G. Rinkenauer, "Review of Three-Dimensional Human-Computer Interaction with Focus on the Leap Motion Controller," *Sensors*, vol. 18, no. 7, p. 2194, Jul. 2018, doi: 10.3390/s18072194.

[39] S. Nicola and L. Stoicu-Tivadar, "Hand Rehabilitation Using a 3D Environment and Leap Motion Device.," *Stud. Health Technol. Inform.*, vol. 251, pp. 43–46, 2018.

[40] J. Cui and A. Sourin, "Interactive shape modeling using leap motion controller," in *SIGGRAPH Asia 2017 Technical Briefs*, New York, NY, USA: ACM, Nov. 2017, pp. 1–4. doi: 10.1145/3145749.3149437.

[41] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the Accuracy and Robustness of the Leap Motion Controller," *Sensors*, vol. 13, no. 5, pp. 6380–6393, May 2013, doi: 10.3390/s130506380.

[42] J. Guna, G. Jakus, M. Pogačnik, S. Tomažič, and J. Sodnik, "An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking," *Sensors*, vol. 14, no. 2, pp. 3702–3720, Feb. 2014, doi: 10.3390/s140203702.

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.