CARNEGIE MELLON UNIVERSITY

School of Architecture

College of Fine Arts

Thesis

Submitted in Partial Fulfillment of the requirements for the degree of

Master of Science in Computational Design

TITLE:

Responsive Realities

Designing Real-Time, Adaptive Learning Environment for Origami

AUTHOR:

Yiming Jiao

ACCEPTED BY ADVISORY COMMITTEE:

Daragh Byrne	Principal Advisor	May 6th, 2025 DATE
Jimmy Cheng	Advisor	May 6th, 2025 DATE

Responsive Realities

Designing Real-Time, Adaptive Learning Environments for Origami

by

Yiming Jiao

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computational Design

Carnegie Mellon University
School of Architecture, College of Fine Arts
May 2025

Thesis Committee:

Prof. Daragh Byrne, Primary Advisor
Jimmy Wei-Chun Cheng, Secondary Advisor



Abstract

Origami can be intuitive, yet traditional tutorials often fail to convey spatial and sequential steps clearly—especially in complex patterns. This thesis presents an interactive origami learning system that integrates a real-time camera, hand gesture recognition, and projection-based feedback to support embodied and responsive folding experiences.

The system operates in three distinct modes: Create Mode allows users to construct custom crease patterns through freeform folding and gesture confirmation; Learner Mode provides step-by-step projection of predefined models with real-time feedback on fold accuracy; and Auto Mode explores data-driven inference by matching user folds to known patterns and suggesting possible next steps.

Technically, the system tracks paper geometry and hand movement using computer vision, builds a dynamic crease graph, and projects instructions and feedback directly onto the paper surface. Users interact entirely through hand gestures, enabling continuous focus on the material without additional devices.

A Miura-ori pattern was used as a case study to demonstrate the system's capabilities in both Create and Learner Modes. The results show that the system successfully supports fold tracking, feedback projection, and gesture-driven interaction, while revealing design challenges related to fold detection resolution, gesture robustness, and projection alignment.

This work contributes a novel framework for embodied origami learning and offers insights into the integration of computational interaction with craft-based practices.

Acknowledgments

I would like to sincerely thank my advisors for their guidance throughout my thesis. I am especially grateful to my primary advisor, Prof. Daragh Byrne, for his continuous direction and support throughout the entire thesis process. I also thank my secondary advisor, Jimmy Cheng, whose thoughtful feedback greatly contributed to the development of the system. Special thanks to Prof. Daniel Cardoso, who generously agreed to serve as my thesis reader despite being on research leave.

I am deeply grateful to my parents for their unwavering support, and to my colleagues and friends for their encouragement and care along the way. I especially want to thank Qingchen Yu and Keyu Xu, my high school classmates and long-time friends, whose continued presence and support have been truly indispensable.

I also wish to thank the communities and inspirations that supported my mental health during this journey, and those whom I love. In particular, I want to express my appreciation for the virtual idols whose presence provided comfort and strength: A-SOUL (Ava, Bella, Carol, Diana, and Ellen), EOE (Youen, Luzao, Minuo, Waner, and Yumo), Fulaco (Aiyaya, Yinguo, Longshelan, and Chengzhi), as well as Xingtong, Xinyi and Sinuo.

Contents

1 Introduction **9**

1.1 Problem Statement and Research Gap 10

1.2 Personal Motivation 10

1.3 Research Questions 11

1.4 Hypothesis 11

1.5 Scope and Objectives 12

1.5.1 Scope 12

1.5.2 Objectives 12

1.6 Structure of the Thesis 13

2 Background and Literature Review **15**

2.1 Origami as a Learning System 15

2.1.1 Educational Value of Origami 15

2.1.2 Tessellation Origami: Complexity, History, and Contemporary Challenges 17

2.1.3 Traditional and Digital Approaches to Origami Instruction 20

2.2 Human-Computer Interaction in Craft Learning 22

2.2.1 Embodied Interaction and Feedback 22

2.2.2 Interactive Systems for Making and Learning 23

2.2.3 Limitations of Existing Learning Interfaces 24

2.3 Vision-Based Paper Detection and Tracking 25

2.3.1 The Role of Sensors in Responsive Environments 25
2.3.2 Computer Vision for Paper Detection 26
2.4 Crease Pattern Modeling 28
2.4.1 Origami Geometry and Crease Logic 28
2.4.2 Computational Modeling of Crease Patterns 30
2.5 Gesture Recognition in Interactive Systems 31
2.5.1 Overview of Gesture Tracking Technologies 31
2.5.2 Real-Time Hand Tracking 33
2.6 Projection-Based Feedback Systems 34
2.6.1 Projection in Learning and Interaction 34
2.6.2 Responsive Feedback and Adaptive Systems 35
2.7 Summary and Research Gap 36
2.7.1 Gaps and Challenges in Existing Research 36
2.7.2 Positioning This Research 37

3 Methodology **39**

3.1 Overview and Design Rationale 39
3.2 System Architecture 40
3.3 Gesture Recognition 41
3.3.1 Hand Tracking and Normalization 41
3.3.2 Pose Matching and Gesture Confirmation 42
3.4 Paper and Fold Detection 45
3.4.1 Paper Detection via Color Masking and Contour Analysis 45
3.4.2 Corner Ordering and Structural Adjustment 46

3.4.3 Fold Detection and Edge Inference 48
3.4.4 Projection and Feedback Alignment 48
3.5 Crease Pattern Modeling 49
3.5.1 Graph Structure - Qingluan Crease Pattern 49
3.5.2 Fold Matching and Feedback Logic 50
3.6 Mode Design and Interaction Flow 52
3.6.1 Create Mode 52
3.6.2 Learn Mode 53
3.6.3 Auto Mode 53
3.6.4 Unified Interaction Flow 54
3.7 Chapter Summary and Transition 54

4 Experiment Implementation 56

4.1 System Setup Overview 56
4.2 Hardware Setup 56
4.2.1 RGB-D Camera: Orbbec Femto Bolt 57
4.2.2 Projector: Miroir L710S Pro 58
4.2.3 Workspace Setup and Calibration 60
4.3 Software Implementation 61
4.3.1 Technology Stack and Frameworks 61
4.3.2 Code Structure and Module Overview 62
4.3.3 Data Flow Between Components 63
4.3.4 TouchDesigner Integration 64
4.3.5 Real-Time Performance Considerations 65

4.4 User Interface and Projection System 66
4.4.1 Visual Encoding and Feedback Logic 66
4.4.2 Projection Rendering 67
4.4.3 Gesture-Driven Interface Control 68
4.4.4 Mode Awareness and Visual State 69
4.5 QLCP Graph and Dataset Handling 70
4.6 User Journey and Interaction Flow 72
4.6.1 User Journey Overview 72
4.6.2 Start Menu Interface 73
4.6.3 Interaction Modes 75
4.7 Chapter Summary 85

5 Results and Discussion **86**

5.1 Experiment Overview 86
5.2 Create Mode Results 86
5.3 Learner Mode Results 87
5.4 Discussion and Reflections 89
5.5 Limitations 90
5.6 Chapter Summary 91

6 Conclusion and Future Work **93**

6.1 Summary of Contributions 93
6.2 Key Insights and Reflections 94
6.3 Future Work 94
6.4 Final Thoughts 96

List of Figures

2.1 Examples of origami tessellations. From left to right: Miura-ori pattern (mechanical folding), hyperbolic paraboloid (structural geometry), triangle tiling (basic modular design), and a complex tiling by Goran Konjevod (artistic deformation). 18

2.2 Examples of digital origami tools. Left: Origami Simulator allows simulation of 3D folding behavior based on crease patterns. Right: Foldify supports interactive paper toy creation using custom textures and printable templates. 21

2.3 Crease pattern and final folded forms. The diagram on the left shows a full crease pattern (CP) using mountain (red) and valley (blue) folds. On the right are two views of the final folded origami model, illustrating how complex figures emerge from systematic folding logic. 28

2.4 MixFab interface. (a) A user interacts with physical objects through projected digital tools. (b) The system uses depth sensing to capture interactions. (c) Resulting customized objects are fabricated using 3D printing. Image credit: Valkov et al., 2014. 35

3.1 System architecture diagram showing the interaction loop between user gestures, paper detection, crease graph modeling, and real-time projection-based feedback. 40

4.1 Schematic layout of the system. The RGB-D camera captures gestures and paper from above; the projector overlays real-time visual guidance onto the folding surface. 57

4.2 Photograph of the physical workspace showing the mounted camera and projector above a table workspace. 58

4.3 Overhead RGB-D camera (Orbbec Femto Bolt) mounted for paper and hand tracking. 59

4.4 Miroir L710S Pro projector used for real-time visual feedback. 59

4.5 Homography transformation from selected corner points (p_1–p_4) in the camera view to a normalized square model space (p'_1–p'_4). 61

4.6 Diagram of the system's interaction flow across the three modes, converging into continuous feedback through projection and hand input. 73

4.7 Mode selection interface using hand-hover and gesture-based interaction. 75

4.8 Diagram of the Learner Mode workflow. The system continuously senses paper shape and user hand positions, projecting expected creases, and updating feedback in real time. 77

4.8 Continuation of the figure. 78

4.9 Create Mode workflow. The system tracks folds and hand movements, adding new creases to a graph structure and projecting them in real time for visual feedback. 80

4.9 Continuation of the figure. 81

4.10 Conceptual flow of Auto Mode. As users fold, the system dynamically tracks and compares emerging crease graphs to known patterns and suggests future fold paths. 83

4.10 Continuation of the figure. 84

5.1 Create Mode folding sequence. Sixteen manually selected frames from the real-time session showing the progression of freeform folding, gesture-based crease confirmation, and live crease graph projection. 87

5.2 Learner Mode folding sequence. Sixteen representative frames from the recorded session demonstrate guided folding using projected instructions, real-time feedback, and gesture-driven step progression. 88

Chapter 1

Introduction

Origami transforms simple 2D sheets into intricate 3D structures, but the process of learning complex patterns remains challenging. Traditional learning methods such as video, photo, and diagram-based tutorials provide valuable guidance but often fail to accommodate varying skill levels and learning styles. Tessellation origami, in particular, presents unique challenges due to its complex, repetitive patterns, which can make it difficult for learners to track their progress and understand the overall folding structure.

This research proposes the development and evaluation of a real-time, adaptive learning environment designed to dynamically adjust to individual progress and folding patterns. By providing personalized, responsive interaction, the system aims to make tessellation origami more accessible, engaging, and effective. The system will explore new educational practices by adapting guidance to the learner's pace and technique, enhancing both learning outcomes and creative exploration.

Tessellation origami has become a growing trend in the origami world. Unlike traditional origami, which often mimics real-world objects, tessellations explore the pure geometric possibilities of paper. Many origami makers are drawn to this field for its simplicity, the repetitive nature of limited patterns, and the aesthetic beauty of pure geometry. Beyond its artistic appeal, tessellation origami has gained interest in fields like metamaterials(Bullivant, 2006; Overvelde et al., 2016), computational materials (Silverberg et al., 2014), and fabrication (Na et al., 2015; Yang et al., 2015). Its ability to create complex yet highly structured patterns makes it valuable for designing flexible and adaptive materials, influencing advancements in architecture (Tibbits, 2014), robotics (Rus & Tolley, 2015), and material science (Melancon et al., 2021).

1.1 Problem Statement and Research Gap

While significant research has been conducted on origami simulation and modeling, there is limited work on real-time, adaptive learning systems for origami. Existing research primarily focuses on **Vision-based step matching**, where the system compares the learner's progress with a predefined sequence. However, this approach often results in poor adaptability when the learner deviates from the expected steps, especially in complex tessellation patterns.

Additionally, **crease recognition** remains underdeveloped. Most systems are unable to recognize new crease formations in real-time or adjust guidance based on individual variations in folding style. Current systems are also limited in providing corrective feedback when the learner makes a mistake or departs from the expected pattern.

This research aims to fill these gaps by:

- Developing a gesture-based system that can detect and learn new crease formations.
- Creating a graph-based model to track and predict the folding state in real-time.
- Designing a dynamic projection system that provides real-time, adaptive feedback directly onto the paper surface, demonstrated through a preliminary assessment using basic origami tasks to evaluate timing accuracy, alignment precision, and user response.

1.2 Personal Motivation

In my experience with origami tutorials, I have created video and photo-based guides for over a decade. However, I have frequently received feedback from learners struggling to follow tessellation patterns. The repetitive nature of these patterns often causes learners to lose track of progress, and static tutorials fail to provide real-time corrections or adaptive guidance.

This motivated me to explore how real-time feedback and predictive modeling could enhance the origami learning experience. By combining gesture-based recognition, graph-based modeling, and dynamic projection, I aim to create a system that adapts to the learner's individual style and pace, improving both the learning process and creative exploration.

1.3 Research Questions

This research will address the following key questions:

1. What methods can be employed to ensure that the interactive system provides improved fold accuracy, reduced error rates, and enhanced learning efficiency through real-time guidance tailored to the individual progress and feedback of origami learners?
2. What effect does this real-time feedback have on the task performance of ongoing folding for learners?
3. How can the system learn and predict complex tessellation patterns while adjusting to individual user variations and improvisations during the folding process?

1.4 Hypothesis

The underlying hypothesis of this research is that a real-time, adaptive origami learning system based on gesture recognition and crease pattern prediction will:

- Improve fold accuracy and reduce error rates through real-time feedback and adaptive guidance.
- Enhance learning efficiency by dynamically adjusting to the user's individual progress and folding style, where efficiency is defined by observable measures such as reduced completion time, fewer repeated mistakes, and more rapid mastery of fold sequences.

- Increase user engagement by offering responsive, real-time feedback that sustains attention and encourages continued interaction. Engagement will be assessed through indicators such as task completion rates, duration of use, and responsiveness to feedback cues.
- Support creative exploration by allowing users to experiment with alternative fold sequences or customized outcomes. This may be observed through user-initiated deviations from standard folds, variations in crease patterns, or feedback reflecting curiosity and experimentation.

1.5 Scope and Objectives

1.5.1 Scope

This research focuses on tessellation origami, where complex and repetitive patterns make traditional learning methods particularly difficult. The system will be designed to operate within a **worktable-sized responsive environment**, making it suitable for close-range, table-top projection and interaction. The system will be designed to:

- Capture hand gestures and paper transformations using an RGB-D camera.
- Analyze the folding state using a graph-based model.
- Provide real-time adaptive projection of crease lines, next steps, and corrective feedback through a projector.

1.5.2 Objectives

The main objectives of this research are to:

1. Develop a real-time system that recognizes hand gestures and paper transformations.

2. Construct a graph-based model to track and predict crease formation.

3. Design and implement a dynamic projection system to provide real-time guidance and feedback.

4. Evaluate the system's effectiveness in improving user learning, accuracy, and creative exploration.

1.6 Structure of the Thesis

This thesis is organized as follows:

- **Chapter 2: Background** – Reviews literature and foundational concepts relevant to this research, including the pedagogical and computational aspects of origami, prior work on gesture-based interaction and computer vision in learning systems, and the use of graph-based models for crease pattern representation. This chapter also introduces key technical and conceptual precedents in responsive learning environments.

- **Chapter 3: Methodology** – Describes the design and development of the proposed system, including the selection and configuration of the RGB-D camera and projector, the process of capturing user gestures, the use of Crease Pattern for representing folding steps, and the integration of real-time projection-based feedback. This chapter outlines how these components collectively support an adaptive origami learning experience.

- **Chapter 4: Experiment Implementation** – Details the physical setup of the interactive table, the software pipeline using TouchDesigner and Python, and the implementation of the step-matching and crease-recognition algorithms. It also presents the preliminary testing process with a set of basic origami folds, describing how the system responds to user input in real time.

- **Chapter 5: Results and Discussion** – Presents findings from technical and user-oriented evaluations, including system accuracy in gesture recognition and crease matching, user responses to feedback quality, and observations on engagement and learning efficiency. The chapter discusses strengths, limitations, and insights gained through initial deployment.

- **Chapter 6: Conclusion and Future Work** – Summarizes the key contributions of the thesis, including the development of a real-time, adaptive learning tool for origami education. It also outlines future directions, such as integrating predictive models for fold recommendation, scaling to more complex folding sequences, and extending the system to other craft-based learning contexts.

Chapter 2

Background and Literature Review

This chapter reviews prior work and foundational concepts that inform the design of the proposed origami learning system. The literature spans multiple domains, including origami education, gesture-based interaction, projection feedback, and vision-based tracking systems.

The chapter begins by positioning origami—especially tessellation origami—as both a cultural practice and a complex spatial learning activity. It then explores traditional and technology-assisted methods for origami instruction, identifying limitations in current tools and opportunities for real-time, embodied feedback.

Following this, the chapter surveys enabling technologies: gesture recognition for intuitive interaction, projection-based augmented interfaces for spatial guidance, and computer vision approaches for real-time sensing and fold detection. These sections frame the technical foundation of the system.

The final part of the chapter reflects critically on how these domains intersect, outlining key design considerations that shape the system presented in this thesis.

2.1 Origami as a Learning System

2.1.1 Educational Value of Origami

Origami, the art of paper folding, offers more than aesthetic or recreational value—it serves as a powerful educational tool across disciplines. Its inherent structure requires learners to engage in sequential thinking, spatial visualization, and precise manipulation,

making it a valuable medium for developing cognitive and motor skills.

In formal educational settings, origami has been integrated into curricula to support the teaching of mathematics, geometry, and problem-solving. Folding a paper into complex forms requires an understanding of angles, symmetry, and transformation, which naturally reinforces geometric concepts such as congruence, reflection, and tessellation. Research has shown that origami activities promote logical reasoning, enhance mathematical understanding, and provide a tactile, visual way to explore abstract concepts In formal educational settings, origami has been integrated into curricula to support the teaching of mathematics, geometry, and problem-solving. Folding a paper into complex forms requires an understanding of angles, symmetry, and transformation, which naturally reinforces geometric concepts such as congruence, reflection, and tessellation. Research has shown that origami activities promote logical reasoning, enhance mathematical understanding, and provide a tactile, visual way to explore abstract concepts (Boakes, 2008; González & García-Murillo, 2024; Patil & Kshirsagar, 2024). Furthermore, origami enhances spatial reasoning, as learners must mentally predict how flat folds will translate into three-dimensional forms—a process shown to develop visuospatial skills and cognitive flexibility (Fan et al., 2022; Migliorini & Andreani, 2023).

Numerous teaching resources and lesson plans have emerged that demonstrate origami's educational potential across topics such as circle theorems, polyhedra construction, coordinate geometry, and surds. Sites such as Artful Maths and Nrich Maths curate rich activities—including folding polyhedra from nets, investigating Huzita–Hatori axioms, and exploring symmetry through tessellation—that support mathematical inquiry at multiple levels of complexity (Grandi, 2024; Hull, 2006). These resources have been adopted not only for their curricular relevance but also for their ability to foster engagement, collaboration, and creativity in the classroom.

From a developmental perspective, origami has been shown to improve fine motor skills and hand-eye coordination, particularly in early education, as the precise folding movements support bilateral coordination, visual-motor integration, and finger dexterity (Beck, 2021; Rahmawati et al., 2022). The process of folding paper along precise lines

encourages attention to detail, patience, and discipline while offering immediate visual feedback on accuracy and technique.

In informal and self-guided learning environments, origami serves as a medium for exploration, creativity, and iterative learning. Learners can experiment with different forms, test variations, and develop an intuitive understanding of structure and transformation. The low material cost and minimal setup also make origami highly accessible, enabling widespread engagement across diverse learning contexts.

In recent years, origami has gained prominence in STEAM (Science, Technology, Engineering, Arts, and Mathematics) education as a cross-disciplinary tool that fosters both creative expression and analytical reasoning. Its inherently tactile and spatial nature makes it especially suited to embodied learning, allowing students to explore mathematical and scientific principles through hands-on folding, iterative experimentation, and visual problem-solving (Miura et al., 2015).

In summary, origami supports a range of educational outcomes—from conceptual understanding in geometry to the cultivation of soft skills like focus and perseverance—positioning it as a compelling entry point for both structured learning and open-ended exploration. Among the many branches of origami, tessellation origami presents unique educational and design challenges due to its mathematical complexity and recursive folding logic. The following section explores the historical and technical development of tessellation origami, highlighting its relevance to both artistic expression and computational design.

2.1.2 Tessellation Origami: Complexity, History, and Contemporary Challenges

Tessellation origami refers to a class of folding techniques that create repeating geometric patterns across a sheet of paper. These designs often balance precision and expressiveness—ranging from functional, mathematically derived structures like the Miura-ori to highly sculptural or decorative forms. Unlike figurative origami, which aims to represent specific objects or creatures, tessellation focuses on pattern logic, modular repetition,

and material behavior.

This type of folding has been explored in artistic, architectural, and engineering contexts. In art, tessellations allow for nuanced expression through layering and deformation. In design and engineering, they offer programmable material behaviors such as expandability, flexibility, and structural strength.

Figure 2.1 showcases a variety of tessellation styles, illustrating the breadth of this form. From regular geometric tilings to complex, spatially expressive surfaces, these examples underscore the value of origami as both an artistic and technical practice.

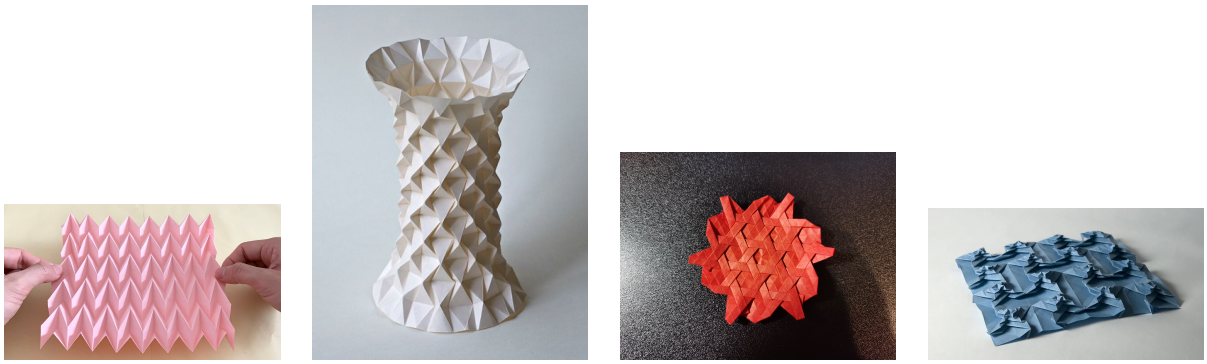


Figure 2.1: Examples of origami tessellations. From left to right: Miura-ori pattern (mechanical folding), hyperbolic paraboloid (structural geometry), triangle tiling (basic modular design), and a complex tiling by Goran Konjevod (artistic deformation).

Tessellation origami, a specialized discipline within the broader field of origami, merges geometric precision with artistic expression by transforming flat sheets into intricate, repeating patterns that seamlessly tile a plane or fold into three-dimensional structures. Unlike traditional origami, which often focuses on representational forms (e.g., animals or flowers), tessellation origami prioritizes the interplay of symmetry, repetition, and mathematical rigor. Its core lies in the systematic arrangement of crease patterns—combinations of mountain and valley folds—that guide the transformation from 2D sheets to complex 3D architectures, such as deployable honeycombs, hyperbolic paraboloids, or non-periodic Penrose-inspired designs.

This art-science hybrid traces its roots to both ancient traditions and modern innovations. Historically, Islamic geometric tiling and Japanese ori-renga (linked folding) laid the groundwork for repetitive folding patterns. In the 20th century, pioneers like Robert

J. Lang formalized its mathematical foundations, demonstrating how crease patterns obey graph theory rules and rigid-foldability principles. These mathematical principles have been instrumental in the development of practical applications, such as Koryo Miura's origami-based foldable satellite panels, which optimize spatial efficiency in aerospace engineering.

Today, tessellation origami transcends aesthetics, enabling applications in aerospace (e.g., foldable satellite arrays), biomedical engineering (e.g., self-assembling stents), and metamaterials research (e.g., structures with tunable mechanical properties). Researchers have explored computational approaches to refining tessellation folding techniques, tackling challenges such as alignment precision, multi-layered symmetry, and recursive fold structures.

The inherent complexity of tessellation techniques—such as managing multi-layered symmetry, aligning recursive folds, and balancing tension across interconnected units—poses unique challenges for learners and designers:

- **Dynamic Geometry:** Unlike static origami models, tessellations often require iterative adjustments. A single misaligned fold in a hexagonal grid can propagate errors across the entire structure.
- **Skill Scalability:** Beginners struggle to interpret abstract crease diagrams, while experts seek to explore non-Euclidean or algorithmically generated patterns (e.g., fractal tessellations).
- **Real-Time Feedback:** Traditional tutorials (e.g., diagrams or pre-recorded videos) fail to address the need for adaptive guidance, such as correcting fold angles or suggesting alternative symmetry axes.

These challenges—particularly the need for dynamic geometric adjustment, scalable instruction, and real-time feedback—highlight the potential value of an interactive learning system that can recognize user gestures, detect fold states, and guide users through adaptive visual cues. While such a system addresses the demands of complex folding tasks

like tessellation origami, it also raises important questions about how origami instruction has traditionally been delivered—and how emerging digital tools have attempted to enhance it. The following section examines both traditional and technology-supported approaches to origami learning, setting the stage for the system proposed in this thesis.

2.1.3 Traditional and Digital Approaches to Origami Instruction

Origami instruction has long relied on traditional methods such as printed diagram books, step-by-step illustrations, and in-person demonstrations. These formats offer learners the ability to follow visual or verbal instructions at their own pace and have been widely used in educational and recreational contexts. However, they often lack responsiveness; the instructions are static and cannot adjust to a learner's pace, errors, or preferences. Misinterpretation of diagrams—especially in more complex folding sequences—can lead to frustration, particularly for novices with limited spatial reasoning experience.

In recent years, digital platforms have emerged to support origami learning through interactive applications, video tutorials, and augmented reality (AR) tools. Platforms such as **YouTube** have popularized origami tutorials that allow learners to follow along in real time with human instructors. Mobile applications like **Foldify(Pixle, 2025)** enable users to design and print 3D paper models, while educational tools such as the **Origami Simulator(Ghassaei, 2016)** by Amanda Ghassaei offer dynamic visualizations of how crease patterns translate into 3D structures. A recent study by Łysakowski et al. (2025) demonstrated the use of augmented reality (AR) integrated with machine learning to guide users through hands-on origami folding. Their system provided real-time feedback using computer vision, validating each folding step and offering dynamic visual support. While the guidance improved user confidence, it also introduced cognitive load—highlighting both the promise and current limitations of interactive AR-based origami instruction (Łysakowski et al., 2025).

Despite these innovations, significant gaps remain. Most digital tools assume a linear,

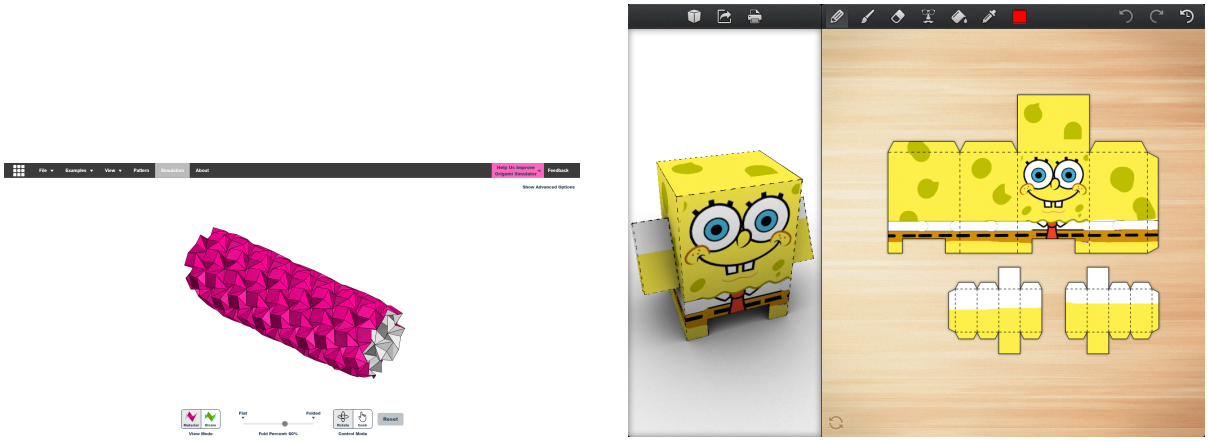


Figure 2.2: Examples of digital origami tools. Left: Origami Simulator allows simulation of 3D folding behavior based on crease patterns. Right: Foldify supports interactive paper toy creation using custom textures and printable templates.

error-free folding process and do not adapt to a user's current folding state or skill level. They typically lack mechanisms for gesture recognition or real-time feedback, offering little support if the user deviates from the expected path. In AR implementations, while the projection of steps enhances spatial understanding, the systems are often limited to fixed tutorials and cannot assess fold quality or user performance.

These limitations underscore the need for a more adaptive approach to origami instruction—one that can detect user progress, respond to variations in folding behavior, and personalize guidance accordingly. The system proposed in this thesis aims to address these gaps by integrating real-time gesture recognition, paper tracking, and dynamic projection to move beyond static, step-by-step instruction. This approach draws on principles from human-computer interaction (HCI), where the design of responsive, embodied, and feedback-rich systems plays a central role in supporting creative and skill-based learning experiences.

2.2 Human-Computer Interaction in Craft Learning

2.2.1 Embodied Interaction and Feedback

Embodied interaction is a foundational concept in human-computer interaction (HCI), emphasizing that learning and cognition are deeply rooted in physical activity, perception, and the body's engagement with the environment (Dourish, 2001; Wilson, 2002). In the context of skill-building tasks—such as craft, handwriting, or folding—motor learning is closely tied to the quality and immediacy of sensory feedback. Learners benefit most when systems can respond in real time, helping them adjust their actions dynamically through visual, auditory, or haptic cues (Schmidt & Lee, 2018; Sigrist et al., 2013).

Traditionally, interactive systems relied on predefined mappings between input and output, offering limited feedback tied to specific triggers. However, advances in sensing and computational modeling have enabled more sophisticated response mechanisms. Responsive environments today often employ machine learning algorithms to analyze user behavior and adapt interactions based on real-time data and prior user performance. These systems can not only react to current input but also anticipate needs, personalize content, and guide learners proactively.

In educational settings, such adaptive systems have been shown to improve learning efficiency by tailoring instruction to a user's pace, comprehension, and progress. For example, in the context of origami instruction, a system might detect when a learner is struggling with a fold—based on hesitation, hand movement patterns, or completion time—and respond by simplifying the step, highlighting key edges, or slowing down the instruction sequence.

These feedback mechanisms embody the principles of embodied interaction: they are situated in physical space, responsive to the learner's activity, and tightly coupled with sensory experience. As such, they are particularly well-suited for learning environments that require fine motor control and spatial reasoning. The interactive origami learning system proposed in this thesis builds on these principles by integrating real-time sensing,

gesture detection, and visual projection to create a responsive, embodied platform for hands-on skill development.

2.2.2 Interactive Systems for Making and Learning

Interactive systems that support hands-on creation—such as drawing, crafting, fabrication, or design—are central to many areas of human-computer interaction (HCI). These systems often aim to augment the user's creative process by providing real-time feedback, adaptive guidance, or new forms of sensory engagement. They draw on a long tradition of research that treats making not just as a productive act, but as a learning process shaped by physical manipulation, experimentation, and reflection.

One influential area is Tangible User Interfaces (TUIs), particularly within the Tangible Media Group (Ishii & Ullmer, 1997). These systems use physical objects as interactive elements, enabling users to manipulate digital information through spatial, embodied actions. Projects like **Topobo** and **SandScape** exemplify how tangible systems support learning through material engagement, particularly in domains like geometry, mechanics, or urban design.

In parallel, initiatives such as **FabLearn** and tangible prototyping toolkits like the **Tower System** have explored how interactive technologies can support making in educational settings. These systems emphasize learner agency, iterative exploration, and feedback-rich environments—often combining physical materials with digital components to scaffold creative and technical skill development (Blikstein, 2013; Mikhak et al., 2002). By making computational processes visible and manipulable through tangible interaction, they help learners develop intuition, correct mistakes, and explore alternative solutions in real time.

A common thread across these systems is their support for multimodal interaction. Whether through gesture, touch, voice, or visual projection, these tools provide users with multiple ways to engage with content. More importantly, they allow the system to perceive and respond to user behavior, creating a feedback loop that enhances both

creativity and precision.

These examples inform the design of the origami learning system presented in this thesis. Like these systems, it aims to create a responsive and multimodal interaction space—blending physical materials, computer vision, and projected feedback—to support the embodied, iterative nature of craft-based learning.

2.2.3 Limitations of Existing Learning Interfaces

While digital tools for creative and educational tasks have advanced significantly, many remain limited in their ability to respond to learners in real time or to account for physical interaction in nuanced ways. Most current origami instruction systems—whether printed diagrams, video tutorials, or app-based guides—are inherently one-directional: they present a sequence of steps without perceiving how the learner interacts with them. These tools assume ideal progress and provide no mechanism for detecting misunderstanding, hesitation, or error.

Even interactive systems that incorporate augmented reality or animated instructions often rely on pre-scripted feedback, offering limited flexibility once a user deviates from the expected path. They typically lack sensory responsiveness—such as visual recognition of folding accuracy or gesture-based control—which constrains their adaptability. In practice, this means that learners must conform to the system's timeline and format, rather than the system adapting to the learner's needs.

These limitations are especially evident in tasks like origami, which involve fine motor control, spatial sequencing, and procedural understanding. Without mechanisms to detect user input or adjust to learner state, current systems fall short of supporting embodied, exploratory, or self-paced learning. This gap points to the need for a new class of perceptual learning interfaces—systems that integrate sensing, recognition, and real-time feedback to guide users more intuitively and adaptively through complex tasks.

The development of such a system requires the convergence of multiple technical components: vision-based paper detection to understand the physical state of the material;

crease pattern modeling to represent folding logic and history; gesture recognition to interpret user intent and performance; and projection-based feedback to provide visual guidance in real time. The following sections review the theoretical foundations and relevant precedents in each of these areas, which together form the basis of the interactive origami learning environment proposed in this thesis.

2.3 Vision-Based Paper Detection and Tracking

2.3.1 The Role of Sensors in Responsive Environments

The capability of responsive environments to interact intelligently with users is largely dependent on their ability to capture accurate and timely data. Sensor technologies play a critical role in this process, acting as the primary interface between users' physical actions and the system's digital responses. They enable the detection of movement, orientation, and interaction patterns, all of which are essential for systems that aim to provide real-time, adaptive feedback.

Early interactive systems employed relatively simple sensors—such as motion detectors, contact switches, or proximity sensors—that could trigger binary outputs based on direct user input. While effective for basic interactions, these approaches offered limited nuance and were insufficient for capturing complex behaviors or interpreting spatial tasks.

As interaction paradigms evolved, so too did the need for more expressive sensing capabilities. Advances in infrared, ultrasonic, and vision-based systems have made it possible to collect rich, continuous streams of spatial data. Among these, **RGB-D cameras**—which simultaneously capture color (RGB) and depth information—have emerged as key enablers of embodied and perceptual interfaces. Devices such as the Microsoft Kinect, Intel RealSense, and Femto Bolt represent a significant leap in sensor technology, offering the ability to track hand movement, object position, and surface geometry in three-dimensional space.

These capabilities have had a transformative effect on a range of applications, including virtual reality, gesture-based gaming, and educational interfaces. In particular, depth cameras can not only detect the presence of individuals within a space, but also map their movements and physical interactions with high fidelity. This level of detail is especially valuable in domains requiring fine motor skills and spatial accuracy—such as adaptive learning environments for origami folding.

In this thesis, depth sensing forms the foundation for real-time paper tracking and interaction detection. It enables the system to observe the physical state of the origami sheet, interpret user gestures, and guide the learning process through context-sensitive visual feedback. A key component of this process is computer vision, which allows the system to detect and track the origami paper dynamically. The following section explores the methods used to perform this detection, including color masking, contour analysis, and shape recognition.

2.3.2 Computer Vision for Paper Detection

Paper detection is a fundamental component in vision-based craft systems, especially for interactive origami learning environments where accurate spatial tracking is necessary for real-time guidance. Broadly, computer vision approaches for detecting paper or fold states fall into two main categories: traditional image processing techniques and data-driven, machine learning–based methods.

Traditional techniques often rely on color segmentation, contour extraction, and geometric approximation. In these pipelines, an image is converted to a more perceptually uniform color space (such as HSV), followed by thresholding to isolate regions of interest—typically based on paper color. From this binary mask, contours are extracted and approximated to polygons, and quadrilaterals are filtered based on area and convexity to represent the paper surface. Homography estimation is then used to align the detected shape with a canonical square, allowing crease patterns or overlays to be projected or compared.

Such approaches are lightweight and interpretable, offering consistent performance under controlled lighting and simple backgrounds. However, they may be sensitive to occlusion (e.g., hands covering the paper), fold-induced deformations, and color variability.

More recent systems adopt machine learning–based object detection to classify and recognize the paper at various fold states. For example, Łysakowski et al. (2025) employed a YOLOv8s model running on a HoloLens 2 headset to detect specific folding steps in real time (Łysakowski et al., 2025). Their system framed each fold configuration as a distinct object class and trained the model on a dataset of annotated paper states. To aid labeling, they used segmentation tools like SAM and Track Anything, and applied color and spatial augmentations to improve generalization.

Similarly, Yamamoto et al. (2022) proposed an AR-based origami instruction system that tracks paper and provides visual overlays via depth sensing and fingertip tracking (Yamamoto et al., 2022). Their system uses a top-mounted camera to identify paper shape and orientation and assists users by projecting folding instructions directly onto the workspace.

Tachibana et al. (2022) also explored real-time step-based recognition by combining visual marker tracking with projection feedback (Tachibana et al., 2022). Their prototype recognizes each step of a predefined folding sequence and advances the instruction accordingly, but requires fixed paper positions and predefined paper sizes.

These three precedents illustrate the range of strategies—from vision-only to marker-assisted and projection-based—currently used in origami learning interfaces, each with trade-offs in robustness, flexibility, and user freedom.

These two approaches illustrate a key trade-off in paper detection strategies: between precision and flexibility. Traditional methods are easy to implement and interpret, but may struggle in complex or unstructured settings. Learning-based methods offer robustness and pattern recognition, but often require curated datasets, longer setup times, and more constrained interaction models. Together, they demonstrate the diversity of vision-based techniques for paper detection, and highlight the importance of selecting methods that balance responsiveness, reliability, and adaptability for craft-based learning environ-

ments.

2.4 Crease Pattern Modeling

2.4.1 Origami Geometry and Crease Logic

Crease patterns (CPs) 2.3 provide a foundational representation of origami forms, describing how a two-dimensional sheet can be transformed into a complex three-dimensional structure. A crease pattern is typically composed of a set of lines indicating folds, along with annotations that specify their types—most commonly as *mountain* folds (convex) or *valley* folds (concave). These annotations encode directional and geometric information that governs how the paper will behave when folded.

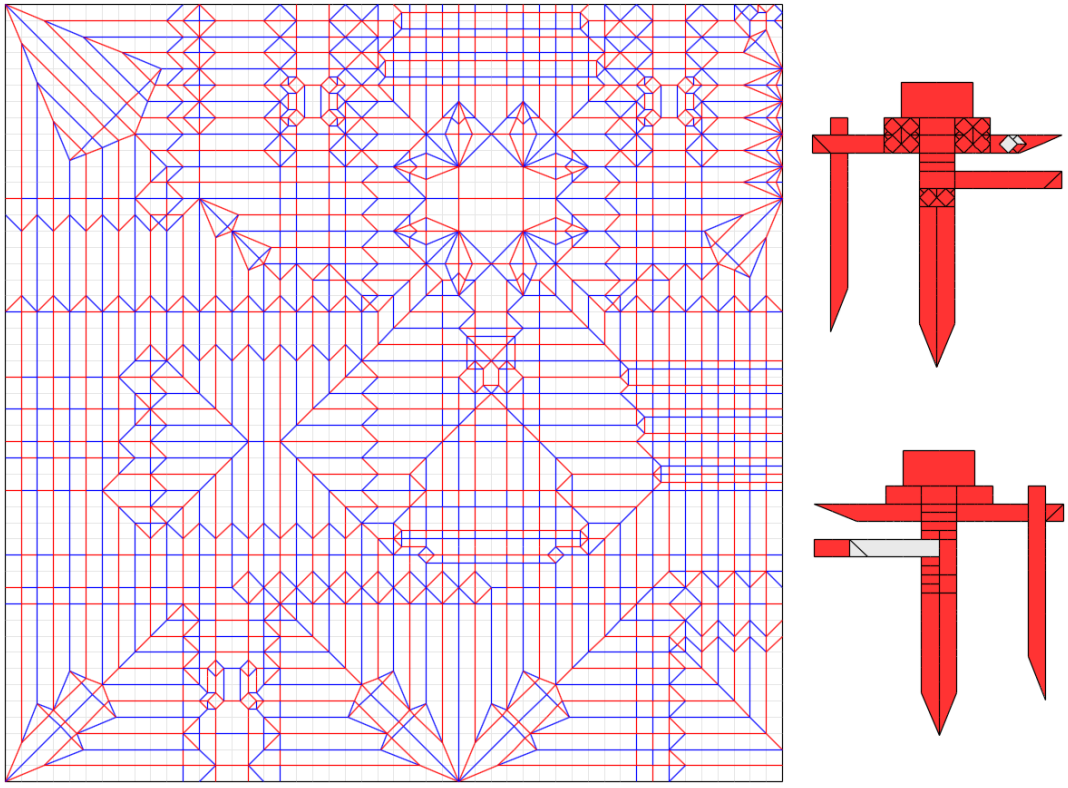


Figure 2.3: Crease pattern and final folded forms. The diagram on the left shows a full crease pattern (CP) using mountain (red) and valley (blue) folds. On the right are two views of the final folded origami model, illustrating how complex figures emerge from systematic folding logic.

The mathematical study of crease patterns has evolved significantly over the past several decades, particularly in the context of computational origami. Early work in this area formalized key geometric constraints, such as Kawasaki's Theorem (which governs flat-foldability at a vertex) and Maekawa's Theorem (which relates the number of mountain and valley folds at a single vertex). These principles help determine whether a given crease pattern can fold into a flat or rigid structure without tearing or overlapping.

Beyond individual creases, crease patterns can be understood as graph structures, where vertices represent intersections of folds and edges represent crease lines. This graph-based view enables both analysis and algorithmic manipulation. For instance, software such as ORIPA models crease patterns as planar graphs and validates them based on origami axioms and local flat-foldability rules (Jun Mitani, 2014). These representations support tasks such as procedural generation, simulation, and step-wise validation. For instance, Robert Lang's work has formalized techniques for designing crease patterns that fold into complex, representational forms, bridging mathematical theory with practical design applications (Lang, 2012). In the context of simulation and physical behavior, Tachi's work on rigid origami explores how crease patterns can be modeled and deformed while maintaining panel rigidity, enabling real-time computational folding environments (Tachi, 2009).

In educational or design contexts, crease patterns serve both as a blueprint and as an interface for learning. For simple origami models, CPs may be sparse and intuitive, but for more complex designs—such as tessellations or curved folds—the crease network becomes highly intricate and abstract. In such cases, the ability to model, visualize, and interact with crease structures becomes essential not only for fabrication but also for pedagogical support.

This thesis builds upon these geometric and logical foundations by treating crease patterns as structured data that can evolve over time, reflecting user interaction and fold progression. While existing systems tend to treat folding as a sequence of static steps, modeling the underlying crease logic enables more flexible, adaptive, and personalized feedback mechanisms.

2.4.2 Computational Modeling of Crease Patterns

Computational modeling of crease patterns plays a central role in enabling origami design, simulation, and analysis. While traditional crease patterns are drawn manually, computational approaches formalize these structures as data representations that support validation, transformation, and interaction.

One common strategy is to model crease patterns as planar graphs, where edges represent crease lines and vertices represent fold intersections. These graph structures can be enriched with metadata—such as fold type (mountain or valley), target angle, and sequence order—to support downstream tasks like animation, fabrication, or error checking. Tools such as ORIPA use this representation to validate crease patterns based on origami axioms and local foldability rules.

For more dynamic modeling, systems like Amanda Ghassaei's Origami Simulator take crease graphs as input and simulate folding behavior in real time using spring-based energy minimization techniques (Ghassaei, 2016). This allows users to visualize how a pattern transforms from a flat sheet into a three-dimensional structure, providing valuable feedback for complex or abstract models.

Other tools, such as Treemaker and ReferenceFinder by Lang, and rigid-foldability simulators developed by Tachi, allow users to design or analyze crease patterns procedurally—often using geometric constraints and optimization to generate efficient fold layouts (Lang, 2012; Tachi, 2009).

Despite their power, many of these systems are optimized for design or simulation, rather than interactive learning. They often assume an idealized, error-free folding process and do not dynamically update the crease model based on user interaction. In educational contexts, however, a more responsive model is required—one that can evolve with the learner's actions, track partial folds, and support adaptive guidance.

This thesis builds on the graph-based representation of crease patterns by integrating it into a real-time interactive system. By tracking user folds as graph updates, the system maintains a live model of the origami state, allowing it to offer targeted visual feedback

and predictive guidance during the folding process.

2.5 Gesture Recognition in Interactive Systems

2.5.1 Overview of Gesture Tracking Technologies

Gesture recognition is a foundational technique in human-computer interaction (HCI), enabling systems to interpret physical movement as input for control, navigation, or creative expression. By observing and interpreting body movements—especially of the hands and arms—gesture-based systems support a more natural, embodied form of interaction that aligns closely with physical tasks such as folding, drawing, manipulating objects, or spatial navigation.

Gesture tracking methods can be broadly classified into **marker-based** and **markerless** approaches. Marker-based systems rely on physical tags (such as QR codes, ArUco markers, or reflective spheres) attached to the user's body or tools. These markers are detected through computer vision or infrared systems, enabling precise tracking of position and orientation. Marker-based systems offer high accuracy and reliability, but they require controlled environments and introduce constraints on user freedom, often making them less suitable for open-ended or creative activities.

In contrast, **markerless systems** use computer vision or depth sensing to detect and track body parts directly. Examples include skeletal tracking via depth cameras (e.g., Microsoft Kinect), or landmark-based hand tracking using convolutional neural networks (e.g., MediaPipe Hands). These systems offer greater flexibility and reduce setup requirements, making them more scalable for educational and consumer applications. Markerless tracking is especially useful in contexts where natural hand gestures are central to the interaction, such as VR/AR, performance interfaces, or gesture-based learning platforms.

Gesture recognition has been employed in a wide range of creative and learning contexts. In design tools, gestures are used to manipulate digital content directly; in performance systems, they control sound, light, or visual effects; and in educational platforms, they provide an intuitive means for learners to interact with spatial or procedural tasks. Research has demonstrated that gesture-based input can enhance user engagement, reduce cognitive load, and facilitate more fluid task progression in hands-on learning environments. For instance, studies have shown that incorporating gestures into educational activities can create an enjoyable and interactive learning environment, positively impacting student performance. Additionally, the use of gestures has been associated with improved learning outcomes without significantly increasing cognitive load. (Antle et al., 2009; Sheu et al., 2013).

In the domain of origami instruction, gesture tracking opens up the possibility for systems to detect fold intent, evaluate folding style, and provide feedback non-verbally—bridging physical interaction with digital guidance in a hands-free, embodied manner. The fol-

lowing sections explore how gesture recognition technologies have been implemented in relevant systems and how they support adaptive, interactive learning.

2.5.2 Real-Time Hand Tracking

Hand tracking is a key technique in gesture-based interfaces, enabling systems to detect, localize, and interpret hand movements in real time. Unlike traditional input devices, hand tracking allows for direct, embodied interaction, which is especially well-suited for spatial and craft-based tasks such as origami folding. In the context of interactive systems, effective hand tracking must be both temporally responsive and spatially precise, offering frame-by-frame updates on hand position and gesture configuration.

Recent advances in computer vision have enabled accurate, real-time hand landmark detection using only RGB video input. Among the most widely adopted frameworks is **MediaPipe Hands**, developed by Google. MediaPipe detects 21 anatomical landmarks per hand—including fingertips, joints, and the wrist—providing a detailed representation of hand pose in both 2D and 3D coordinates. The system supports simultaneous tracking of left and right hands, with built-in handedness classification and gesture stability filtering. Because it operates in real time on CPUs and mobile devices, it offers high efficiency and portability, making it suitable for educational and interactive installations.

The output from MediaPipe Hands can be used in a wide variety of gesture-based control scenarios: navigating interfaces, triggering actions, adjusting visual parameters, or guiding instructional sequences. In learning environments, it enables systems to interpret user intent based on hand position, pose, or motion without requiring physical contact or external devices. As such, hand tracking opens up the possibility for fully hands-free interaction—critical for applications where users' hands are already occupied with physical materials, such as in origami learning systems.

2.6 Projection-Based Feedback Systems

2.6.1 Projection in Learning and Interaction

Projection-based interfaces have emerged as powerful tools in interactive systems, especially for tasks that involve physical making, spatial reasoning, or procedural learning. By overlaying digital content directly onto physical surfaces, projection enables a unique form of *spatial feedback*—guiding user actions in real time, exactly where interaction occurs. This supports more natural, embodied workflows and reduces cognitive load by keeping attention focused on the task.

In educational and design contexts, projection has been used to guide manual actions, visualize sequences, and provide step-by-step support. For example, Lee and Hudson's **LightGuide** system projects visual annotations onto handheld tools, helping users complete physical assembly and maintenance tasks by displaying relevant information directly in their field of operation (Lee & Hudson, 2004). Similarly, **MixFab** 2.4 integrates projection into a mixed-reality fabrication workspace, enabling users to combine digital design with real-world materials by projecting design constraints, instructions, or alignment guides onto the physical workspace (Weichel et al., 2014).

These systems demonstrate how projection can bridge digital instructions with physical tasks, providing learners or makers with co-located, real-time guidance. In comparison to screen-based feedback, projection allows users to maintain physical engagement and visual continuity throughout the task—particularly valuable in processes that involve fine motor control and sequential manipulation.

Origami folding presents an ideal use case for projection-based interaction. The ability to align crease patterns, fold directions, or gesture cues directly onto the paper surface allows the system to communicate instructional intent without diverting the learner's attention. This modality supports embodied learning, minimizes disruptions, and helps reinforce spatial understanding—especially in complex folding scenarios such as tessellation origami.

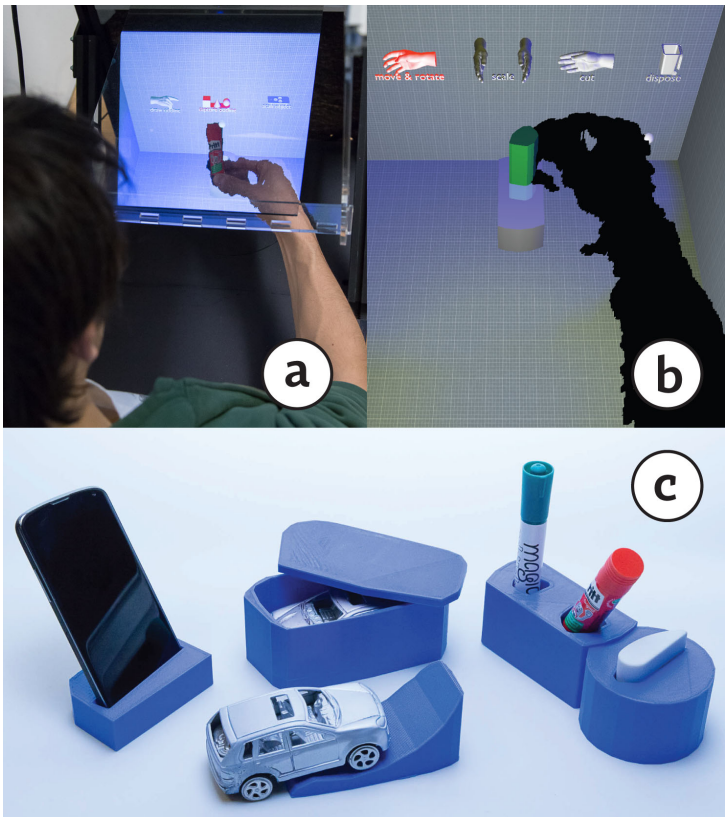


Figure 2.4: MixFab interface. (a) A user interacts with physical objects through projected digital tools. (b) The system uses depth sensing to capture interactions. (c) Resulting customized objects are fabricated using 3D printing. Image credit: Valkov et al., 2014.

2.6.2 Responsive Feedback and Adaptive Systems

Once user input is captured and processed, the next critical component of a responsive environment is the system's ability to generate meaningful, context-aware responses. Early interactive systems typically relied on fixed mappings between inputs and outputs—triggering predefined responses for specific actions. While effective for basic automation, these systems lacked the ability to accommodate variation in user behavior, learning pace, or task complexity.

With the integration of more advanced computational models and machine learning techniques, response systems have become increasingly adaptive and personalized. Modern systems can analyze behavioral data to identify user needs and respond in real time—adjusting instructions, visual feedback, or task pacing based on prior interactions

and performance trends. These adaptive capabilities enable systems to not only react but also anticipate, creating more fluid and learner-sensitive interactions.

In educational contexts, such responsive feedback has been shown to improve learning efficiency, user engagement, and task retention. Systems that adapt to a user's real-time performance—whether through error detection, pace adjustment, or multimodal feedback—allow learners to progress at their own speed while receiving personalized support when needed (Kapur & Bielaczyc, 2012; Vavoula & Sharples, 2009).

In the domain of origami instruction, this adaptivity is especially valuable. Folding tasks are spatial, sequential, and sensitive to small deviations. A responsive system can detect signs of difficulty—such as hesitation, repeated gestures, or extended time on a particular step—and respond by simplifying visual cues, repeating instructions, or offering corrective overlays. This form of feedback aligns with embodied learning principles, allowing users to remain immersed in the physical task while receiving non-intrusive, real-time support.

The convergence of sensor technology and intelligent feedback mechanisms forms the backbone of many modern interactive learning environments. By combining continuous perception with adaptive response strategies, such systems create interactive platforms that support hands-on, perceptual learning across a variety of domains—particularly in complex, spatially grounded tasks like origami folding.

2.7 Summary and Research Gap

2.7.1 Gaps and Challenges in Existing Research

Although significant progress has been made in origami-related education tools, interactive learning systems, and computational folding simulations, several limitations remain—particularly when applied to hands-on, responsive learning contexts. Across systems like the Origami Simulator (Ghassaei, 2016), Foldify, and projection-based AR platforms (Łysakowski et al., 2025; Tachibana et al., 2022; Yamamoto et al., 2022), a

number of recurring challenges become evident:

- **Lack of real-time or adaptive feedback:** Most systems operate as static, linear instruction tools with limited responsiveness to user progress, hesitation, or error.
- **Gesture recognition underused in origami learning:** While gesture-based interaction is common in broader HCI applications, it has rarely been used to interpret folding intent, detect hesitation, or trigger instructional feedback in origami-specific systems.
- **Instructional systems often assume ideal behavior:** Existing tutorials and AR guides typically rely on predefined sequences and rigid camera viewpoints, failing to accommodate variation in folding technique, speed, or skill level.
- **Crease modeling disconnected from physical paper:** Tools like Origami Simulator or ORIPA model crease logic computationally, but do not connect these representations to real-world, evolving user folds.
- **Few systems use projection to guide real folding:** Although projection has been explored in design and fabrication contexts (Lee & Hudson, 2004; Weichel et al., 2014), few systems project crease patterns or feedback directly onto paper surfaces to support physical origami learning in real time.

These limitations point to a need for a new class of origami learning systems—ones that treat user input as dynamic, multimodal, and embodied; that use sensing and modeling to maintain real-time awareness of folding progress; and that deliver context-aware feedback aligned with the user's actions.

2.7.2 Positioning This Research

This thesis responds to these challenges by proposing a vision-based, real-time origami learning system that integrates four key components:

- **Gesture-based control** using MediaPipe Hands for detecting user pose, hesitation, and confirmation gestures
- **Real-time paper and fold tracking** through color segmentation, contour detection, and spatial alignment using RGB-D sensing
- **Graph-based crease modeling (CP)** that evolves with user input, allowing the system to understand fold history and structural intent
- **Projection-mapped, adaptive feedback** that visually guides the user directly on the paper surface, reinforcing spatial and sequential understanding

Together, these components constitute a responsive origami learning environment that bridges the gap between physical craft and digital intelligence. Unlike prior systems that treat learning as linear or static, this system maintains continuous awareness of the learner's context and progress—offering visual feedback, gesture-based control, and crease-aware guidance as the folding process unfolds.

Chapter 3

Methodology

3.1 Overview and Design Rationale

This chapter presents the methodology behind the development of a real-time, interactive origami learning system designed to support hobbyist learners through visual projection, gesture-based interaction, and responsive feedback. This system is grounded in the principles of embodied learning, spatial reasoning, and exploratory making. The goal is not to accelerate folding, but to clarify ambiguous steps, reduce cognitive switching, and enable learners to remain immersed in the physical craft.

The system is organized around three core interaction modes—*Create*, *Learn*, and *Auto*—each supporting different levels of structure and guidance. In *Create Mode*, users add creases freely on the paper surface, using gesture controls to project and refine crease graphs. In *Learn Mode*, the system projects predefined crease patterns and provides real-time visual feedback on fold accuracy. In *Auto Mode*, user folds are detected without predefined goals, and the system dynamically matches the current graph to a dataset of known models, offering suggestions or divergence detection as the user continues to fold.

This methodology integrates computer vision, graph-based modeling, and projection-based feedback into a unified interaction loop. Real-time hand tracking using MediaPipe enables gesture-based control, while paper detection and crease recognition maintain awareness of folding progress. All components are designed to minimize cognitive interruptions, preserve spatial continuity, and support an intuitive, learner-driven origami experience. Figure 3.1 illustrates the overall system architecture, including key modules and their data flow. It shows how user gestures and paper folds are processed through

parallel sensing modules and fused into a live crease graph, which then informs the projection feedback loop.

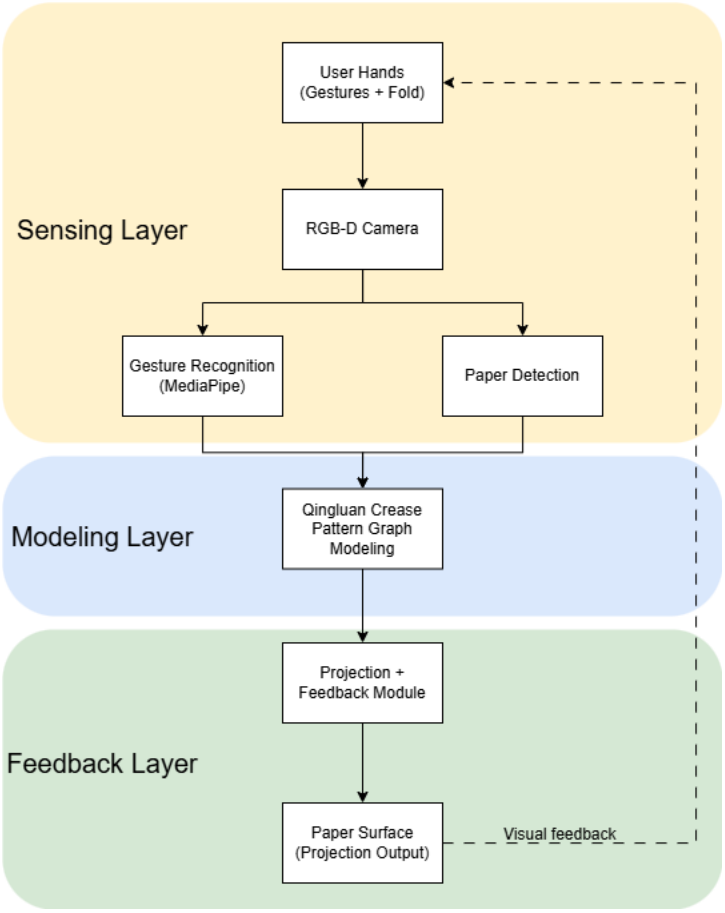


Figure 3.1: System architecture diagram showing the interaction loop between user gestures, paper detection, crease graph modeling, and real-time projection-based feedback.

The following sections detail the system's architecture and the key modules that enable this experience: gesture recognition, crease pattern modeling, projection logic, and the structure of each interaction mode.

3.2 System Architecture

The system architecture consists of four primary modules that work in real time to support gesture-based interaction, fold tracking, crease modeling, and visual projection. These modules include: gesture recognition, paper detection and tracking, crease graph

modeling, and projection output. Figure 3.1 in the previous section provides a visual overview of the complete data and interaction loop.

At the input layer, an RGB-D camera captures both the user's hand gestures and the paper surface. These visual streams are processed in parallel. One stream is sent to the gesture recognition module, which uses MediaPipe Hands to detect 21 hand landmarks and identify gestures through pose normalization and statistical matching. The other stream is used by the paper detection module, which isolates and tracks the square paper sheet in real time, identifying new folds and updating its position.

Data from both modules are passed into the crease graph modeling module. This module maintains a dynamic graph-based structure (QLCP) that encodes each crease as an edge and each fold event as a graph update. The model supports feedback on fold accuracy, progression, and matching against known models.

The updated graph is passed to the projection module, which visually overlays crease lines and system feedback directly onto the paper surface. Projection is performed using a calibrated projector that aligns with the camera's detected quadrilateral, enabling spatially accurate augmentation in the user's workspace.

All components are designed to run continuously in a real-time loop, minimizing latency and supporting uninterrupted, embodied interaction. The following sections will describe each of these modules in detail, starting with the gesture recognition system.

3.3 Gesture Recognition

3.3.1 Hand Tracking and Normalization

To support gesture-based control without interrupting the hands-on nature of folding, the system uses real-time hand tracking to detect, monitor, and interpret user gestures. This is implemented using MediaPipe Hands, a lightweight and high-performance computer vision framework developed by Google. MediaPipe detects 21 anatomical landmarks per hand—including wrist, joints, and fingertips—and returns their normalized 2D and 3D

positions in camera space.

Once the raw hand landmarks are detected, the system performs a normalization procedure to ensure that gestures can be consistently recognized regardless of the hand's position, scale, or in-plane rotation. This normalization enables reliable comparison between saved gestures (learned poses) and current input.

The normalization pipeline proceeds as follows:

1. **Translation:** The wrist landmark (landmark 0) is translated to the origin, effectively centering the hand.

2. **Rotation:** The vector from landmark 0 to landmark 1 (typically the base of the index finger) is rotated to align with the positive x-axis using a 2D rotation matrix:

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

   where $\theta = \arctan 2(y_1 - y_0, x_1 - x_0)$.

3. **Scaling:** The full set of landmarks is scaled so that the maximum distance from the origin is normalized to 1. This ensures gesture comparisons are scale-invariant.

The resulting set of normalized hand landmarks is used for both recording new gestures and comparing live input. Each hand is treated independently, and the system supports simultaneous tracking of both left and right hands. Details of how these normalized landmarks are used for pose comparison are provided in Section 3.3.2.

3.3.2 Pose Matching and Gesture Confirmation

Once hand landmarks are normalized, the system uses them to detect specific poses that control interaction modes or confirm user intent. These poses—referred to as *gesture attitudes*—are learned through a short recording process and later matched during interaction using a statistical comparison approach.

Pose Learning Phase

When learning a new pose, the user is prompted to hold the intended gesture steady. The system allocates a 1-second preparation window followed by a 3-second recording period. During this period, it captures the normalized positions of each of the 21 landmarks for every frame. These temporally collected points are used to model each landmark as a 2D Gaussian distribution. Let $\mathbf{x}_i^{(t)} \in \mathbb{R}^2$ represent the normalized position of landmark i at time t, where $t = 1, \ldots, T$ over the recording duration. Then for each landmark i, the system estimates:

$$\boldsymbol{\mu}_i = \frac{1}{T} \sum_{t=1}^{T} \mathbf{x}_i^{(t)} \quad \text{and} \quad \Sigma_i = \frac{1}{T-1} \sum_{t=1}^{T} (\mathbf{x}_i^{(t)} - \boldsymbol{\mu}_i)(\mathbf{x}_i^{(t)} - \boldsymbol{\mu}_i)^\top$$

This yields a mean vector $\boldsymbol{\mu}_i$ and a 2×2 covariance matrix Σ_i per landmark, capturing the spatial distribution of the pose during the learning window. These per-landmark distributions form the full learned representation of the gesture. :

$$\mathcal{P}_i = \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i), \quad i = 1, \ldots, 21$$

Each hand pose is thus represented as a set of 21 distributions—one per landmark—capturing the natural variation and stability of the user's gesture during the learning phase.

Live Pose Matching

During real-time operation, the system compares each frame's current normalized pose (21 landmark points) against the learned distributions. A similarity score is computed per landmark, and poses are evaluated frame-by-frame against the learned statistical model of the gesture.

Multiple statistical similarity metrics are used to compare the current pose against the learned distributions. These include:

KL Divergence (Kullback–Leibler): Measures the information loss when using dis-

tribution Q to approximate P. For two multivariate Gaussians $P(\boldsymbol{\mu}_P, \Sigma_P)$ and $Q(\boldsymbol{\mu}_Q, \Sigma_Q)$:

$$D_{\mathrm{KL}}(P \parallel Q) = \frac{1}{2}\left[\mathrm{tr}(\Sigma_Q^{-1}\Sigma_P) + (\boldsymbol{\mu}_Q - \boldsymbol{\mu}_P)^\top \Sigma_Q^{-1}(\boldsymbol{\mu}_Q - \boldsymbol{\mu}_P) - d + \ln\frac{|\Sigma_Q|}{|\Sigma_P|}\right]$$

where $d = 2$ for 2D landmarks.

Overlap Area: Approximates the overlapping region between two distributions using Monte Carlo integration:

$$\text{Overlap} \approx \frac{1}{N}\sum_{i=1}^{N} \min\left[\mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_P, \Sigma_P), \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_Q, \Sigma_Q)\right]$$

where $\mathbf{x}_i \sim P$ are samples drawn from distribution P.

Mahalanobis Distance: Measures standardized distance between two mean vectors:

$$d_{\mathrm{M}} = \sqrt{(\boldsymbol{\mu}_P - \boldsymbol{\mu}_Q)^\top \Sigma^{-1}(\boldsymbol{\mu}_P - \boldsymbol{\mu}_Q)}$$

where Σ may be set to Σ_P, Σ_Q, or $\frac{1}{2}(\Sigma_P + \Sigma_Q)$.

Hotelling's T^2 Statistic: Tests the significance of the difference between two multivariate distributions:

$$T^2 = \frac{n_P n_Q}{n_P + n_Q}(\boldsymbol{\mu}_P - \boldsymbol{\mu}_Q)^\top \Sigma_{\text{pooled}}^{-1}(\boldsymbol{\mu}_P - \boldsymbol{\mu}_Q)$$

with pooled covariance:

$$\Sigma_{\text{pooled}} = \frac{(n_P - 1)\Sigma_P + (n_Q - 1)\Sigma_Q}{n_P + n_Q - 2}$$

Bhattacharyya Distance: A symmetric measure of distribution overlap:

$$D_B = \frac{1}{8}(\boldsymbol{\mu}_P - \boldsymbol{\mu}_Q)^\top \Sigma^{-1}(\boldsymbol{\mu}_P - \boldsymbol{\mu}_Q) + \frac{1}{2}\ln\left(\frac{|\Sigma|}{\sqrt{|\Sigma_P||\Sigma_Q|}}\right)$$

where $\Sigma = \frac{1}{2}(\Sigma_P + \Sigma_Q)$.

Each landmark's score is computed individually, and the system aggregates them across all 21 points. A match is confirmed if the aggregated score remains below a threshold in at least 80% of frames within a 1-second sliding window. This threshold is currently fixed as a heuristic based on pilot testing, though future versions of the system may learn or adapt this parameter based on user variability. This approach balances real-time responsiveness with robustness against brief noise or jitter.

Gesture-to-Action Mapping

Confirmed poses are associated with system-level actions such as:

- Switching between interaction modes (Create, Learn, Auto)
- Confirming a fold step or triggering undo
- Initiating a new graph or resetting the projection

These mappings are designed to minimize disruption and preserve continuous, hands-on interaction with the paper. Implementation details of the matching loop and scoring thresholds are provided in Section 4.3.

3.4 Paper and Fold Detection

This section describes how the system detects the origami paper in the video stream and tracks fold events over time. These capabilities form the basis for crease graph updates and real-time projection feedback.

3.4.1 Paper Detection via Color Masking and Contour Analysis

To robustly detect the origami paper surface under varying lighting conditions and fold states, the system uses a custom function, `get_white_mask()`, to isolate high-brightness,

low-saturation regions. This function:

- Applies a Gaussian blur to reduce noise and small texture variations.
- Converts the image from BGR to HSV color space.
- Thresholds pixels where the saturation is below 60 and the value (brightness) is above 180:

$$\text{Mask}(x, y) = \begin{cases} 1 & \text{if } S(x, y) < 60 \text{ and } V(x, y) > 180 \\ 0 & \text{otherwise} \end{cases}$$

This method provides a reliable binary mask of the paper, even when creased or partially folded.

After generating the mask, Canny edge detection is applied, and contours are extracted using `cv2.findContours()`. Each candidate contour is approximated with `cv2.approxPolyDP()`, and the system selects the largest contour with 3–5 corners that falls within a reasonable area range (typically 5,000 pixels to 90% of the frame size). This process accommodates the shifting shape of the paper as folds are introduced, while maintaining stability across frames.

3.4.2 Corner Ordering and Structural Adjustment

Once the corners of the detected paper are found, they are ordered clockwise based on angular direction around the centroid:

$$\theta_i = \arctan 2(y_i - y_c, x_i - x_c)$$

where (x_c, y_c) is the centroid and (x_i, y_i) is the ith corner point. Sorting these angles produces a consistent clockwise ordering.

Based on the number of detected points, the system infers missing corners or adjusts shapes as follows:

Handling 3-Point (Triangle) Contours

Input: 3 ordered corner points: p_1, p_2, p_3

Goal: Infer a fourth point p_4 to form a square

```
p4 = p1 + (p3 - p2)
```

Handling 5-Point (Pentagon) Contours

Input: 5 corner points and known paper length L

Goal: Identify 3 stable points and infer the missing fourth

```
for each group of 3 consecutive points:
    if edge lengths  L:
        p1, p2, p3 = those points
        p4 = p1 + (p3 - p2)
```

Adjusting Imperfect 4-Point Squares

Input: 4 corner points and paper length L

Goal: Snap edges to expected length

```
for each edge i:
    if abs(length[i] - L) > threshold:
        p[i] = point_along(p[i-1], p[i], L)
```

This correction process ensures consistent spatial alignment of the paper representation, even when folds distort the shape. It is crucial for maintaining accurate projection alignment and graph matching downstream.

3.4.3 Fold Detection and Edge Inference

To detect new folds, the system compares the current corner configuration $\mathbf{p}_i^{(t)}$ to the previous frame $\mathbf{p}_i^{(t-1)}$, and computes displacement for each point:

$$\Delta_i = \|\mathbf{p}_i^{(t)} - \mathbf{p}_i^{(t-1)}\|$$

The top two indices with the largest Δ_i values define the candidate fold edge.

Once identified, the two points are mapped to model space using a homography transform H from the detected quadrilateral to the normalized square:

$$\mathbf{x}_{\text{model}} = H\mathbf{x}_{\text{paper}}$$

A gesture-based trigger (e.g., both hands held at screen edges) confirms the addition of the fold to the crease graph (QLCP), avoiding false positives.

3.4.4 Projection and Feedback Alignment

All edges in the QLCP graph are projected back onto the physical paper using OpenCV's `getPerspectiveTransform()` and `perspectiveTransform()`. The projected creases are drawn with distinct visual cues depending on their matching status, ensuring the user receives continuous and accurate feedback.

Further details on graph structure and interaction modes are presented in Sections 3.5 and 3.6.

3.5 Crease Pattern Modeling

3.5.1 Graph Structure - Qingluan Crease Pattern

To represent origami creases in a computationally flexible and incrementally updatable format, this system uses a custom graph structure called the *Qingluan Crease Pattern* (QLCP). QLCP encodes each crease as an edge and each significant point—such as paper corners, fold intersections, and endpoints—as a vertex. This structure supports real-time modification, feedback, and comparison as the user folds.

Formally, the crease pattern is defined as an undirected graph:

$$G = (V, E)$$

where:

- V is the set of 2D vertex positions in normalized paper coordinates (e.g., $[0, 1] \times [0, 1]$).
- $E \subseteq V \times V$ is the set of crease edges connecting vertex pairs.

Each edge $e \in E$ may contain additional metadata:

- *Direction vector:* Encodes the orientation of the fold.
- *Fold type:* (e.g., inferred or matched; used in Learn Mode feedback)
- *Time of creation:* Helps sequence fold steps or detect recency.
- *Confidence score:* Optional metric for Auto Mode matching.

QLCP begins with an initial quadrilateral graph representing the paper's boundary. As folds are detected, new vertices and edges are appended to the graph. Vertices are either snapped to existing graph points (if within a small tolerance) or inserted as new

nodes. This allows the graph to remain geometrically consistent and adaptable to a wide range of folding behaviors, from crisp models to imprecise or improvisational folds.

During interaction:

- In **Create Mode**, each user-defined crease directly updates the graph.
- In **Learn Mode**, the system compares detected creases to a reference QLCP and annotates them using visual cues (see Section 3.6.2).
- In **Auto Mode**, the current graph is compared to a dataset of known QLCPs to infer possible model matches (see Section 3.6.3).

Runtime updates to the QLCP structure are handled incrementally. Implementation details, including how vertex snapping and crease confirmation are performed, are described in Section **??**.

3.5.2 Fold Matching and Feedback Logic

Fold matching is central to the system's ability to provide real-time, visual feedback during the folding process. Once a new crease is detected and added to the QLCP graph, the system compares it against a target set of reference creases—either from a selected model in Learn Mode or dynamically inferred in Auto Mode. This comparison determines whether the user's fold aligns with the intended design and enables projection-based cues to guide or correct user behavior.

Matching Criteria

Each reference crease is defined as an edge in the reference QLCP with known endpoints and direction. To evaluate a newly detected fold $e = (v_1, v_2)$, the system checks for alignment against any target edge $e' = (v_1', v_2')$ using three criteria:

- **Spatial proximity:** The endpoints of e must lie within a fixed Euclidean distance threshold δ of the corresponding endpoints in e'.

- **Directional similarity:** The orientation vectors of e and e' must be within an angular threshold θ.
- **Length tolerance:** The total edge length should be approximately equal (within ratio ϵ).

If a detected fold meets these criteria, it is marked as a match to a reference crease.

Feedback Encoding

Matched and unmatched folds are encoded using distinct projection colors:

- **Blue** lines indicate folds that match the target crease pattern.
- **Green** lines indicate valid but unmatched folds—useful for experimentation or divergence.
- **Red** lines represent the next expected creases in Learn Mode, helping guide the user step by step.

This color-coded feedback is updated in real time as folds are detected and evaluated. In Auto Mode, the system also displays nearby model suggestions as transparent overlays or bounding edges, enabling the user to converge toward a known model or intentionally branch away.

Temporal and Structural Considerations

Fold matching is not strictly sequential. Users may fold in a different order than the reference sequence, and the system is tolerant to reordering. However, a history of fold timestamps is maintained in QLCP to:

- Track user progress over time.
- Prioritize recent folds for feedback.

- Avoid redundant matches to the same crease.

By combining spatial matching with real-time visual cues, the system supports immediate, embodied feedback and helps learners develop an intuition for crease structure, accuracy, and alignment. Implementation details for real-time feedback rendering are provided in Section **??**.

3.6 Mode Design and Interaction Flow

To support various learning styles and use cases, the system is organized into three interaction modes: *Create*, *Learn*, and *Auto*. Each mode defines a distinct structure of interaction, feedback, and crease model behavior, while sharing a common underlying architecture for input detection, graph updating, and projection rendering.

The user can switch between modes using predefined hand gestures or by restarting the system with a different input configuration. All modes are built on the same real-time loop of gesture tracking, paper detection, crease graph modeling, and projection feedback.

3.6.1 Create Mode

Create Mode functions as a freeform crease design tool. Users can draw creases by folding the paper, and the system detects and stores these as new edges in the QLCP graph. Each added fold is projected back onto the paper surface for visualization and refinement.

Key characteristics of Create Mode include:

- No predefined target model.
- All folds are accepted and visualized.
- Users may "save" the resulting QLCP structure for future use in Learn Mode.
- Useful for prototyping, teaching, or developing new folding sequences.

3.6.2 Learn Mode

Learn Mode guides the user through folding a predefined model, either loaded from an SVG file or created previously in Create Mode. The system sequentially or structurally compares each new fold to the reference QLCP and provides immediate feedback on fold correctness.

Feedback logic includes:

- Red projection lines indicate expected creases not yet folded.
- Blue lines show correctly matched folds.
- Green lines indicate creases that do not match any target in the reference model.

This mode is intended to help learners better understand structural relationships between folds, improve accuracy, and gain confidence through immediate visual confirmation.

3.6.3 Auto Mode

Auto Mode supports exploratory folding without a fixed target. As the user folds, the system builds a live QLCP graph and compares it to a dataset of known origami structures. If a close match is found, suggestions or partial projections are shown to assist or inspire the user. If no match is found, the user is free to continue developing a new crease pattern.

Auto Mode includes:

- Dynamic matching of the live graph against a dataset.
- Optional visual projection of partial candidate structures.
- Divergence detection: alerts when the current fold diverges significantly from all known models.

- Ability to convert the final folded pattern into a new saved QLCP.

This mode supports creative discovery, remixing of known folds, and learning through improvisation.

3.6.4 Unified Interaction Flow

Despite differing objectives, all three modes share a continuous, gesture-driven interaction model. Users fold, receive feedback, and control the system through embedded pose-based gestures without interrupting the physical act of folding. This unified interaction flow reinforces the embodied nature of origami learning while allowing flexibility in learning pathways.

A detailed breakdown of how each mode is implemented and linked to the projection system is provided in Chapter 4.

3.7 Chapter Summary and Transition

This chapter presented the methodological foundation for a real-time, gesture-driven origami learning system. It introduced the system architecture and explained how each module—gesture recognition, paper detection, crease pattern modeling (QLCP), fold matching, and projection feedback—contributes to a seamless, embodied interaction loop.

The system operates in a continuous, real-time pipeline:

1. **Input Capture:** An RGB-D camera captures both color and depth data of the user's hands and the paper surface.
2. **Gesture Recognition:** Hand landmarks are extracted using MediaPipe, normalized, and matched against recorded poses to detect interaction commands.
3. **Paper Detection:** The system detects and refines the paper shape using HSV masking, contour approximation, and geometric reasoning.

4. **Fold Detection and Graph Update:** Differences in the paper's shape are analyzed to identify new folds, which are then mapped into QLCP graph space upon gesture confirmation.

5. **Matching and Feedback Logic:** Detected creases are compared against a reference model (in Learn Mode) or a database (in Auto Mode). Matching status is encoded as visual feedback.

6. **Projection Output:** Projected overlays of crease patterns, feedback cues, and suggested next steps are rendered onto the paper using homography-based transformation.

As shown earlier in Figure 3.1, these stages form a cohesive system that allows users to interact directly with the paper surface while receiving real-time guidance. The tight coupling of perception and feedback minimizes cognitive switching and supports continuous learning through embodied action.

The next chapter describes the concrete implementation of this methodology, detailing the hardware components, software integration, and data structures that enable this real-time origami learning environment.

Chapter 4

Experiment Implementation

4.1 System Setup Overview

The interactive origami system is implemented as a tabletop projection workspace with overhead sensing and feedback. A calibrated RGB-D camera and projector are mounted above a flat work surface to capture user input and display visual guidance in real time. The spatial alignment between input sensing and output projection is critical to enabling continuous, hands-on interaction with the paper.

Figure 4.1 provides a schematic view of the setup, including the camera, projector, and folding surface. Figure 4.2 shows the actual workspace used for development and testing.

The rest of this chapter details the hardware components, software integration, user interface behavior, and crease graph handling used to realize this environment.

4.2 Hardware Setup

The physical implementation of the system consists of an RGB-D camera, an overhead projector, and a worktable surface onto which guidance is projected. The setup is designed to minimize occlusion, support accurate gesture tracking, and maintain spatial consistency for projection alignment.

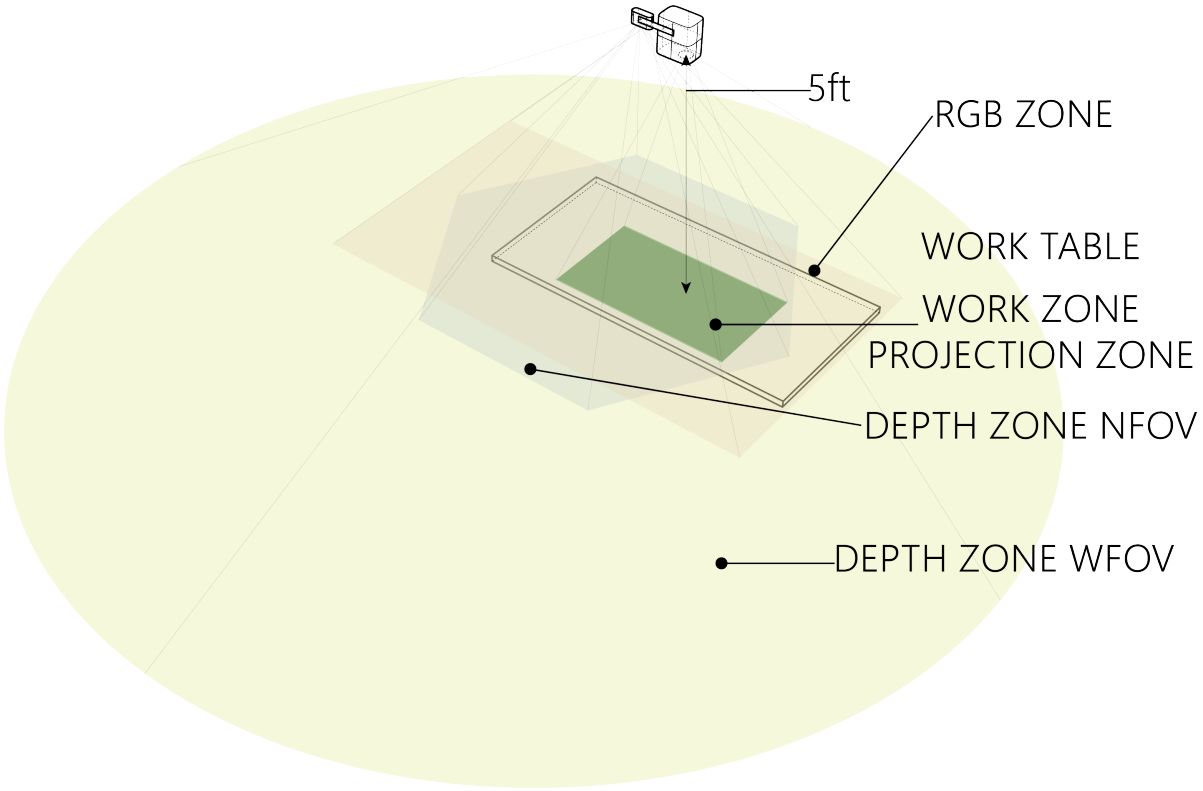


Figure 4.1: Schematic layout of the system. The RGB-D camera captures gestures and paper from above; the projector overlays real-time visual guidance onto the folding surface.

4.2.1 RGB-D Camera: Orbbec Femto Bolt

The system uses the Orbbec Femto Bolt RGB-D camera to capture both visual and depth data from the overhead view of the workspace. This camera enables the system to simultaneously track hand gestures and detect changes in the paper's shape over time.

- **Connection:** Type-C connection to the host computer.
- **Data Streams:**
  - RGB: Used for color-based paper detection and projection alignment.
  - Depth: Used for accurate hand tracking and occlusion-aware gesture recognition.
- **Placement:** Mounted directly above the folding surface to minimize distortion



Figure 4.2: Photograph of the physical workspace showing the mounted camera and projector above a table workspace.

and ensure a consistent field of view.

- **Resolution:** 1280×720 (RGB) and 640×576 (depth) at 30 FPS.
- **Software Integration:** Integrated through TouchDesigner, which streams RGB-D data into the gesture recognition and paper tracking modules.

4.2.2 Projector: Miroir L710S Pro

An overhead projector is used to display guidance directly onto the paper surface, allowing users to receive visual feedback without shifting attention to a separate screen. This real-time projection is crucial for maintaining embodied interaction.

- **Connection:** HDMI connection to the host system via TouchDesigner.
- **Resolution:** Full HD 1920×1080 for precise line and crease rendering.

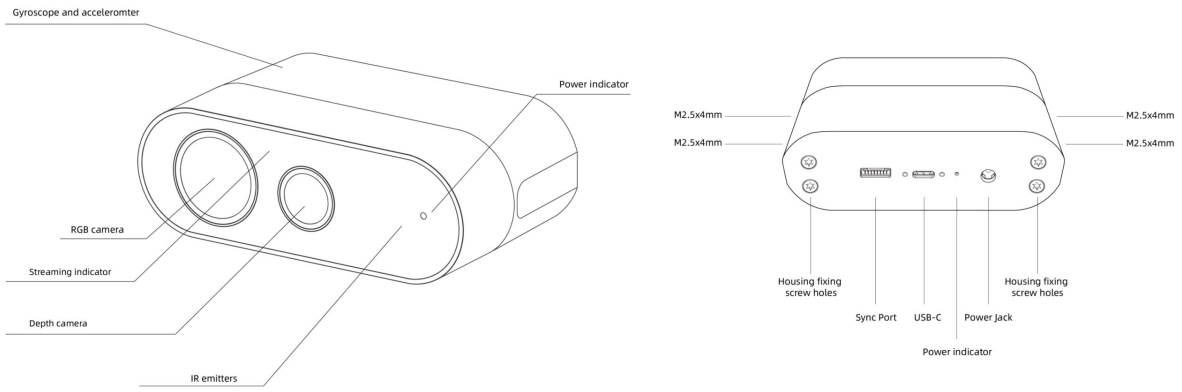


Figure 4.3: Overhead RGB-D camera (Orbbec Femto Bolt) mounted for paper and hand tracking.

- **Brightness:** 400 ANSI lumens to ensure visibility under moderate lighting.
- **Projection Size:** Adjustable from 20" to 120", calibrated to a table-sized workspace.
- **Software Integration:** Receives transformed visual overlays (red, blue, green creases) for display. Projection is dynamically updated based on graph state and fold feedback.

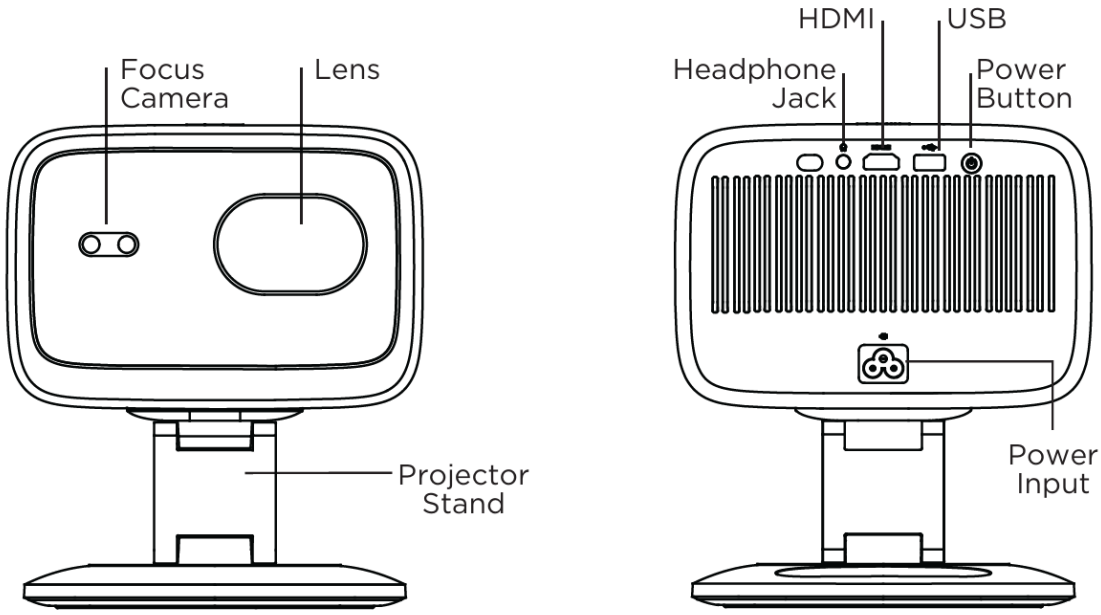


Figure 4.4: Miroir L710S Pro projector used for real-time visual feedback.

4.2.3 Workspace Setup and Calibration

To ensure stable paper detection and consistent projection alignment, the system uses a manual four-point calibration step to define the workspace boundaries. The user selects the four corners of the physical table area by clicking directly on a live camera feed. These points are saved to a file (`selected_points.txt`) and used to crop, warp, and normalize every incoming video frame.

During calibration:

- The user clicks four points $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4 \in \mathbb{R}^2$ on the table boundary in clockwise or counter-clockwise order.
- These are stored as:

$$\texttt{selected\_points.txt} = \begin{bmatrix} x_1, y_1 \\ x_2, y_2 \\ x_3, y_3 \\ x_4, y_4 \end{bmatrix}$$

- These points form the source quadrilateral $\mathbf{Q}_{\text{src}}$, and are mapped to a target square $\mathbf{Q}_{\text{dst}}$ in normalized model space using a homography:

$$H = \texttt{cv2.getPerspectiveTransform}(\mathbf{Q}_{\text{src}}, \mathbf{Q}_{\text{dst}})$$

- Each input frame I is processed as:

$$I' = \texttt{warp}(\texttt{crop}(I, \mathbf{Q}_{\text{src}}), H)$$

- A rotation and flipping step is applied to standardize orientation across setups.

This calibration pipeline ensures that the physical paper region is consistently aligned with the internal coordinate system used for QLCP modeling and projection feedback.

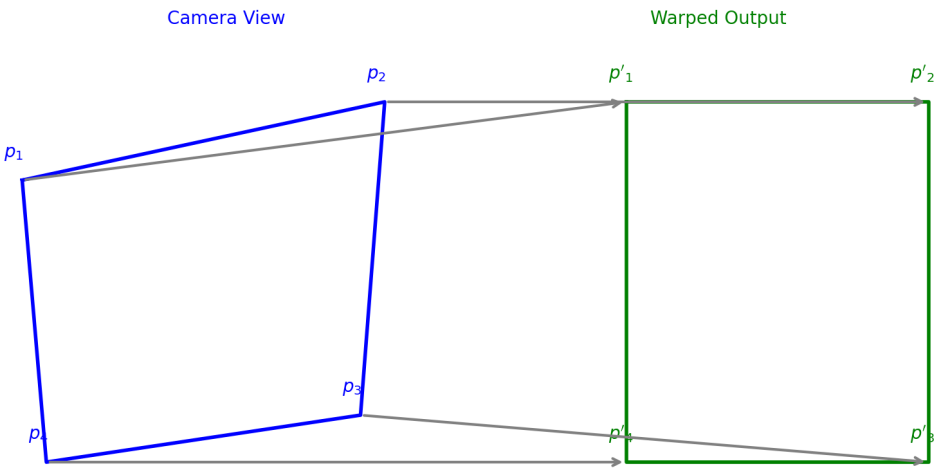


Figure 4.5: Homography transformation from selected corner points (p_1–p_4) in the camera view to a normalized square model space (p'_1–p'_4).

The calibration only needs to be performed once per setup and can be reused in subsequent sessions.

4.3 Software Implementation

4.3.1 Technology Stack and Frameworks

The system is implemented primarily in Python, using a set of real-time vision, projection, and interaction libraries. Each component of the system is modular, enabling clear separation between sensing, data processing, interaction handling, and visual feedback.

- **OpenCV** is used for image processing tasks, including color masking, contour detection, homography transformation, corner visualization, and edge drawing.
- **MediaPipe** provides efficient, real-time hand tracking. The system uses its built-in hand landmark detector to extract 21 keypoints per hand, enabling gesture matching and pose-based control.
- **pyvirtualcam** enables the system to stream processed visual output to a virtual camera device. This output can be previewed in real time or captured by external

tools.

- **TouchDesigner** is used solely for projection. The system sends visual output from Python to a virtual camera via `pyvirtualcam`, which is then captured by OBS. TouchDesigner receives the OBS feed and uses Kantan Mapper to align and project the visuals precisely onto the work zone surface.
- **NumPy and SciPy** are used for efficient vectorized calculations, including pose normalization, covariance estimation, and statistical comparison of hand gestures.
- **networkx** powers the internal crease graph structure (QLCP), allowing dynamic updates of vertices and edges based on user folds and interactions.

These tools were chosen for their performance, flexibility, and integration with real-time interactive systems. The combination supports the system's goal of delivering responsive feedback without interrupting the user's physical interaction with the paper.

4.3.2 Code Structure and Module Overview

The system is organized into modular components that separate the concerns of sensing, preprocessing, interaction logic, and visual feedback. This modular structure supports both real-time responsiveness and iterative development.

- **Frame Preprocessing:** A dedicated module handles camera calibration and frame warping. The user manually selects four corner points from the workspace, which are saved and used to crop, warp, and orient each input frame. The function `process_frame()` standardizes all frames to a top-down view of the workspace.
- **Gesture Detection and Pose Matching:** Hand landmarks are detected in each frame using MediaPipe. The system normalizes each hand's landmark positions and compares them against a stored distribution using statistical similarity metrics (e.g., Mahalanobis distance, KL divergence). If a match is confirmed over time, the corresponding system action is triggered.

- **Paper Detection and Fold Inference:** A contour-based detection pipeline extracts the paper boundary using HSV masking and corner analysis. By comparing paper shapes across frames, the system identifies fold edges and maps them into model space for graph updates.
- **Crease Graph Modeling (QLCP):** The crease structure is maintained in a directed graph. Vertices correspond to points in model space, and edges represent folds. The graph is updated dynamically as users confirm new creases. It supports operations such as overlap checking, pattern saving, and format export.
- **Projection Feedback Generation:** Based on the current graph state and system mode, visual overlays (creases, feedback colors, guidance) are generated and rendered onto the frame. These visuals are streamed to a virtual camera for projection via OBS and TouchDesigner.

All components share data through structured function calls and consistent coordinate systems, ensuring that input tracking, fold detection, and projection remain aligned.

4.3.3 Data Flow Between Components

The system operates as a continuous real-time loop that connects camera input, gesture recognition, paper tracking, graph updates, and projection rendering. Data flows through the following sequence:

1. **Camera Input:** The RGB-D camera captures color and depth frames. These frames are preprocessed using the manually calibrated region of interest.
2. **Frame Processing:** The cropped region is warped into a normalized model space using a perspective transformation derived from the selected calibration points.
3. **Gesture and Paper Detection:** In each processed frame:
    - Hand landmarks are extracted and normalized for gesture recognition.

- The paper shape is extracted using HSV thresholding and contour approximation.

4. **Crease Inference and Graph Update:** If a change in paper shape is detected and a gesture confirmation condition is met, a new crease is inferred and added to the QLCP graph.

5. **Visual Feedback Generation:** Based on the updated graph and system mode, feedback overlays are drawn (e.g., red = expected crease, blue = matched, green = unmatched).

6. **Projection Output:** The visual output is streamed to a virtual camera and projected back onto the paper surface, closing the feedback loop.

This pipeline is synchronized to run in real time at approximately 30 frames per second. Key processing steps are modularized to maintain consistent timing and minimize latency between gesture detection and visual feedback.

4.3.4 TouchDesigner Integration

TouchDesigner is used exclusively for projecting the final system output onto the physical workspace. Rather than handling any internal processing or sensing, it functions as a visual pipeline that maps virtual camera output from the system onto the real-world paper surface using spatial alignment tools.

The integration is structured as follows:

- **Virtual Camera Output:** Python uses the `pyvirtualcam` library to stream the final annotated frame (with projected creases, feedback lines, and interface elements) to a virtual webcam device.

- **OBS Routing:** Open Broadcaster Software (OBS) captures the virtual webcam feed and forwards it as a live video source. This allows the stream to be accessible by any third-party application, including TouchDesigner.

- **Projection Mapping with Kantan Mapper:** In TouchDesigner, the OBS feed is received and passed into a Kantan Mapper projection setup. Kantan allows manual mapping of the video output to the paper zone on the table, accounting for perspective distortion, skew, and projector alignment.
- **Dynamic Update:** As the system runs, the frame-by-frame visuals (e.g., updated creases, gesture highlights) are continuously refreshed and reprojected with low latency.

This setup decouples visual rendering from projection calibration, enabling high frame-rate feedback while preserving flexibility for physical workspace changes.

4.3.5 Real-Time Performance Considerations

The system is designed to operate in real time, maintaining responsiveness for gesture recognition, paper tracking, and feedback projection. The following strategies are employed to ensure consistent performance:

- **Efficient Frame Processing:** All image operations (e.g., masking, contour extraction, corner detection) are implemented using optimized OpenCV functions. Gaussian blurring and color thresholding are kept minimal to reduce latency.
- **Modular Matching Pipeline:** Gesture recognition is implemented using lightweight per-frame matching and buffered pose comparison. Rather than relying on deep learning inference per frame, normalized landmark distributions are matched using statistical metrics, enabling consistent evaluation at 30 FPS.
- **One-Time Inference for Paper Calibration:** Paper calibration is performed once at startup and reused. This prevents redundant homography recalculation and speeds up frame warping.
- **Visual Output Decoupling:** Projection rendering is separated from core data processing. The final frame is passed to a virtual camera, ensuring that any lag introduced by projection software does not interfere with sensing or graph updates.

- **Gesture Smoothing and Thresholding:** To prevent flickering or false detection, gesture matches are confirmed only if a similarity score stays below threshold in at least 80% of frames within a one-second window. This balances robustness with responsiveness.
- **Target Frame Rate:** The system maintains an average performance of 28–30 frames per second in practical usage, depending on ambient lighting, fold complexity, and gesture activity.

These performance optimizations allow the user to interact naturally with the paper and receive immediate visual cues, preserving a fluid and embodied learning experience.

4.4 User Interface and Projection System

4.4.1 Visual Encoding and Feedback Logic

The system provides visual feedback to guide the user through the folding process by overlaying crease information directly onto the paper surface. Each crease is drawn with a color corresponding to its current status within the system:

- **Red:** An expected crease from a reference model or dataset that has not yet been folded.
- **Blue:** A user-created fold that correctly matches an expected crease in position and orientation.
- **Green:** A new fold detected that does not match any of the expected creases.

These encodings are determined by comparing the edges in the user-generated QLCP graph to those in the reference pattern. A similarity match (e.g., within a threshold of position and angle) results in a color reassignment from red to blue. Unmatched folds are marked green to indicate divergence or novelty.

The function `project_qlcp_edges()` is responsible for generating the set of crease lines to be drawn, mapping the QLCP graph from model space to screen space using a homography transformation. Each edge is classified as:

- `original only` (red),
- `user only` (green), or
- `both` (blue).

This classification is based on set membership of crease edges (represented as undirected vertex pairs) between the original and user-generated graph states.

The lines are rendered frame-by-frame using OpenCV drawing functions in a black-on-white projection image. Colors are chosen for high contrast and perceptual clarity to ensure feedback is visible even under moderate lighting conditions.

4.4.2 Projection Rendering

All visual feedback is rendered directly onto the physical paper surface via an overhead projector. To ensure that crease lines align precisely with the actual paper geometry, the system uses a perspective transformation to map coordinates from a normalized model space to the current camera view.

The projection pipeline proceeds as follows:

1. The system defines a base model space as a square region, typically $[0, 0]$ to $[L, L]$, where L is the scaled edge length of the paper.
2. The function `project_qlcp_edges()` computes a homography matrix H from this model space to the current paper shape as detected in the camera frame. This is done using:

$$H = \texttt{cv2.getPerspectiveTransform}(\text{base\_quad}, \text{target\_quad})$$

3. All vertex positions in the QLCP crease graph are transformed using:

$$\mathbf{x}' = H \cdot \mathbf{x}$$

where $\mathbf{x}$ is a 2D vertex in model space and $\mathbf{x}'$ is its projected location.

4. Edges are drawn using OpenCV's `cv2.line()` function, and assigned colors based on their classification (see previous subsection).

5. The final rendered frame is streamed via `pyvirtualcam` to OBS and then projected using TouchDesigner with Kantan Mapper for alignment.

This rendering pipeline ensures that feedback remains spatially consistent with the user's folding activity. Even as the paper deforms or partially occludes, the system maintains accurate alignment through continuous recalibration of corner detection and homography.

4.4.3 Gesture-Driven Interface Control

To support hands-on learning without reliance on external input devices, the system allows users to trigger actions through recognized hand gestures. Specific hand poses are matched using MediaPipe landmark tracking and a distribution-based pose matching system described in Section 3.3.1.

The gesture interface supports the following commands:

- **Add Crease:** During setup, the user performs their preferred two-hand pose three times. The system records and models each hand's 21 landmark positions over time, constructing a statistical distribution for each landmark using mean and covariance. During runtime, the system compares incoming poses against these learned distributions using statistical distance metrics (e.g., Mahalanobis, KL divergence). A pose match is confirmed when current observations fall below learned

similarity thresholds across a majority of frames in a one-second window. This approach enables flexible, user-defined gesture control without requiring fixed spatial positions.

- **Undo / Clear:** Distinct one-handed poses (e.g., open palm at the upper corner) are associated with utility actions such as undoing the last crease or clearing all folds.

The system tracks gesture similarity over a short time window to reduce false positives. A pose match is confirmed only if the observed landmark distributions maintain sufficient similarity to the user's learned pose profile in at least 80% of frames within a one-second window. The similarity threshold is adaptive, derived from the statistical characteristics of the user's recorded poses.

During this waiting period, a countdown message is projected onto the workspace (e.g., "Adding Edge in: 2s"), helping the user anticipate system actions without diverting attention. This gesture-based interface avoids reliance on external controls, supporting a continuous and embodied folding experience.

4.4.4 Mode Awareness and Visual State

The user interface dynamically adapts based on the active system mode. Each mode presents a different visual logic and interaction style, enabling a range of learning workflows from freeform creation to guided folding and automatic pattern discovery.

- **Create Mode:** The user freely adds folds to a blank crease graph. Each added crease is projected in white, and the graph evolves in real time. This mode acts as a digital crease sketchpad, enabling exploratory design and saving of custom patterns.
- **Learn Mode:** A reference model is loaded from a saved graph or SVG-converted dataset. The system projects the expected creases in red. When the user performs a

matching fold, the projected line turns blue. Non-matching folds appear in green. This real-time visual encoding helps learners understand correctness, alignment, and fold sequencing.

- **Auto Mode:** The system starts with an empty graph. As the user folds, detected creases are compared to a dataset of existing models. When similarity passes a threshold, the system projects red candidate folds from the nearest model. The user may fold toward or diverge from the suggestion, and the system continuously updates the projection based on observed structure.

All three modes maintain consistent spatial feedback and visual structure, allowing users to fluidly move between freeform creation, guided learning, and adaptive discovery. Transitions between modes are triggered via gesture-based interface control (see previous subsection).

4.5 QLCP Graph and Dataset Handling

The system represents all creases as a dynamic graph structure called QLCP (Qingluan Crease Pattern). Each fold is defined as an edge between two vertices in 2D model space. The graph is updated in real time based on detected folds and user confirmation gestures, and is used to drive both feedback projection and model comparison.

Graph Representation

Vertices are defined as (x, y) coordinates in model space, typically normalized to a square region scaled by the paper length. When a new fold is detected, its endpoints are mapped from the image frame into model space using a homography transformation. Each endpoint is either matched to an existing vertex or added as a new vertex if no close match is found.

Each crease is stored as an undirected edge:

$$e = (v_1, v_2) \quad \text{where} \quad v_1, v_2 \in \mathbb{R}^2$$

The graph is implemented using the `networkx` library, which supports fast operations such as:

- Vertex deduplication based on spatial proximity
- Graph traversal and pattern comparison
- Serialization to `.graphml` or `.json` formats

Graph Saving and Loading

The user can save the current QLCP graph to a file using an auto-incremented filename (e.g., `crease_pattern_002.graphml`). This allows both recovery of in-progress folds and conversion to reference models for Learn Mode. Graphs can be loaded during runtime and scaled dynamically to match the current paper size.

Dataset Matching for Auto Mode

In Auto Mode, the system compares the current QLCP graph to a dataset of known crease patterns. Each comparison uses:

- Normalized edge vector directions
- Relative edge positions
- Overlapping subgraph similarity

If a match exceeds a similarity threshold, the corresponding model is loaded and projected partially around the user's folds. This suggests potential next steps or allows

deviation from known patterns. The system supports gradual convergence as the user continues folding.

Reference Pattern Integration

Graphs created in Create Mode or imported from SVG-converted designs are treated identically in Learn and Auto Modes. Internally, all reference patterns are stored as QLCP graphs, making them searchable, scalable, and suitable for feedback overlay.

4.6 User Journey and Interaction Flow

This section presents a walkthrough of how users interact with the system across its three operational modes: Create, Learn, and Auto. Each mode supports a distinct interaction flow, ranging from freeform crease design to structured learning and dataset-driven inference. Mode selection is performed at the beginning of the session via a projected start menu.

4.6.1 User Journey Overview

The user's experience begins at a projected menu interface controlled through hand gestures. From this starting point, the user selects one of three operational modes:

- **Create Mode** enables freeform construction of crease patterns. Each fold made by the user is tracked and visualized as part of a growing graph structure, allowing spontaneous design and documentation of novel folding paths.
- **Learn Mode** presents predefined origami models. The system projects expected folds step-by-step, providing adaptive feedback: correctly aligned folds are highlighted in blue, while misaligned ones appear in green. This mode guides users through structured learning experiences.

- **Auto Mode** allows open-ended exploration. As the user folds, the system dynamically compares the evolving crease graph to a dataset of known patterns. When a match is found, suggestions for possible next steps are projected, offering non-linear guidance without enforcing a fixed sequence.

Regardless of the selected mode, all workflows converge into a continuous real-time feedback loop between the *projected interface* and *gesture-based hand interaction*. This loop forms the basis of an embodied and visually responsive learning system.

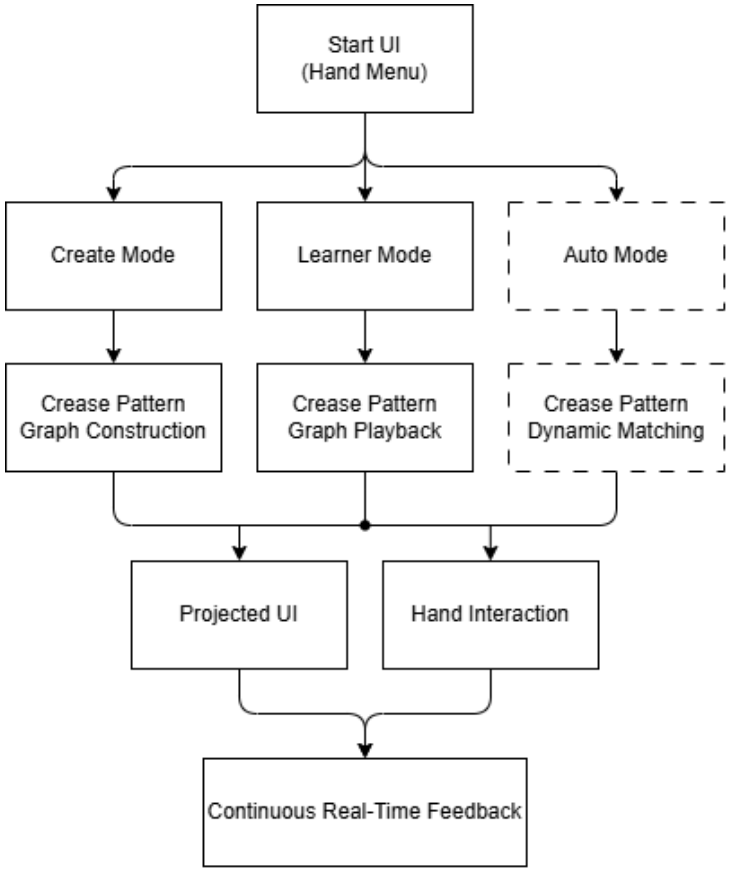


Figure 4.6: Diagram of the system's interaction flow across the three modes, converging into continuous feedback through projection and hand input.

4.6.2 Start Menu Interface

At the beginning of the interaction, users are presented with a projected *Start Menu Interface*. The interface displays three labeled blocks: *Create Mode*, *Learner Mode*, and

Auto Mode, each corresponding to a distinct system workflow.

Hand-Hover + Gesture-Based Selection. Users select a mode by hovering their hand over a block to highlight it, and then performing a closed-hand gesture to confirm the selection. This contactless approach creates an embodied interaction space, keeping the user's attention focused on the workspace surface.

Palm positions are tracked using MediaPipe's hand landmark detector, which provides 21 3D landmarks per hand. A hand is classified as "closed" when three or more fingers are curled inward. This is determined by comparing the y-coordinates of each fingertip and its corresponding DIP joint:

$$\text{Closed fingers count} = \sum_{i=1}^{4} \left[y_{\text{tip}_i} > y_{\text{dip}_i} \right]$$

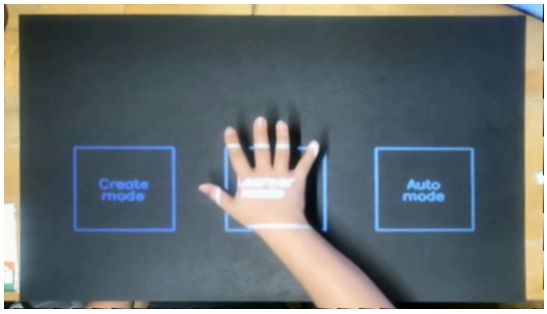
$$\text{Hand is closed if count} \geq 3$$

Once the hand is identified and tracked, its center is compared against the rectangular bounds of each menu block. The selection is triggered when the following condition is met:

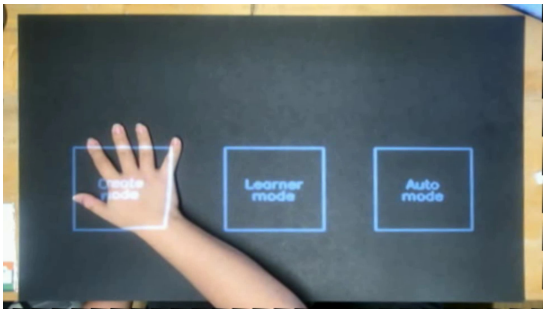
```
if hand_closed and palm_center is inside block_rect then
```
→ **trigger mode selection**

Learner Mode Model Selection. If *Learner Mode* is selected, the interface transitions to a scrollable grid of model blocks. The user can scroll vertically through the list by dragging a closed hand up or down, and select a model using a quick "tap" gesture (open → closed → open) over the desired block.

This projected interface eliminates the need for physical controls and supports a smooth, intuitive entry point into the system's three modes, enhancing the embodied and uninterrupted flow of interaction.



(a) User hovering over "Learner mode" block.



(b) User hovering over "Create mode" block.

Figure 4.7: Mode selection interface using hand-hover and gesture-based interaction.

4.6.3 Interaction Modes

The system supports three distinct interaction modes—*Create*, *Learner*, and *Auto*—each designed to accommodate different approaches to origami folding. These modes reflect a spectrum of engagement, from open-ended exploration to structured guidance and intelligent assistance.

This multimodal structure allows the system to support a wide range of users, including beginners seeking step-by-step instruction, hobbyists creating their own crease patterns, and curious learners experimenting with unknown outcomes. By adapting to different levels of structure and feedback, the system lowers the barrier to entry and encourages an embodied, hands-on approach to learning.

The shared feedback loop across all modes ensures continuity in interaction: folds are tracked, projected, and interpreted in real time, enabling the user to stay focused on the paper without switching between devices or tools.

Learner Mode. This mode guides users through a step-by-step origami folding process by projecting crease patterns from a saved model file. It supports real-time fold tracking and immediate visual feedback to facilitate learning and error correction.

The system first detects the paper and hands through the overhead camera using computer vision techniques. Once the paper's corner points are identified, the loaded

crease pattern is aligned and scaled to the current paper position. Fold lines are projected directly onto the surface.

As the user folds, the system compares detected creases against the expected pattern and updates the display with color-coded feedback:

- **Red:** Expected fold (yet to be made)
- **Green:** New, unexpected fold (user deviation)
- **Blue:** Correctly matched fold

To confirm a fold, the user performs a personalized hand gesture previously learned by the system. This gesture triggers the addition of the fold to the internal graph structure, prompting the system to project the next step.

Purpose: Recreate saved crease patterns through projected guidance and physical folding.

Key Features:

- Load previously saved models (`.graphml` or `.json`)
- Dynamically align and scale patterns to the detected paper
- Provide continuous visual feedback and correction

User Interaction:

- Perform hand gestures to confirm folds
- Navigate the entire learning sequence without external devices

Goal: Support hobbyist learners by making the folding process clear, adaptive, and responsive to individual variations.

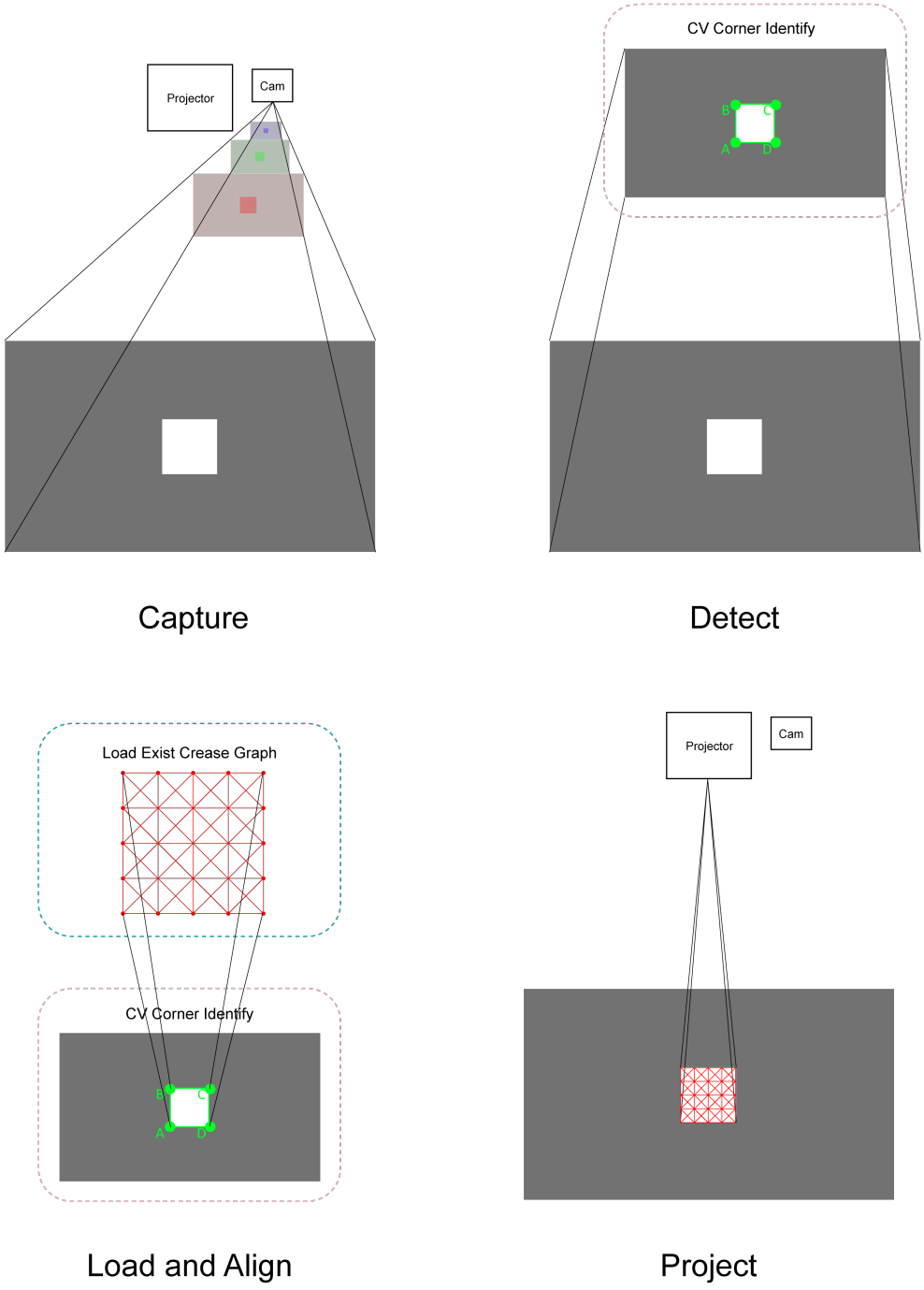


Figure 4.8: Diagram of the Learner Mode workflow. The system continuously senses paper shape and user hand positions, projecting expected creases, and updating feedback in real time.

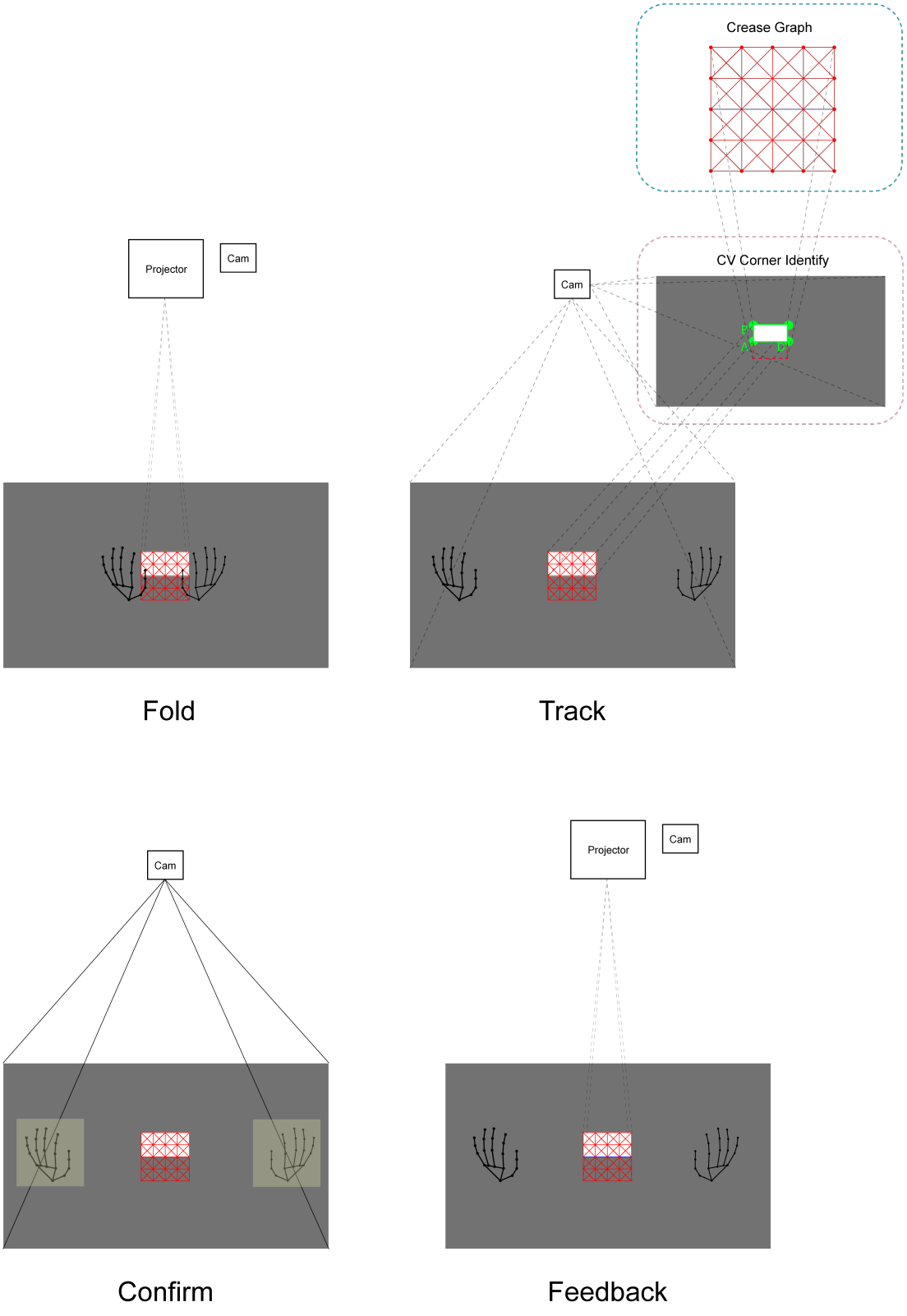


Figure 4.8: Continuation of the figure.

Create Mode. In this mode, users can freely design custom crease patterns by physically folding paper and confirming each fold through gesture interaction. The system starts by detecting the paper and initializing an empty crease graph aligned with the detected sheet. As folds are made, the system tracks the paper geometry and hand activity, allowing each crease to be added to the graph via a personalized confirmation gesture.

Projected feedback overlays each added fold directly onto the paper, enabling users to visually construct intricate patterns in real time without disrupting their tactile workflow.

Purpose: Enable users to design and document unique crease patterns interactively.

Key Features:

- Starts with a blank crease graph over a detected paper sheet
- New folds are added by performing a gesture after folding
- The system maps each new crease into the internal graph structure
- Immediate visual feedback: creases are projected onto the physical paper
- Generated crease graphs can be saved for later learning or remixing

User Interaction:

- Perform a gesture to confirm and record each new fold
- No traditional interface — the workflow is fully gesture- and fold-driven

Goal: Support creative exploration and design by allowing users to construct their own folding sequences with intuitive, embodied control.

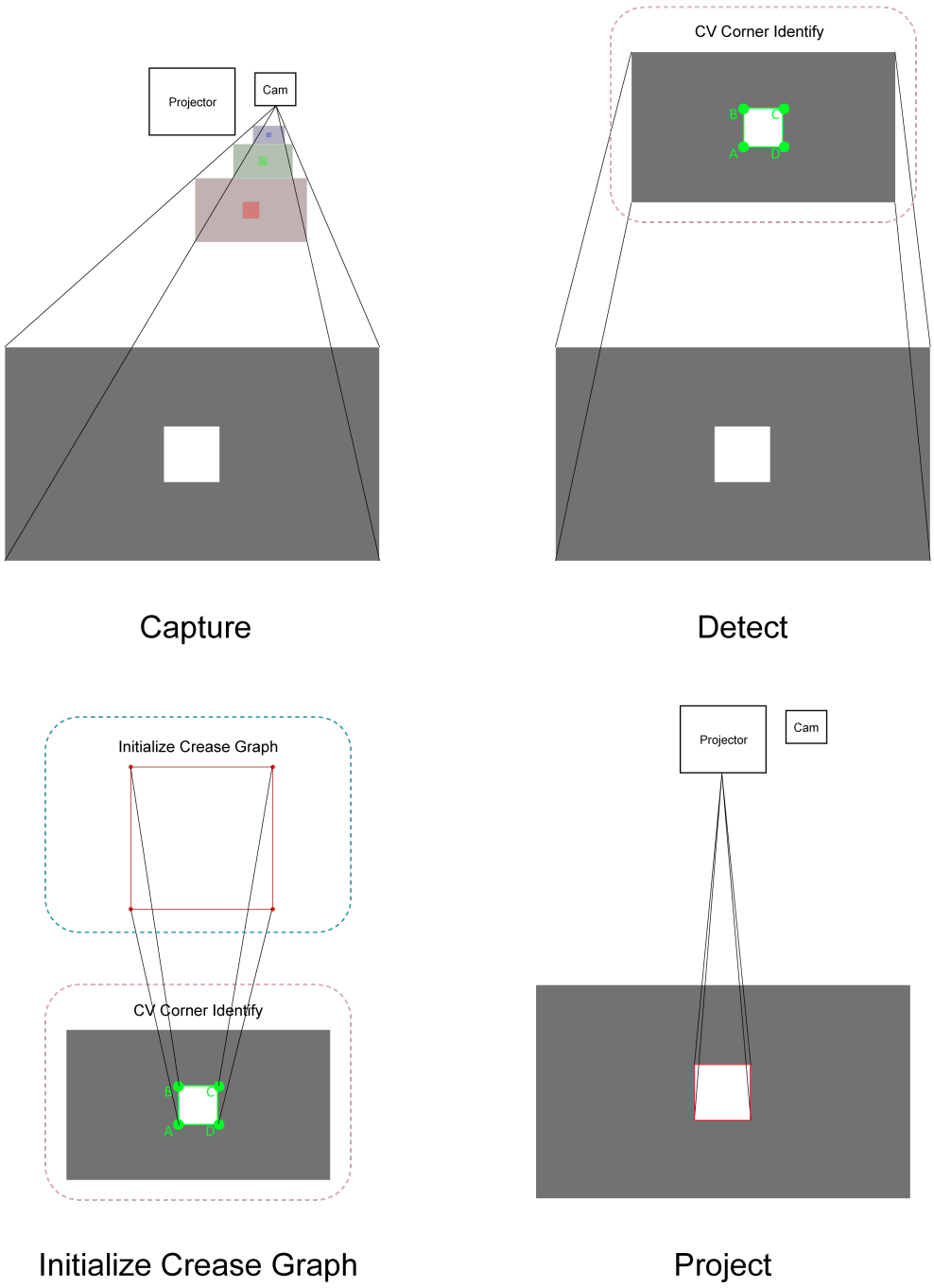


Figure 4.9: Create Mode workflow. The system tracks folds and hand movements, adding new creases to a graph structure and projecting them in real time for visual feedback.

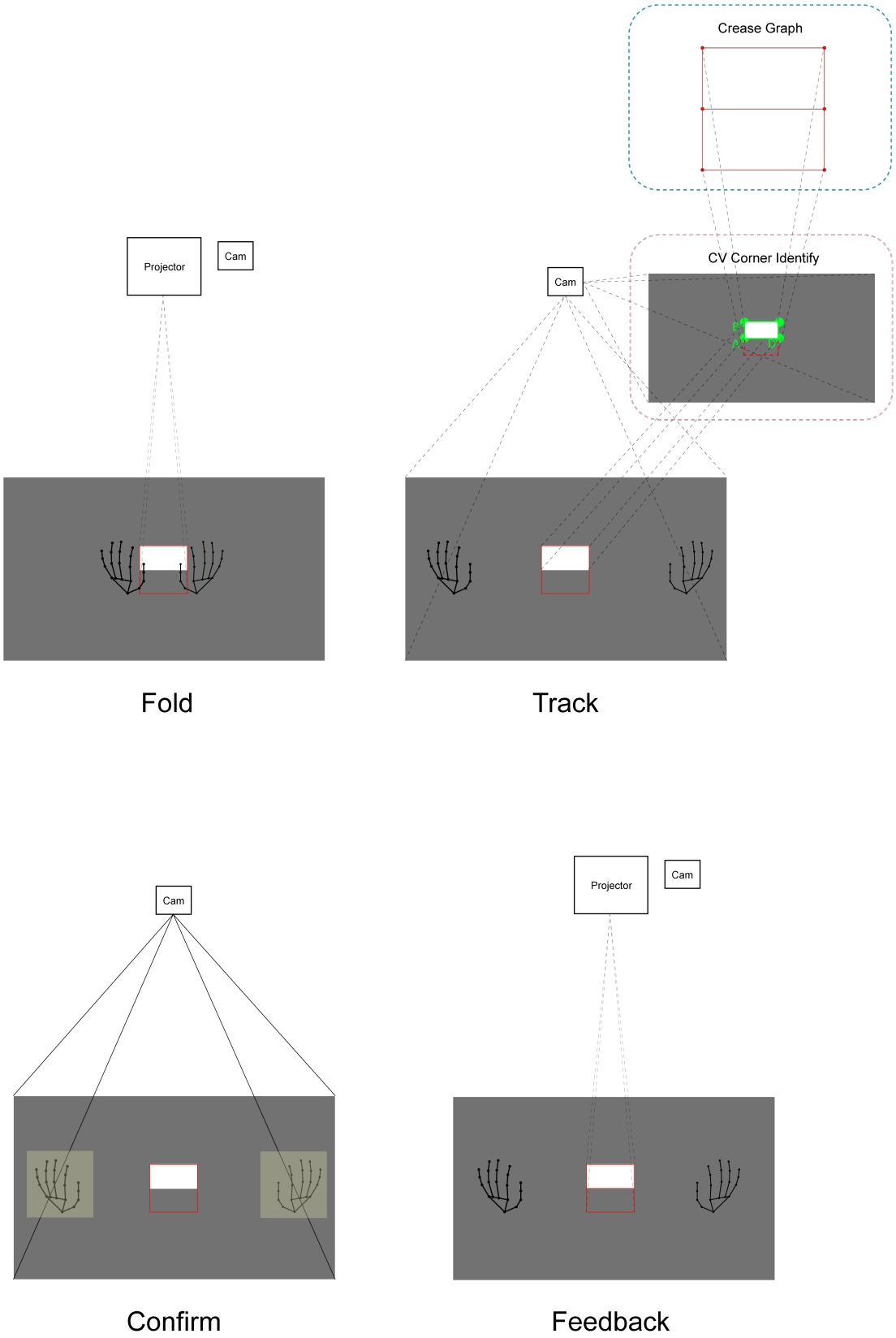


Figure 4.9: Continuation of the figure.

Auto Mode. Auto Mode supports a discovery-driven approach to origami learning by enabling users to fold freely while the system continuously tracks the evolving crease pattern. Unlike Learner Mode, which projects a predefined sequence, this mode offers guidance only when patterns begin to emerge — transforming origami into an improvisational and exploratory experience.

As the user folds, the system builds a real-time crease graph and compares it against a dataset of known patterns. When partial or full matches are detected, the system projects suggested next creases directly onto the paper. These suggestions are intended to assist rather than constrain, giving users the option to follow or diverge from known models.

Purpose: Enable open-ended folding exploration without predefined targets.

Conceptual Workflow:

- Begin with a blank crease graph
- Track new creases as they are added during folding
- Continuously compare the live graph to a dataset of known patterns
- Project suggestions for next folds when partial matches are found

Goal: Encourage playful and improvisational learning by shifting from rigid instruction to flexible, real-time discovery. This mode supports experimentation, remixing, and gradual understanding of complex fold logic.

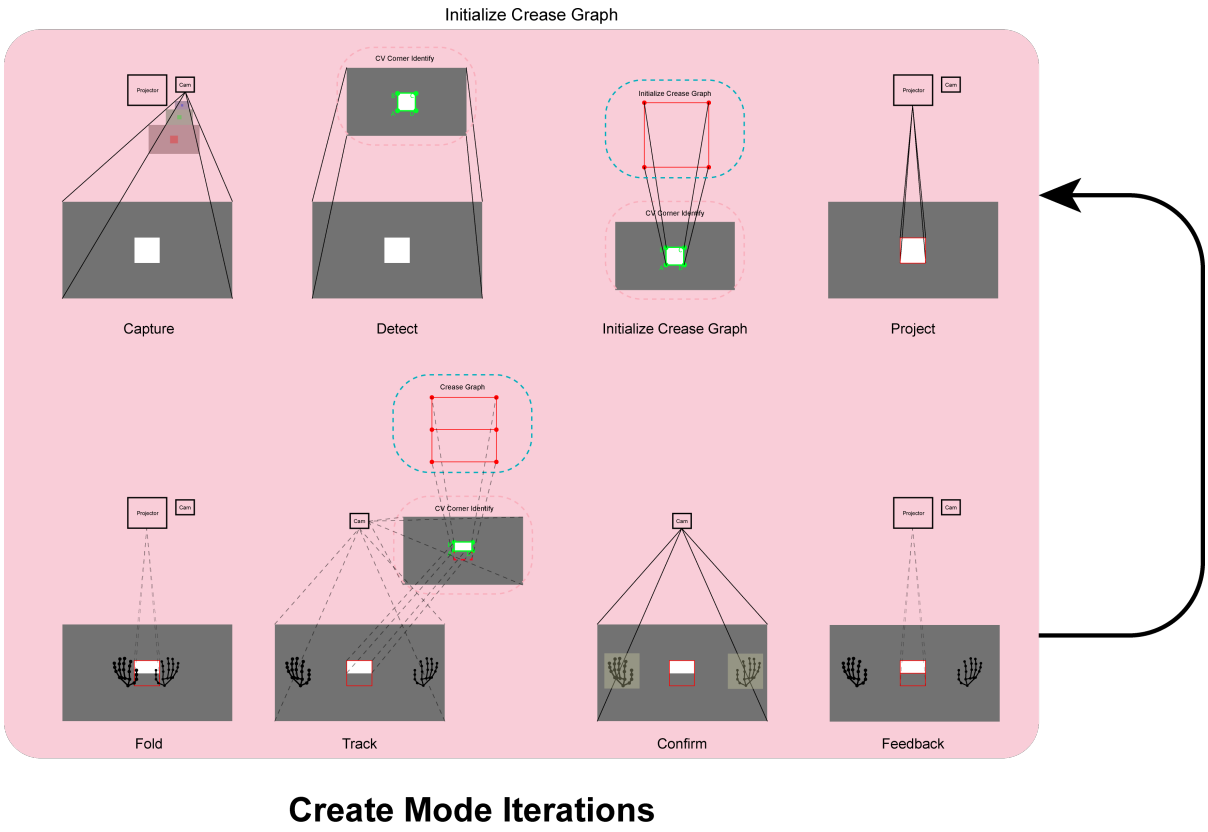


Figure 4.10: Conceptual flow of Auto Mode. As users fold, the system dynamically tracks and compares emerging crease graphs to known patterns and suggests future fold paths.

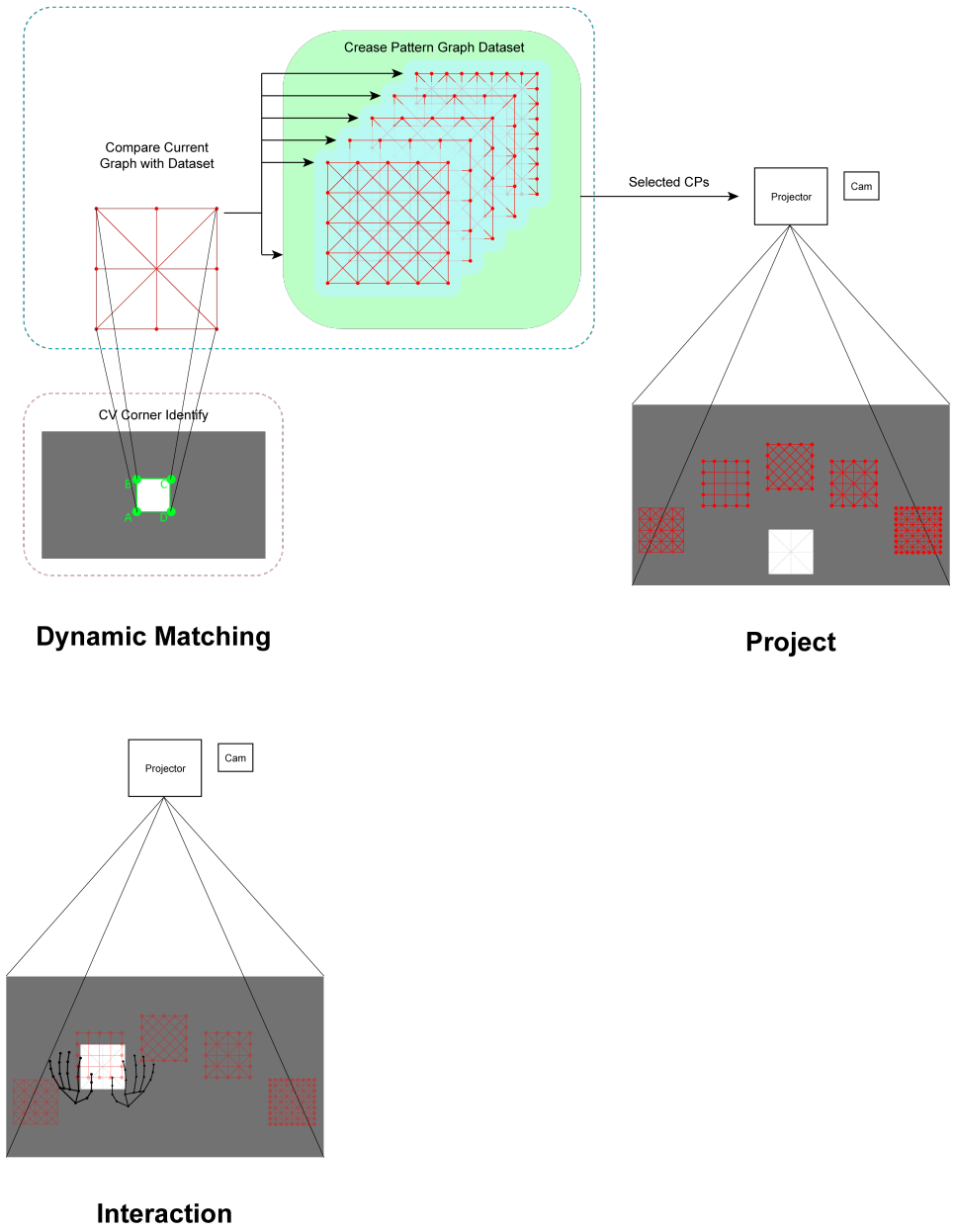


Figure 4.10: Continuation of the figure.

4.7 Chapter Summary

This chapter detailed the implementation of a real-time origami learning system that combines RGB-D sensing, graph-based crease modeling, and gesture-driven interaction. Each mode—Create, Learn, and Auto—offers a different balance of freedom and guidance, supported by projection-based feedback aligned with user actions. While the system performs well under typical conditions, it remains sensitive to occlusion, calibration drift, and hardware setup constraints. These limitations highlight opportunities for future improvements, including more robust gesture recognition, adaptive projection calibration, and support for fluid mode transitions. The next chapter presents the results and user observations collected during system testing.

Chapter 5

Results and Discussion

5.1 Experiment Overview

To demonstrate and evaluate the system's technical performance and interaction flow, a Miura-ori pattern was folded using both *Create Mode* and *Learner Mode*. This evaluation was conducted directly by the system designer, focusing on responsiveness, gesture interaction, and feedback accuracy rather than on user experience or learning outcomes.

In the first phase, the Miura-ori pattern was constructed from scratch on a 6-inch square sheet using Create Mode. Each fold was confirmed via gesture input, and the system tracked and projected the crease pattern in real time. In the second phase, the saved crease graph was reloaded and followed on a 12-inch paper using Learner Mode, where projected creases and fold feedback guided the same sequence.

Both sessions were recorded using the system's overhead RGB-D camera during live operation. Screenshots from the recordings were manually selected to represent key moments in each folding sequence. For each mode, 16 representative frames were chosen to illustrate the progression of folding, interaction, and system response over time.

5.2 Create Mode Results

In Create Mode, the user started with a blank crease graph and constructed a Miura-ori pattern through free folding. After each fold, a learned gesture was used to confirm and add the crease to the graph, which was then projected back onto the paper.

The system successfully tracked hand and paper position, projected updated crease

graphs in real time, and allowed spontaneous construction without predefined steps. The figure below shows 16 key frames from the recording:

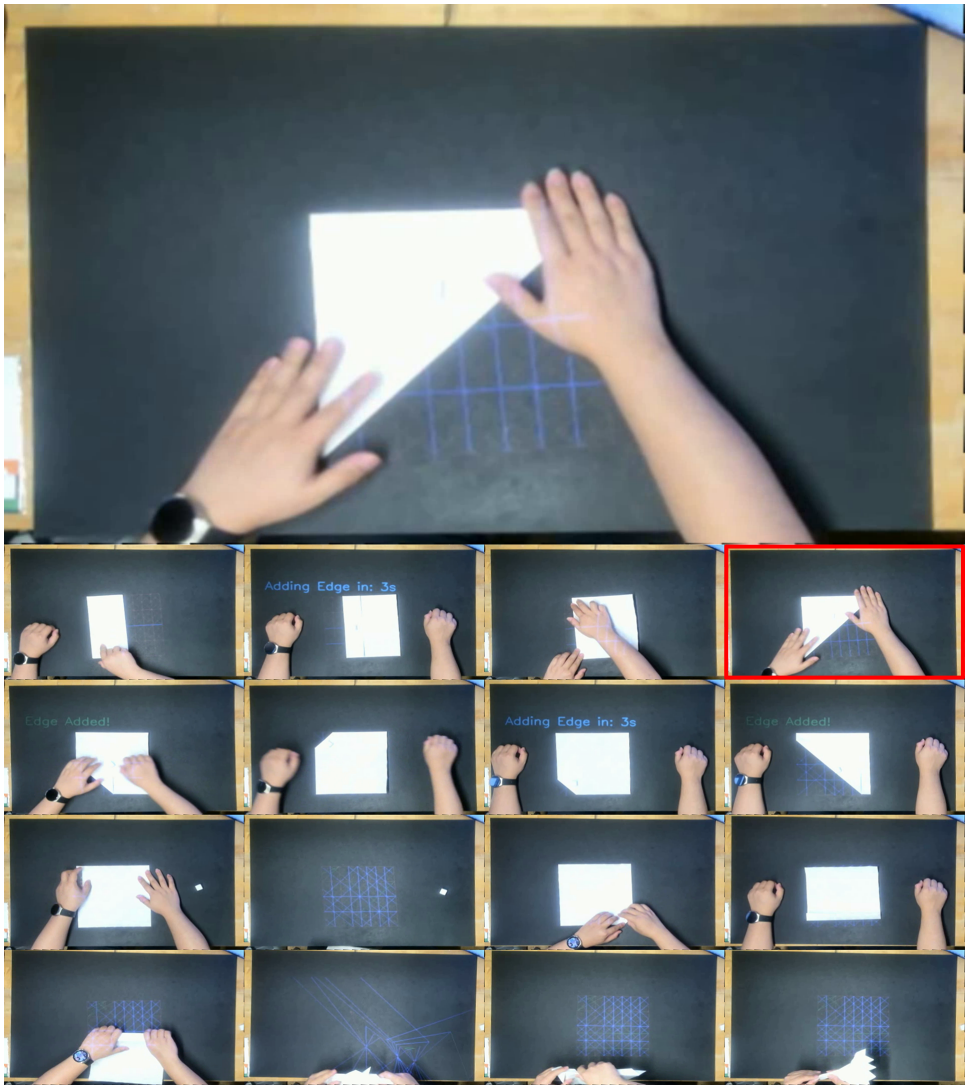


Figure 5.1: Create Mode folding sequence. Sixteen manually selected frames from the real-time session showing the progression of freeform folding, gesture-based crease confirmation, and live crease graph projection.

5.3 Learner Mode Results

In Learner Mode, the previously saved crease graph was loaded and scaled to fit a larger paper. The system projected expected folds in red, detected user-made creases in real time, and updated feedback with blue (matched) and green (unexpected) highlights. Fold

confirmation was again performed via gesture input.

The system maintained visual alignment, interpreted fold progress accurately, and responded to hand gestures across the full folding sequence. However, a few folds that were physically well-aligned with the expected creases were still marked incorrectly—most often appearing green instead of blue. This likely resulted from slight deviations in fold angle or detection order, revealing the system's sensitivity to small geometric inconsistencies. Sixteen representative frames illustrating this guided interaction are shown in Figure 5.2.

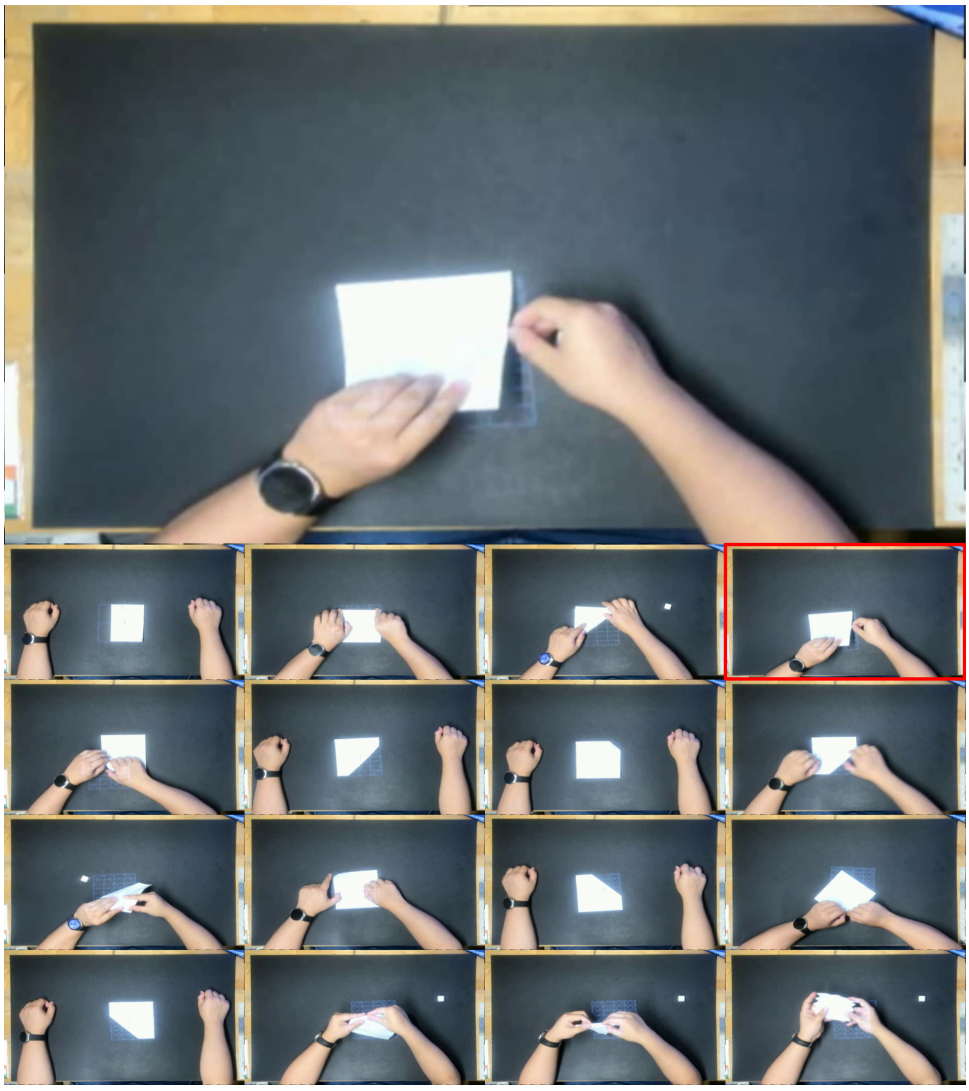


Figure 5.2: Learner Mode folding sequence. Sixteen representative frames from the recorded session demonstrate guided folding using projected instructions, real-time feedback, and gesture-driven step progression.

5.4 Discussion and Reflections

The Create and Learner Mode experiments highlight two distinct forms of interaction enabled by the system: open-ended construction and structured guidance. Each mode supports a different learning experience and presents different challenges for real-time interaction.

In Create Mode, the system successfully supported freeform folding. Gesture-based crease confirmation felt fluid and responsive, allowing the user to stay engaged with the paper without switching context. However, accuracy depended on stable lighting and consistent paper detection, especially as folds accumulated and paper shape deviated from the initial square. Projection feedback was immediate, but could drift slightly when the paper shifted between gestures, requiring occasional user adjustment.

Learner Mode offered a more guided experience. The system's ability to align a saved crease pattern to a new paper position and provide real-time feedback on folding accuracy made it effective for step-by-step learning. The color-coded overlay (red, green, blue) helped the user recognize mistakes and corrections without needing verbal or textual instructions. Gesture recognition remained reliable during most of the sequence, though occasional hand occlusion—especially when folds crossed the paper midline—led to momentary detection gaps.

Comparing the two modes revealed important differences in user focus and interaction rhythm. Create Mode encouraged spontaneity and exploration, giving the user more agency over structure. Learner Mode, by contrast, required more frequent attention to projected feedback and fold alignment, supporting a more instructional mindset. In both cases, the system enabled continuous, embodied interaction and reduced reliance on traditional interface devices.

These tests suggest that the system performs effectively in real-time origami contexts, with design tradeoffs emerging between openness and structure, gesture control and visual feedback, and detection accuracy versus interaction flexibility.

One notable challenge that appeared in both modes was fold resolution and alignment. Although the system enforces a minimum fold resolution—such as requiring creases to span at least one-eighth of the paper—some folds were still detected with inaccurate start or end points. This issue became more pronounced when the paper was slightly sheared or tilted, leading to visual drift or incorrect graph updates. These detection inaccuracies occasionally affected crease classification and alignment, particularly in Learner Mode where feedback relies on precise matching.

5.5 Limitations

While the system performed well in both Create and Learner Modes, several limitations emerged during implementation and experimentation.

One recurring challenge was hand occlusion. When the user's hands overlapped with key areas of the paper—especially during folds near the center or along shared creases—paper detection became less stable. This sometimes affected corner tracking or caused temporary inaccuracies in the crease graph. A potential solution would involve masking hand regions or applying inpainting to recover missing paper data.

Projection alignment was another sensitivity point. Minor shifts in paper position, changes in ambient lighting, or slight paper curvature affected the accuracy of projected crease lines. Over longer sequences, drift will occasionally appear, requiring manual repositioning or recalibration.

Gesture recognition, while generally robust, occasionally misfired during transitional movements or under inconsistent lighting. Since the current system uses static hand pose classification, it can miss gestures expressed in motion or misinterpret finger curl variation.

Auto Mode was not included in the test sequence due to its dependency on a larger pattern dataset, which is currently limited. As a result, dynamic fold matching and inference-based feedback remain underdeveloped in the current system version.

Lastly, mode switching is only available at system startup. Users cannot shift between Create and Learner Modes mid-session, which limits the ability to iteratively switch between design and learning workflows within the same folding process.

These limitations point to key areas for refinement—particularly in sensing robustness, projection calibration, and dataset scaling—to expand the system's versatility and reliability in future iterations.

5.6 Chapter Summary

This chapter presented the results of folding a Miura-ori pattern using both Create and Learner Modes of the system. Through these experiments, the system demonstrated its ability to support freeform crease design, projected step-by-step instruction, real-time gesture interaction, and visual feedback.

The recorded sessions showed how the system tracks and responds to folding actions, providing both flexibility and guidance across different modes of engagement. While several limitations emerged—such as sensitivity to paper alignment, gesture misclassification, and fold resolution—these insights reveal areas for future improvement and validate the core premise of using projection and gesture-based interaction for embodied origami learning.

These experiments offer preliminary evidence and reflections in relation to the research questions posed in Chapter 1:

- **RQ1:** The system successfully demonstrated adaptive visual feedback and gesture-based fold confirmation, indicating that it can support more accessible and error-tolerant learning interactions.
- **RQ2:** While no formal user study was conducted, observations from system use suggest that real-time feedback helps users maintain folding flow and focus, potentially enhancing engagement.
- **RQ3:** Although Auto Mode was not fully implemented or tested, the current

system architecture lays the foundation for recognizing and predicting complex patterns through dynamic graph matching.

The next chapter summarizes the overall contributions of this work and outlines directions for continued development.

Chapter 6

Conclusion and Future Work

6.1 Summary of Contributions

This thesis presented an interactive origami learning system that integrates real-time sensing, gesture-based control, and projection-based feedback. The system supports three interaction modes—Create, Learner, and Auto—each designed to accommodate different learning workflows: from open-ended exploration to structured step-by-step instruction.

Technically, the system combines sensing, computer vision-based paper and crease tracking, hand gesture recognition, and dynamic graph construction to model the folding process. Visually, it projects immediate feedback onto the workspace, allowing users to engage physically with paper while receiving live guidance. The Create Mode enables users to build crease patterns from scratch, Learner Mode guides them through saved models with feedback cues, and Auto Mode lays the groundwork for pattern-based inference through dataset matching.

The Create–Learn experiment with a Miura-ori pattern served as a system-level validation, showing that users could construct and follow complex patterns with minimal disruption, using only their hands and paper as input and surface.

Together, these contributions provide partial evidence in support of the thesis's original research questions—particularly in demonstrating real-time responsiveness (RQ1), the impact of embodied feedback on folding progression (RQ2), and the feasibility of structure-guided inference (RQ3).

6.2 Key Insights and Reflections

The system demonstrates that origami learning can be meaningfully enhanced through embodied interaction. Projected guidance helps clarify complex folding steps, while gesture control reduces the need for external input devices and preserves continuity of focus. The combination of visual feedback and physical manipulation supports a more intuitive and exploratory folding experience.

The three-mode structure proved valuable for accommodating different levels of expertise and learning goals. Create Mode encourages self-directed design; Learner Mode supports scaffolded practice and correction; Auto Mode opens up the potential for guided improvisation.

However, the experiments also revealed limitations, particularly around gesture robustness, fold resolution accuracy, and projection alignment. These challenges highlight the need for more resilient detection techniques and smoother transitions between user intention and system recognition. These insights reinforce the premise that interactive, adaptive systems can enhance origami learning by addressing both instructional clarity and creative flexibility—central themes reflected in the original research questions.

6.3 Future Work

System Improvements

- **Color-Aware Paper:** Using dual-color or patterned paper can help detect fold directions and assist in recognizing complex assembly stages. Preliminary tests with white-red paper already showed improved crease-side classification.
- **Hand Occlusion Handling:** Incorporating hand masking and background inpainting would reduce errors in paper and fold detection when the user's hand overlaps key regions of the sheet.

- **Lighting Adaptation:** Projection and vision modules could be improved by integrating dynamic calibration methods to adapt to changing light conditions and surface reflectivity.
- **Gesture Recognition Enhancement:** Expanding from static gesture classification to temporal modeling (e.g., time-series analysis) may reduce false positives and better capture gestures performed in motion.
- **Mid-Session Mode Switching:** Supporting transitions between Create and Learner Modes during a session would allow users to shift seamlessly between exploration and instruction without restarting the system.

Research Opportunities

- **Learning Evaluation:** A formal user study could assess the system's impact on learning outcomes, comparing it to conventional origami tutorials in terms of clarity, retention, and engagement.
- **Collaborative Interaction:** Supporting two-person interactions (e.g., teacher–student or peer–peer) would enable studies of how shared feedback, gestures, and physical proximity affect the folding process and social learning.
- **Generative Guidance:** With a larger dataset, the system could suggest new folds based on the user's current graph state—transforming it into a co-creative agent for origami discovery.
- **Multi-User and Remote Use:** Exploring synchronous interaction between users in shared or remote spaces may open pathways for collaborative or distributed origami learning environments.

6.4 Final Thoughts

This thesis began with a central premise: that origami, as a spatial and embodied craft, offers a unique context for rethinking how learning systems engage the body, material, and machine together. By integrating real-time computer vision, gesture-driven input, and projection-based feedback into a single, continuous interaction loop, the system developed here offers a proof of concept for how digital technologies can support tactile, uninterrupted learning experiences.

The three-mode architecture—Create, Learner, and Auto—was designed to support a spectrum of user intentions and skill levels, from open-ended creation to guided instruction and exploratory pattern recognition. This multimodal design, grounded in real-time sensing and spatial feedback, illustrates how embodied interaction can be made both technically feasible and pedagogically meaningful.

The results presented in Chapter 5 suggest that the system can clarify complex folds, reduce cognitive switching, and allow learners to maintain focus on the physical surface. While not a formal user study, the folding sessions—captured in Create and Learner Modes—demonstrated how the system supports interaction fluency, adaptability, and visual-verbal alignment of instruction and action. These findings provide partial but encouraging responses to the research questions posed at the outset, particularly around visual guidance (RQ1), continuous feedback (RQ2), and the potential for structure inference (RQ3).

From a methodological standpoint, the project contributes a novel synthesis of gesture recognition, crease modeling, and dynamic projection. The QLCP graph structure supports incremental fold tracking, while the pose-based gesture learning system allows for personalized control that avoids physical input devices. This architecture not only reinforces embodied learning, but also sets the stage for further exploration in multimodal sensing and interaction design.

In response to reviewer feedback and early experimentation, several promising directions have emerged. A dual-color paper system—tested using red and white sides—enabled

directional fold detection and could be expanded to support automatic crease orientation or assist in assembly stages. Patterned paper may offer even greater potential for 3D reconstruction through visual features, though this remains a technical challenge for future work. Other suggestions, such as integrating subtle audio cues to reinforce visual feedback, or supporting collaborative folding—whether co-located or remote—highlight opportunities to expand the system's scope into social and multisensory domains. While these extensions were not fully implemented, they reflect the system's modular potential and the researcher's awareness of broader application contexts.

This work also enters a broader conversation in HCI and design research about material engagement, gesture, and feedback. It echoes recent interest in systems that teach by doing, where the line between instruction and creation becomes increasingly blurred. The project responds to calls for more tactile, participatory technologies that do not ask users to translate their bodily activity into abstracted clicks or typed commands but instead meet them in their own mode of interaction.

While this prototype is limited in scope and robustness, its design points to new possibilities for physical-digital systems that foster exploration, embodiment, and learning. By treating the folding process itself as a dialog between system and user, it opens the door for future research in collaborative making, adaptive instruction, and the aesthetics of interface design. In that sense, origami becomes not only a medium of folding—but of unfolding new modes of interaction.

Glossary

crease recognition A computer vision technique used to identify and classify the location, type, and geometry of creases on a sheet of paper, typically in the context of origami or folding processes. It enables the system to track folding progress, detect errors, and guide the user by analyzing visual patterns such as lines, edges, or shadows on the paper surface..

marker-based A tracking system that relies on physical markers—such as QR codes, fiducial tags, or reflective objects—placed on users or tools to enable computer vision systems to detect and interpret position, orientation, or interaction events with high precision.

markerless A tracking approach that uses computer vision or depth sensing to directly recognize body parts, gestures, or objects without requiring any physical markers or tags, allowing for more natural and flexible interaction.

MediaPipe Hands A real-time hand tracking framework developed by Google that detects 21 anatomical landmarks per hand from a single RGB camera, supporting left/right hand detection, handedness classification, and 2D/3D output suitable for gesture-based interfaces.

QLCP Short for *Qingluan Crease Pattern*, a dynamic graph-based data structure used to represent origami crease patterns. It encodes each crease as an edge and each paper vertex as a node, supporting real-time updates as the user folds. The QLCP graph enables the system to track fold sequences, validate fold alignment, and compare user-generated structures to predefined models or a dataset of known origami patterns..

vision-based step matching A method that uses computer vision to observe and an-

alyze a sequence of physical actions or transformations — such as folds in origami — and compares them against a predefined or learned model of correct steps. It enables real-time feedback by detecting discrepancies between performed and expected steps, often using techniques like pose estimation, shape matching, or temporal sequence alignment..

Bibliography

Antle, A. N., Corness, G., & Droumeva, M. (2009). What the body knows: Exploring the benefits of embodied instruction in early childhood. *Interacting with Computers*, *21*(1–2), 66–75. https://doi.org/10.1016/j.intcom.2008.10.005

Beck, C. (2021). Easy eye-hand coordination activities with paper [Accessed March 2025]. https://www.theottoolbox.com/easy-eye-hand-coordination-activities-with-paper/

Blikstein, P. (2013). Digital fabrication and 'making' in education: The democratization of invention [Available at: https://fablearn.org]. *FabLearn Conference Proceedings*.

Boakes, N. J. (2008). Origami instruction in the middle school mathematics classroom: Its impact on spatial visualization and geometry knowledge of students. *Mathitudes*, *1*(1), 1–8. https://www.fau.edu/education/centersandprograms/mathitudes/documents/20080901bmathitudesoct08revisionfinalversionforpublicationoct242008.pdf

Bullivant, L. (2006). *Responsive environments: Architecture, art and design*. V&A Publications.

Dourish, P. (2001). *Where the action is: The foundations of embodied interaction*. MIT Press.

Fan, L., Mareschal, D., & Xu, F. (2022). Origami performance as a measure of spatial ability in children and adults. *Frontiers in Psychology*, *13*, 825462. https://doi.org/10.3389/fpsyg.2022.825462

Ghassaei, A. (2016). Origami simulator [Accessed March 2025]. https://origami.amandaghassaei.com/

González, E., & García-Murillo, F. J. (2024). Mathematics teachers' professional experimentation with mathematical origami in secondary education. *European Journal of Mathematics and Science Education*, *20*(1), 45–61. https://www.ejmste.com/

download/mathematics-teachers-professional-experimentation-with-mathematical-origami-in-secondary-education-15911.pdf

Grandi, C. (2024). Origami in lessons [Accessed March 2025]. https://www.artfulmaths.com/origami-in-lessons.html

Hull, T. (2006). *Project origami: Activities for exploring mathematics*. A K Peters/CRC Press.

Ishii, H., & Ullmer, B. (1997). Tangible bits: Towards seamless interfaces between people, bits and atoms. *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 234–241.

Jun Mitani. (2014). Oripa: Origami pattern editor [Available at https://mitani.cs.tsukuba.ac.jp/oripa/].

Kapur, M., & Bielaczyc, K. (2012). Designing for productive failure. *The Journal of the Learning Sciences*, *21*(1), 45–83. https://doi.org/10.1080/10508406.2011.591717

Lang, R. J. (2012). *Origami design secrets: Mathematical methods for an ancient art*. CRC Press.

Lee, J. C., & Hudson, S. E. (2004). Lightguide: Projected visualizations for handheld tools. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 473–480. https://doi.org/10.1145/985692.985748

Łysakowski, M., Gapsa, J., Lyu, C., Bohné, T., Tadeja, S. K., & Skrzypczyński, P. (2025). Enhancing augmented reality with machine learning for hands-on origami training. *Frontiers in Virtual Reality*, *6*. https://doi.org/10.3389/frvir.2025.1499830

Melancon, D., Gorissen, B., Garcia-Mora, L., Reyssat, E., Bertoldi, K., & Studart, A. R. (2021). Mechanical metamaterials for soft robotics. *Nature Reviews Materials*, *6*, 485–503.

Migliorini, L., & Andreani, F. (2023). Origami and descriptive geometry: Enhancing geometric thinking through foldable models. *Nexus Network Journal*, *25*, 585–603. https://doi.org/10.1007/s00004-023-00694-4

Mikhak, B., Lyon, C., & Gorton, T. (2002). The tower system: A toolkit for prototyping tangible user interfaces [Available via ResearchGate. Accessed March 2025]. *MIT Media Lab Technical Report*. https://www.researchgate.net/publication/

228560423_The_Tower_system_A_toolkit_for_prototyping_tangible_user_interfaces

Miura, K., Kawasaki, T., Tachi, T., Uehara, R., Lang, R. J., & Wang-Iverson, P. (Eds.). (2015). *Origami⁶: Proceedings of the sixth international meeting on origami in science, mathematics and education* (Vol. 2) [Part 2: Origami in Technology, Science, Art, Design, History, and Education]. American Mathematical Society.

Na, J. H., Evans, A. A., Bae, J. S., Chiappelli, M. C., Santangelo, C. D., Lang, R. J., Hull, T. C., & Hayward, R. C. (2015). Programmable origami with energy storing composite materials. *Advanced Materials*, *27*(1), 79–85.

Overvelde, J. T., Weaver, J. C., Hoberman, C., & Bertoldi, K. (2016). A three-dimensional actuated origami-inspired transformable metamaterial. *Nature Communications*, *7*, 10929.

Patil, R. S., & Kshirsagar, V. K. (2024). Effectiveness of origami in teaching mathematical concepts at primary level. *International Journal of Novel Research and Development*, *9*(7), 248–252. https://www.ijnrd.org/papers/IJNRD2407289.pdf

Pixle. (2025). Foldify [Mobile application for creating printable paper models. Accessed March 2025]. https://www.foldifyapp.com/

Rahmawati, Y., Setiawati, I., & Suryana, A. (2022). The effect of origami activities on the fine motor development of children aged 4-5 years. *Proceedings of the 4th International Conference on Early Childhood Education*. https://www.atlantis-press.com/article/125988069.pdf

Rus, D., & Tolley, M. T. (2015). Design, fabrication and control of soft robots. *Nature*, *521*(7553), 467–475.

Schmidt, R. A., & Lee, T. D. (2018). *Motor learning and performance: From principles to application* (6th). Human Kinetics.

Sheu, F.-R., Fang, W., & Chen, N.-S. (2013). How gesture-based technology is used in education to support teaching and learning: A content analysis. *Proceedings of the 21st International Conference on Computers in Education, ICCE 2013*, 463–468. https://doi.org/10.58459/icce.2013.498

Sigrist, R., Rauter, G., Riener, R., & Wolf, P. (2013). Augmented visual, auditory, haptic, and multimodal feedback in motor learning: A review. *Psychonomic Bulletin & Review*, *20*(1), 21–53.

Silverberg, J. L., Evans, A. A., McLeod, L., Hayward, R. C., Hull, T., Santangelo, C. D., & Cohen, I. (2014). Using origami design principles to fold reprogrammable mechanical metamaterials. *Science*, *345*(6197), 647–650.

Tachi, T. (2009). Simulation of rigid origami. *Origami 4: Proceedings of the 4th International Meeting of Origami in Science, Mathematics, and Education*, 175–187.

Tachibana, T., Kojima, S., Tanaka, T., & Kawaguchi, N. (2022). Origamiar: A projection-based origami instruction system with real-time step recognition. *Proceedings of the 2022 ACM International Conference on Interactive Surfaces and Spaces (ISS)*, 219–221. https://doi.org/10.1145/3532106.3563540

Tibbits, S. (2014). Active matter. In *Architectural design* (pp. 50–55, Vol. 84). Wiley.

Vavoula, G., & Sharples, M. (2009). Meeting the challenges in evaluating mobile learning: A 3-level evaluation framework. *International Journal of Mobile and Blended Learning*, *1*(2), 54–75. https://doi.org/10.4018/jmbl.2009040104

Weichel, C., Lau, M., Wigdor, D., Alexander, J., & Gellersen, H. (2014). Mixfab: A mixed-reality environment for personal fabrication. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 3855–3864. https://doi.org/10.1145/2556288.2557090

Wilson, M. (2002). Six views of embodied cognition. *Psychonomic Bulletin & Review*, *9*(4), 625–636.

Yamamoto, A., Waki, M., Akaguma, R., & Terada, T. (2022). Interactive origami learning system using projection-based ar and depth camera. *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*, 1–6. https://doi.org/10.1145/3491101.3519695

Yang, Y., Chen, K., White, T. J., & Bunning, T. J. (2015). Origami-inspired robotics: A decade of progress. *Soft Robotics*, *2*(4), 155–164.