Full-surround 3D Reconstruction using Kaleidoscopes

Submitted in partial fulfillment of the requirements for
the degree of

Doctor of Philosophy
in
Electrical and Computer Engineering

Byeongjoo Ahn

B.S., Electrical and Computer Engineering, Seoul National University
M.S., Electrical Engineering and Computer Science, Seoul National University

Carnegie Mellon University
Pittsburgh, Pennsylvania

December 2023

Acknowledgments

I would like to extend my deepest gratitude to my PhD advisors, Aswin C. Sankaranarayanan and Ioannis Gkioulekas. Every interaction with them, enriched by their intelligent and insightful discussions, has consistently felt like a privilege, offering me a valuable glimpse into the ideal form of scholarly engagement I aim to pursue throughout my academic and professional life. Their mentorship has been exemplary, characterized by an enduring curiosity, a passion for learning, openness to new ideas, and a dedicated commitment to students, day and night. This has set a high standard for me and provided a solid foundation for my academic journey. My heartfelt thanks also go to my committee members, Shree K. Nayar and Manmohan Chandraker, for their valuable insights and significant contributions that have profoundly impacted my research. Their inspiring works, particularly about interreflections, have not only been mirrored in my research but also illuminated it with deeper understanding.

To my labmates at Image Science Lab – Harry Hui, Chia-Yin Tsai, Jian Wang, Rick Chang, Vishwa Saragadam, Yi Hua, Anqi Yang, Michael De Zeeuw, Wei-Yu Chen, Leron Julian, Tyler Nuanes, Natalie Janosik, Yingsi Qin, Haejoon Lee, Sagnik Ghosh, Kuldeep Kulkarni, Vijay Rengarajan, Denis Guo, Shigeki Nakamura, Bhargav Ghanekar, Yash Belhe, Tejas Gokhale, Aparajith Srinivasan, Ethan Tseng, Yongyi Zhao, Angela Gao, Manu Gopakumar, Jeremy Klotz, Carlos Taveras – thank you for making my PhD life memorable and enriching. The time spent and discussions we shared together in the lab, surrounded by the hardware and experiments, will always be cherished and unforgettable. Your friendship and support have been invaluable.

I am also grateful to the members of the CMU Imaging Group and my collaborators at Rice University – Srinivasa Narasimhan, Matthew O'Toole, Ashok Veeraraghavan, Adithya Pediredla, Mark Sheinin, Akshat Dave, Alankar Kotwal, Shumian Xin, Bailey Miller, Bakari Hassan, Dorian Chan, Benjamin Attal, Arjun Teh – for their invaluable discussions and insights that greatly enriched my work.

To my roommates and my Pittsburgh family – Juyong Kim, Jinmo Rhee, Hun Namkung, Byungsoo Jeon, Kiwan Maeng, Soyong Shin, Jay-Yoon Lee, Daehyeok Kim, Jisu Kim, Kijung Shin, Giljoo Nam – your support and companionship have been a source of comfort and joy during this journey. I am particularly grateful to Chris Asenjo, who has become like family in Pittsburgh, enriching my experience with enjoyable discussions about American culture. A heartfelt thank you to my colleagues from KIST – Ig-Jae Kim, Junghyun Cho, Heeseung Choi, Gi Pyo Nam, Haesol Park – for their wisdom and guidance, and belief in my abilities.

My dissertation and PhD studies were made possible through the support of the KFAS Scholarship, DARPA REVEAL (grant numbers HR0011-16-C-0025, HR0011-16-C-0028), the National Science Foundation (grant numbers 1652569, 1730574, 1730574, 1900849, 2008464), a gift from AWS Cloud Credits for Re- search, as well as a Sloan Research Fellowship for Ioannis Gkioulekas. I am deeply thankful for this generous support, which has significantly contributed to my ability to concentrate on my research.

Last but certainly not least, my deepest appreciation is reserved for my family. To my parents, parents-in-law, and my brother, for their unwavering love and encouragement. I dedicate this thesis to my family, particularly to my wife, Yeonsoo Jung, for her endless patience, understanding, and love. Your support has been the foundation of my strength and success.

Abstract

3D scanning of a single view of an object seldom suffices. Be it for 3D printing, augmented reality, or virtual reality, scanning of the shape of the entire object in all its complexity—what we refer to as full-surround 3D—is critical to have a faithful digital twin.

A key factor in achieving full-surround 3D scan is the diversity and number of viewpoints. Standard techniques, involving multiple cameras and sometimes projectors, often become prohibitively expensive and complex, placing them beyond the reach of the average consumer of 3D technology. This thesis proposes a novel and accessible solution by using a kaleidoscope system. Comprising multiple planar mirrors, the kaleidoscope allows for a combinatorial increase in viewpoints through repeated light reflections, all captured by a single camera. This approach enables the construction of a virtual multi-view imaging system that is easy to build, calibrate and deploy, with components that are easily available.

In this thesis, we establish the theoretical and practical foundations of kaleidoscopic techniques for full-surround 3D reconstruction. We are particularly interested in the reconstruction of highly complex objects with intricate geometry that include self-occlusions. This work aims to create a kaleidoscopic equivalent to the multiple view geometry in classical computer vision, making the following contributions.

First, we explore kaleidoscope design and its calibration. It encompasses the development of crucial design factors, including the number of mirrors and their configuration. Additionally, we formulate metrics for assessing various kaleidoscope designs, enabling the quantitative evaluation of their coverage and accuracy. We also address the precise calibration of the extrinsic parameters of mirrors, a vital aspect of kaleidoscopic reconstruction. This precision is essential, as errors can be significantly amplified through the multiple reflections within the kaleidoscope.

Second, we introduce kaleidoscopic structured light, which serves as a kaleidoscopic equivalent to traditional structured light. The kaleidoscope generates the structured light system with hundreds of projectors and cameras, providing high accuracy and coverage, thanks to redistribution of the rays diverse directions corresponding to large baselines. It enables the reconstruction of intricate geometry, that include severe self-occlusions, concavities, and large genus number.

Third, we present kaleidoscopic neural rendering, an adaptation of neural rendering within a kaleidoscopic context. While neural rendering has recently achieved significant advancements in speed and accuracy, its handling of reflections remains inadequate. Our framework addresses this by integrating multiple specular reflections within the kaleidoscope alongside neural surface representations. This integration enables the creation of silhouette-consistent and photo-consistent 3D shapes from a single

kaleidoscopic image. This method facilitates single-shot full-surround 3D reconstruction, particularly advantageous for dynamic objects, as a single frame in this setup contains all necessary information for complete surround coverage.

For my family

Contents

List of Figures

List of Tables

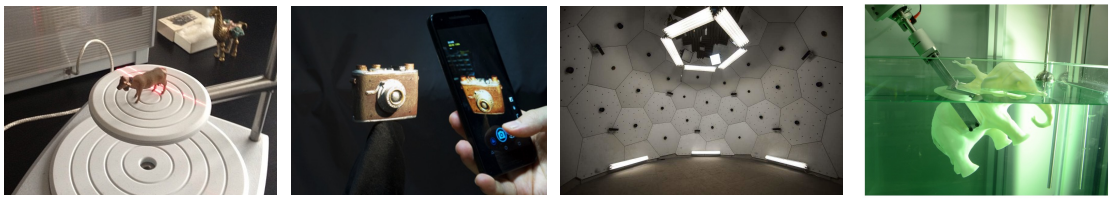
1 Introduction

3D scanning of a single view of an object seldom suffices. Be it for 3D printing, augmented reality, or virtual reality, scanning of the shape of the entire object in all its complexity—what we refer to as *full-surround 3D*—is critical to have a faithful digital twin.

A key factor in achieving a full-surround 3D scan is the number of viewpoints from which an object is imaged. Covering the entire object that we seek to scan typically requires a large number of diverse viewpoints. Furthermore, this number increases with the complexity of the object. This requirement has led to the construction of light stages with multiple cameras, and potentially projectors, that capture digital content at high fidelity. Unfortunately, the cost and complexity of these systems place them beyond the reach of the average consumer of 3D technology.

1.1 Problem: Full-surround 3D reconstruction

Common approaches for reconstructing a complete scan of an object include rotating the object [Kang *et al.*, 2019; NextEngine, 2000; Park *et al.*, 2016; Xia *et al.*, 2016; Zhou *et al.*, 2013], moving a camera around the object [Cui *et al.*, 2010; Holroyd *et al.*, 2010; Kolev *et al.*, 2014; Lichy *et al.*, 2021; Nam *et al.*, 2018; Newcombe *et al.*, 2011; Ondrúška *et al.*, 2015; Wu *et al.*, 2015; Wu and Zhou, 2015], or constructing a multi-camera system [Ghosh *et al.*, 2011; Joo *et al.*, 2017; Schwartz *et al.*, 2013]. However, rotating the object around a fixed axis (e.g., using a turntable) constrains viewpoints to lie on a plane perpendicular to the rotation axis; such viewpoints may be insufficient for the full-surround reconstruction of intricate objects. Moving a single camera requires estimating the camera pose, whereas multi-camera systems are generally costly and difficult to build. A fascinating approach for full-surround 3D reconstruction is the so-called *dip transform*, where the object is dipped into a fluid at multiple orientations [Aberman *et al.*, 2017]. Measuring the liquid displacement, which encodes the submerged volume, provides sufficient information to recover the shape of the object. Although this method provides superior results for



(a) Turntable
(-) Limited viewpoints

(b) Moving a camera
(-) Registration

(c) Multi-camera
(-) Hard to build

(d) Dip transform
(-) Requires immersing

Figure 1.1. **Non-kaleidoscopic full-surround 3D reconstruction approaches.** (a) Rotating an object constrains viewpoints to lie on a plane perpendicular to the rotation axis. (b) Moving a single camera requires the registration for estimating the camera pose. (c) Multi-camera systems are generally costly and difficult to build (d) Dip transform [Aberman *et al.*, 2017]—a technique that dips the object into a fluid at multiple orientations and reconstructs the shape from liquid displacements—requires immersing the object in the fluid.

intricate objects, it requires immersing the object in the fluid, which is not always feasible. Figure 1.1 summarizes these surround reconstruction approaches and their limitations.

1.2 Kaleidoscope

A simple way to achieve a very large number of viewpoints is to surround the object we want to scan with mirrors, which conveniently provide additional viewpoints without the need to move a camera or construct a multiple-camera system. In particular, a kaleidoscope [Brewster, 1858], which consists of multiple planar mirrors, allows light to bounce around repeatedly until it hits the camera, thereby providing a combinatorial increase in the number of viewpoints. Thus, with a kaleidoscope and a single camera, we can construct a *virtual* multi-view imaging system that is easy to build, calibrate and deploy, with components that are easily available.

This thesis aims to solve the full-surround 3D reconstruction problem using a kaleidoscope. This approach has the following advantages over the above conventional 3D reconstruction approaches.

- **Combinatorial increase in the number of viewpoints.** A key factor in achieving a full-surround 3D scan is the number of viewpoints from which an object is imaged. A kaleidoscope allows light to bounce around repeatedly until it hits the camera, thereby providing a combinatorial increase in the number of viewpoints. Thus, we achieve a very large number of viewpoints without constructing multi-camera systems that are generally costly and difficult to build.

Table 1.1. **Thesis organization.**

Kaleidoscopic 3D reconstruction	Multi-view 3D reconstruction
Kaleidoscope calibration	**Camera calibration**
input: {kaleidoscopic corerspondences}	*input*: {multi-view correspondences}
output: {mirror geometry}	*output*: {camera geometry}
Kaleidoscopic structured light	**Structured light**
input: {kaleidoscopic correspondences, mirror geometry}	*input*: {multi-view correspondences, camera (projector) geometry}
output: {object geometry, label}	*output*: {object geometry}
Kaleidoscopic neural rendering	**Inverse rendering**
input: {a kaleidoscopic image, mirror geometry}	*input*: {multi-view images, camera geometry}
output: {object geometry, label}	*output*: {object geometry}

- **Redistribution of viewpoints.** Covering the entire object requires diverse viewpoints especially when the object shape includes intricate geometry (e.g., strong self-occlusions). Compared to the approaches using turntable where the every viewpoint lies on a plane perpendicular to the rotation axis, the reflections in a kaleidoscope can redistribute rays in 3D space and thereby produce the virtual viewpoints all around the object. A proper design of the kaleidoscope can optimize the redistribution of viewpoints and enables the maximum utilization of camera pixels.

- **Synchronization.** Every camera should be synchronized for 3D reconstruction in a multi-camera system, and it is practically challenging especially when there are hundreds of cameras for full-surround 3D reconstruction. Kaleidoscopic imaging does not suffer from this issue as the virtual cameras generated by a kaleidoscope are synchronized with the speed of light.

- **Identical intrinsic parameter.** The virtual cameras generated by a kaleidoscope share the intrinsic parameter with the real camera, which facilitates the computation of epipolar geometry.

To motivate the utilization of mirrors and kaleidoscopes in 3D reconstruction, in this chapter, we begin by outlining the background and previous methods used for full-surround 3D reconstruction. We then compare these traditional approaches to kaleidoscopic 3D reconstruction. Additionally, we present the pipeline of kaleidoscopic 3D reconstruction as a counterpart to the multiple view geometry in classical computer vision. We highlight the challenges that complicate the direct adaptation of traditional techniques to the kaleidoscopic approach.

Table 1.2. **Comparison of surround reconstruction methods.**

Method	Registration	Correspondence	Labeling
Single camera	x	x	-
Multi-camera system	o	x	-
Multi-projector camera system	o	o	-
Camera + mirrors [Reshetouski *et al.*, 2011]	o	x	visual hull
ToF + mirrors [Xu *et al.*, 2018]	o	o	path length
Structured light + mirrors [Lanman *et al.*, 2009]	o	o	manual
Structured light + mirrors [Ahn *et al.*, 2021a]	o	o	epipolar constraint

1.3 Background: 3D reconstruction with mirrors

We review prior work on 3D reconstruction usiing mirrors. We summarize key features of our and prior techniques in Table 1.2.

Label-free approaches. There is extensive literature on the use of mirrors for 3D reconstruction, in combination with passive-illumination camera systems [Forbes *et al.*, 2006; Fuchs *et al.*, 2013; Gluckman and Nayar, 2001, 2002; Goshtasby and Gruver, 1993; Hu *et al.*, 2005; Huang and Lai, 2006; Mitsumoto *et al.*, 1992; Murray, 1995; Nene and Nayar, 1998; Taguchi *et al.*, 2010a,b; Ying *et al.*, 2010], time-of-flight (ToF) cameras [Nobuhara *et al.*, 2016], and projector-camera systems [Bangay and Radloff, 2004; Garg *et al.*, 2006; Han and Perlin, 2003; Lanman *et al.*, 2009; Tahara *et al.*, 2015]. The use of mirrors is in large part due to the increase in viewpoints they provide. In most systems using mirrors this way, the number of reflections and virtual viewpoints are carefully controlled to be few in number, which makes manual labeling practical.

Lanman *et al.* [2009] propose a structured light system that combines an orthographic projector, a camera, and mirrors, and is designed to remove the interference between reflected projector patterns. They solve the interference problem by illuminating with patterns that are perfectly aligned after one or multiple mirror reflections. However, achieving this requires using a special configuration of mirrors, which in turn constrains the locations of virtual cameras to lie on a plane. Such a viewpoint set can be insufficient for full-surround 3D coverage when scanning complex objects. Additionally, their configuration has only four virtual cameras, to make manual labeling of the virtual images observed in the camera practical. Tahara *et al.* [2015] extend this approach to perspective projectors, again with manual labeling. Kaleidoscopes have also been used for measuring bidirectional texture functions [Bangay and Radloff, 2004; Han and Perlin, 2003]; in these works, the underlying shapes are nearly planar, which

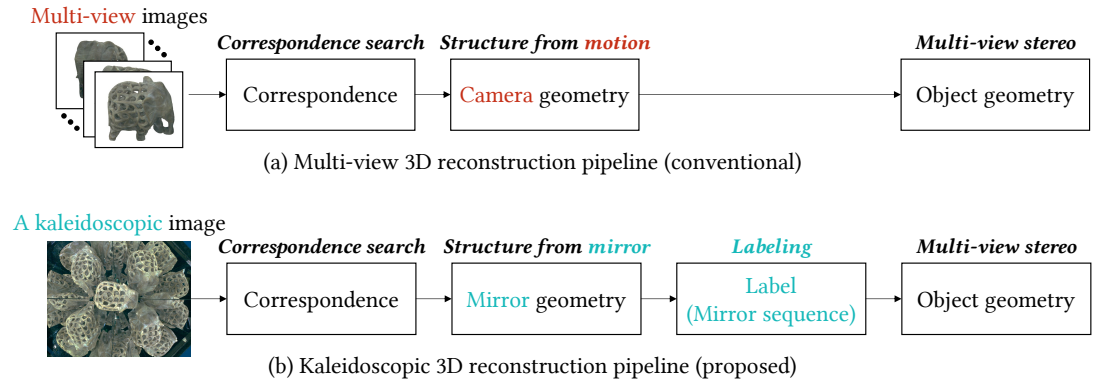
simplifies solving the labeling problem.

Approaches that estimate labels. Using numerous virtual viewpoints for better full-surround coverage requires being able to estimate labels automatically. Reshetouski *et al.* [2011] solve the labeling problem in a passive-illumination kaleidoscopic system by using space carving [Kutulakos and Seitz, 2000]. As the background pixels on the captured image do not intersect with the object, even after repeated mirror reflections, the rays corresponding to those pixels can be backprojected and "carved". This provides a visual hull of the object inside the kaleidoscope, which can be combined with ray tracing to obtain the label map. Ihrke *et al.* [2012] combine this labeling method with a structured light system that illuminates only one label at a time, to obtain correspondences while avoiding interference. As these methods rely on space carving for labeling, they are inaccurate when the visual hull is not a good approximation to the object; typically, this is the case for concave objects. Xu *et al.* [2018] combine a kaleidoscope with a ToF camera. The ToF information provides the total path length from the camera to the object. This allows estimating the label and 3D point, by folding the ray using the known mirror configuration. Unfortunately, this approach requires using a pulsed ToF system, which can be costly.

1.4 Reconstruction pipeline and challenges

Pipeline. A kaleidoscope provides a large number of diverse viewpoints. Thus, a kaleidoscopic imaging system can be considered as a virtual multi-view imaging system and thereby can follow a similar pipeline with the conventional real multi-view imaging system as follows:

- **Correspondence search.** Correspondences are generally obtained by feature matching (i.e., passive multi-view stereo) or by structured light (i.e., active multi-view stereo). However, the traditional active scanning scheme such as column scanning is in applicable in the kaleidoscopic imaging because of the interference. We explore a scanning technique for correspondence search in a kaleidoscope.
- **Structure from mirror.** In the conventional multi-view approach, *structure from motion* optimizes the camera parameters and is one of the key factors for high-quality 3D reconstruction. In this thesis, we address *structure from mirror* that optimizes mirror geometry from the correspondences in a kaleidoscopic image for high-quality 3D reconstruction from an uncalibrated kaleidoscope.
- **Labeling.** The kaleidoscopic 3D reconstruction has an additional step of labeling as the assignment of a pixel to a viewpoint is changed depending on the object shape, whereas it is fixed in the conventional multi-view reconstruction. This makes the kaleidoscopic imaging particularly challenging, and



(a) Multi-view 3D reconstruction pipeline (conventional)

(b) Kaleidoscopic 3D reconstruction pipeline (proposed)

Figure 1.2. **Pipeline of kaleidoscopic 3D reconstruction.** The kaleidoscopic 3D reconstruction shares a similar pipeline with the conventional multi-view 3D reconstruction pipeline, but each step cannot be directly adapted because of the several additional challenges including the labeling problem. Our goal is to address each step of the kaleidoscopic 3D reconstruction pipeline.

we explore the labeling problem by exploiting the epipolar geometry between virtual cameras and optionally projectors.

- **Kaleidoscopic neural rendering.** Additional to conventional 3D reconstruction pipeline, we explore to combine a kaleidoscope with the inverse rendering techniques that has recently seen immense success for 3D reconstruction.

Figure 1.2 illustrates the pipelines of both approaches.

Challenges. Each step of the pipeline cannot be directly adapted from the conventional counterpart because of the following challenges in kaleidoscopic 3D reconstruction.

- **Labeling problem.** The key challenge when using a kaleidoscope lies in interpreting the captured image and decoding the numerous views of the object that it provides. The assignment of a pixel to a virtual viewpoint is unknown in kaleidoscopic 3D reconstruction, whereas it is naturally known in the conventional 3D reconstruction. Specifically, we need to identify the virtual viewpoint corresponding to each pixel on the captured image. We call this the labeling problem. In the absence of this information, a correspondence cannot lead to a triangulation because the starting point of the triangulation is unknown.

- **Interference.** A kaleidoscope generates not only virtual cameras but also virtual light sources. As a consequence, we encounter complex interference patterns inside the kaleidoscope as a point on the object is illuminated multiple times via different mirror sequences or by different virtual light sources. The interference makes the correspondence search challenging as traditional scanning scheme for structured light (e.g., column scanning) inapplicable.

1.5 Thesis contributions

Thesis statement. In this thesis, we establish the theoretical and practical foundations of kaleidoscopic techniques for full-surround 3D reconstruction. We are particularly interested in the reconstruction of highly complex objects with intricate geometry including self-occlusions.

Based on the above contributions, this thesis establishes a theory of kaleidoscopic geometry, and explores the practices of kaleidoscopic 3D reconstruction including: (i) kaleidoscopic calibration; (ii) kaleidoscopic structured light; and (iii) kaleidoscopic neural rendering. The table 1.1 summarizes the thesis organization. This thesis also delves into the design of kaleidoscopes, detailing the various design choices and the metrics to evaluate their effectiveness.

2 Kaleidoscope Design and Calibration

In this chapter, we delve into the kaleidoscope design and their calibration processes. We initially establish the fundamental design considerations for constructing a kaleidoscope, and introduce a set of metrics to assess and compare various kaleidoscope designs. With this design consideraions and metrics, we simulate multiple different kaleidoscopes and evaluate their efficacy quantiatively. Additionally, we introduce the method for calibrating the extrinsic parameters of the mirrors.

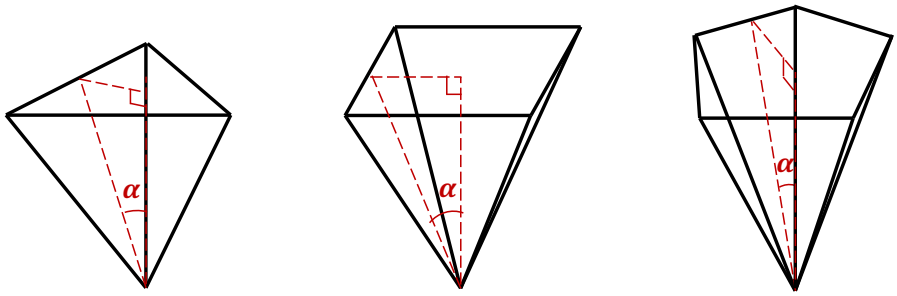
2.1 Kaleidoscope design

Our exploration into kaleidoscope design involves identifying key factors critical for its construction and function. This involves establishing specific design considerations and developing metrics to evaluate these designs quantitatively.

2.1.1 Design considerations

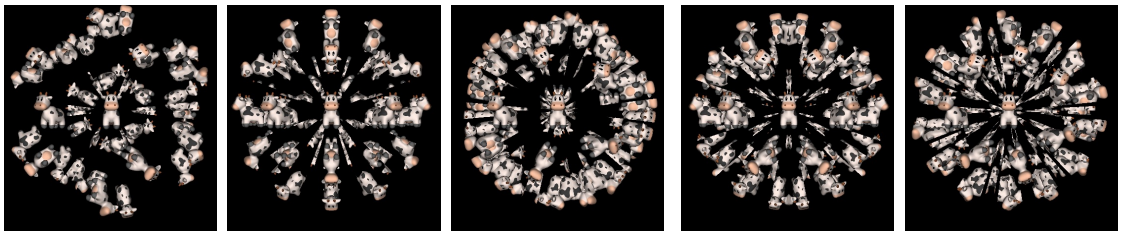
In our design process, we primarily focus on planar mirrors due to their ease of construction and calibration, which is consistent with the original purpose of buliding kaleidoscopes compared other imaging setup. In particular, we examine various pyramid shapes for the kaleidoscope, such as triangular and square pyramids. These shapes are chosen for their ability to virtually encircle the object, providing comprehensive coverage. Our specific focus is on the right pyramid structure, selected for its demonstrated optimal coverage [Xu *et al.*, 2018]. The key aspects of our design exploration include:

- **Number of mirrors.** Determining the appropriate number of mirrors for the side of the pyramid (e.g., triangular or square pyramid).
- **Configuration of mirrors.** Deciding on the assembly of the mirrors, with considerations for the apex angle of the pyramid, affecting the overall shape (e.g., sharp or wide pyramid).



(a) $N = 3$ (b) $N = 4$ (c) $N = 5$

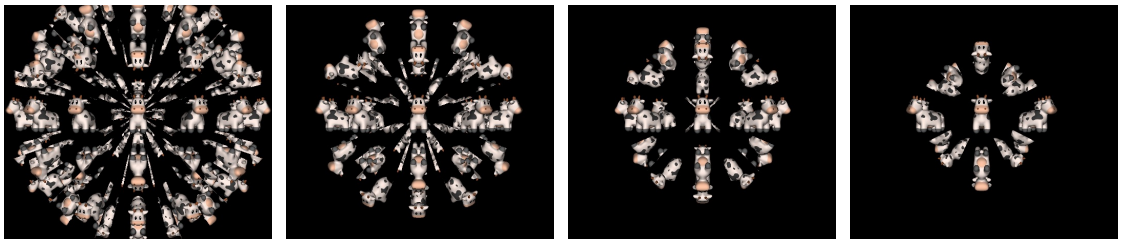
Figure 2.1. **Design considerations: pyramid shapes.** We examine various pyramid shapes for the kaleidoscope, with different number of mirrors (i.e., number of sides N), configurations (i.e., apex angle α), and sizes. We illustrate example pyramids with different design parameters and visualize the apex angle in each case. The object will be placed inside the kaleidoscope, and the camera will be located at the base side of the pyramid, looking at the tip of the pyramid.



(a) $N = 3$ (b) $N = 4$ (c) $N = 5$ (d) $N = 6$ (e) $N = 7$

Figure 2.2. **Example kaleidoscopic images: number of mirrors.** We show example kaleidoscopic images with different number of mirrors N, with apex angle $\alpha = 18°$.

- **Size of mirrors.** The mirror size is a critical factor, especially when considering larger objects that might occlude mirrors and reduce reflection efficiency (e.g., small or large pyramid). Interestingly, changing size of mirrors is equivalent to adjusting the distance between the camera and the mirrors when the mirrors are sifficiently large to cover the field of view. Thus, we explore size of mirrors by adding offset to the distance from camera to the kaleidoscope.



(a) $\alpha = 12°$ (b) $\alpha = 18°$ (c) $\alpha = 24°$ (d) $\alpha = 30°$

Figure 2.3. **Example kaleidoscopic images: apex angle.** We show example kaleidoscopic images with different apex angles α, with number of mirrors $N = 4$.

2.1.2 Evaluation metrics

To assess the performance of our kaleidoscope designs, we focus on two main aspects: coverage and accuracy. Coverage is evaluated based on how much portion of the object surface is observable by the camera at least once, following methodologies established in Xu *et al.* [2018]. Accuracy is measured by examining both the quantity and distribution of rays in each observation, which are directly related to triangulation.

Coverage. To quantify the coverage, we simulate a kaleidoscope with an object and run ray tracing, and define the coverage as percentage of mesh triangles that is observed at least once by the camera. This metric assesses whether rays intersect with a given triangle. By averaging this metric across all triangles, we calculate the percentage of total triangles that are observed by the camera, thereby indicating coverage.

Accuracy. The accuracy of 3D reconstruction is determined by triangulation, which depends on two key factors of the rays in the correspondence: (i) the number of rays, and (ii) the diversity of ray directions, based on the traditional baseline analysis in triangulation. Quantifying the number of rays is straightforward through kaleidoscope simulation and ray tracing. To measure the diversity of ray directions, we define two metrics: ray angles and ray conditioning. These metrics are described as follows:

- **Ray quantity.** This metric is defined as the number of rays per triangle. It is calculated by counting rays originating from distinct viewpoints. It's important to note that a single triangle, particularly in cases where pixel resolution is higher compared to triangle resolution, can receive multiple rays from

the same virtual viewpoint (i.e., the same sequence of mirrors). However, these repeated rays from an identical viewpoint do not aid in triangulation. Therefore, for our purposes, we only count rays that are from unique viewpoints.

- **Ray angles.** This metric is the average of pair-wise angles between rays. We use the same set of rays as counted for the 'ray quantity' metric to calculate this, meaning that only rays originating from distinct viewpoints are included in this computation.
- **Ray conditioning.** This metric is the condition number of the triangulation matrix. We recognize that angles might not fully represent ray diversity in 3D space. Therefore, we conduct conditioning analysis, to evaluate how much the triangulation result can change for a small change in the ray origin and direction. This can be achieved by treating triangulation as a linear inverse problem (detailed in Chapter 3) as $\mathbf{q} = \mathbf{A}^{-1}\mathbf{b}$ and computing the condition number of the triangulation matrix $\mathbf{A}$, where $\mathbf{q}$ denotes the 3D location nearest to all rays, and $\mathbf{A}$ and $\mathbf{b}$ are derived from the origins and directions rays in correspondences.

2.1.3 Analysis and discussion

To evaluate and analyze the kaleidoscope design, we compute the metric from the simulation with multiple different meshes with different poses with each kaleidoscope design. For the simulation, we assume the object size is 5 cm in diameter, and the distance from the camera to the object is 67.5 cm.

Tables 2.1–2.4 present the results for coverage, ray count, ray angles, and ray conditioning, respectively. These results are shown for varying numbers of mirrors N and apex angles α. For each combination, the mirror size with the best metric selected for display. The following section summarizes the impact of each design consideration.

Number of mirrors. The tables indicates that the number of mirrors N does not significantly impact coverage when $N > 2$, at which point the virtual viewpoint resides on a 2D plane. In terms of accuracy, a greater number of mirrors increases the quantity of rays. However, the angles and conditioning show no significant correlation with the number of mirrors.

Configuration of mirrors. It is evident that smaller apex angles enhance the metrics in coverage, both from Table 2.1 and Figure 2.3. With respect to accuracy, ray quantity is notably higher with smaller apex angles, as a sharper triangle offers a more densely distributed virtual camera space, optimizing pixel utilization. Nevertheless, ray angles and conditioning appear to be uncorrelated with the apex angle.

Table 2.1. **Design analysis: coverage.** Coverages for different kaleidoscope designs.

α \ N	2	3	4	5	6	7	8	9	10
12°	77.73%	97.73%	97.80%	97.90%	98.02%	98.50%	98.26%	98.14%	97.80%
18°	76.20%	95.87%	96.04%	97.08%	96.65%	96.84%	97.23%	96.16%	97.05%
24°	73.29%	92.50%	93.05%	95.61%	94.64%	95.42%	95.17%	96.26%	95.25%
30°	68.68%	92.33%	90.73%	91.94%	90.00%	92.93%	90.32%	93.43%	90.66%
36°	71.07%	82.04%	88.34%	80.19%	91.01%	84.65%	92.20%	87.79%	93.24%

Table 2.2. **Design analysis: ray quantity.** Average number of observed rays per triangle for different kaleidoscope designs.

α \ N	2	3	4	5	6	7	8	9	10
12°	3.63	15.11	16.42	20.44	19.40	22.96	21.09	22.24	22.59
18°	2.97	10.27	10.30	14.27	12.29	13.90	13.30	13.65	14.27
24°	2.47	5.79	6.56	10.66	7.89	9.83	8.82	11.13	9.83
30°	1.66	5.43	4.49	5.66	4.25	6.29	4.85	7.44	5.65
36°	1.93	2.47	4.03	2.78	4.59	3.80	5.67	4.97	6.67

Size of mirrors. Table 2.5 demonstrates the impact of mirror size, controlled by adjusting the distance between the camera and the kaleidoscope, showcasing metrics for a specific mirror count ($N = 4$) and apex angle ($\alpha = 18°$). The findings suggest that increased offsets yield better coverage and accuracy results. This improvement is attributed to larger mirrors minimizing occlusion among reflected objects, thereby allowing more efficient pixel utilization. The offset is set to zero when the distance from the mirror to the object's center equals the radius of the object (i.e., 2.5 cm).

Discussion. From the analysis, we conclude that a smaller apex angle significantly enhances coverage and ray quantity, contributing to accuracy, while a larger mirror size improves metrics across both coverage and accuracy. Additionally, increasing the number of mirrors can marginally boost coverage.

However, we also need to consider practical issues as well as theoretical analysis to construct a real

Table 2.3. **Design analysis: ray angles.** Average pair-wise angles between rays per triangle for different kaleidoscope designs.

α \ N	2	3	4	5	6	7	8	9	10
12°	20.39	41.06	91.52	87.54	61.77	91.72	29.04	57.02	28.29
18°	33.09	86.72	35.99	71.84	53.95	37.08	47.35	49.53	68.14
24°	65.01	46.67	44.83	42.98	40.52	49.12	53.83	68.45	19.15
30°	110.63	99.65	48.08	37.88	37.79	85.14	36.82	33.92	57.88
36°	136.38	102.74	37.04	70.12	31.80	49.99	55.79	38.89	19.55

Table 2.4. **Design analysis: ray conditioning.** Average condition number of triangulation matrix for different kaleidoscope designs.

α \ N	2	3	4	5	6	7	8	9	10
12°	6.89	2.41	2.46	2.81	3.07	2.52	2.36	3.31	2.75
18°	5.33	2.92	2.46	2.81	2.32	2.77	2.48	3.63	2.41
24°	3.36	2.29	12.00	3.49	2.93	2.42	3.07	2.60	3.61
30°	3.26	3.05	2.82	18.18	7.74	6.48	3.01	17.46	3.27
36°	2.95	2.43	3.43	2.44	2.47	2.91	2.54	3.05	2.81

kaleidoscope. Two primary issues are system size and mirror boundaries.

First, system size needs to be considered for efficient kaleidoscopic imaging. A small apex angle combined with large mirrors leads to an excessively long kaleidoscope. This design complicates object placement, camera mounting, and robust kaleidoscope construction, hindering the ease of use. Thus, when the improvement is marginal, maintaining moderate values for apex angle and mirror size is advisable.

Second, real-world setups face challenges with pixels near mirror boundaries. These areas are unstable due to gaps between mirrors and calibration errors. Minor inaccuracies can dramatically alter reflection sequences, resulting in signifincatly different reconstruction. Figure 2.2 illustrates how large mirror counts reduce segment size, exacerbating these issues. As the number of mirrors increases, the segment

Table 2.5. **Design analysis: size of mirrors.** $N = 4$, $\alpha = 18^\circ$

offset	coverage	ray quantity	condition	angles
0 mm	94.24%	8.36	2.79	56.80
10 mm	94.97%	8.94	2.67	66.70
20 mm	95.32%	9.49	2.56	51.89
30 mm	95.51%	9.97	2.54	45.35
40 mm	95.64%	10.34	2.48	56.12

size diminishes, meaning rays are more likely to hit near mirror boundaries during reflections. In practice, these boundary areas pose significant problems, as small discrepancies can lead to large errors in final triangulations. Therefore, it is preferable to avoid relying on pixels near the boundary regions, and thereby use moderate number of mirrors.

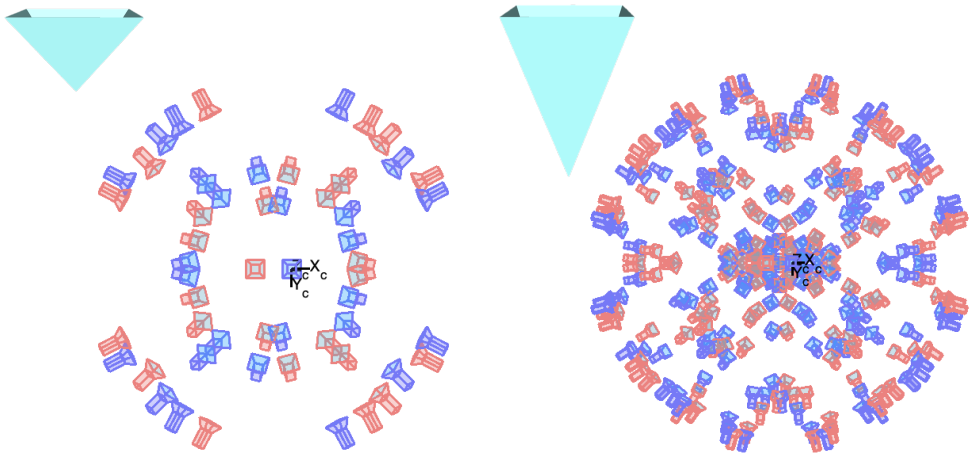
Further analysis with square-pyramid shape. We analyze the square pyramid additionally, which shows good coverage and accuracy, with small boundary region, and also proved to be effective from Xu *et al.* [2018]. Having more viewpoints is important for improving the accuracy of the multi-view triangulation procedure our technique uses. This is different from the methodology of Xu *et al.* [2018], where the ToF depth-sensing mechanism does not require having multiple views of the same point. Having more views also helps improve coverage, and thus can enable reconstructing occluded parts of objects with highly complex visibility. Fig. 2.4 shows that the sharp pyramid provides significantly more virtual projectors and cameras than one with a larger angle. The sharp configuration also provides good scanning coverage, as we show empirically in Fig. 2.5, where we visualize the number of projector and camera pixels that observe each vertex of an object mesh. We leave the systematic optimization of the kaleidoscope configuration as an important future research direction.

2.2 Kaleidoscope calibration

We describe how we calibrate kaleidoscopes by obtaining mirror geometry from a kaleidoscopic image.

2.2.1 Initial calibration

We calibrate our projector-camera pair using the algorithms of Zhang [2000] and Moreno and Taubin [2012]. We calibrate the kaleidoscope using the algorithm of Takahashi *et al.* [2017]; Takahashi and



(a) Wide pyramid (b) Sharp pyramid

Figure 2.4. **Mirror design.** Virtual projectors and cameras produced by pyramidal kaleidoscopes with different tip angles. A sharp pyramid produces more virtual projectors and cameras, which can provide better coverage. The apex angles are (a) 36° and (b) 18°, and the number of the virtual cameras (or projectors) are 27 and 145, respectively.

Nobuhara [2021], which estimates the location and pose of the mirrors relative to the camera from correspondences of a single 3D point. We refer readers to Takahashi *et al.* [2017] for the initial calibration of pose of the mirrors, which can be formualted as a linear inverse problem.

2.2.2 Kaleidoscopic bundle adjustment

To improve upon this initial calibration, we use a bundle adjustment procedure inspired by Xu *et al.* [2018]. As we show in Fig. 2.6, we optimize the parameters of the projector, camera, and mirrors, based on scanning results for a reference object (sphere of diameter 40 mm). We first sparsely scan the reference sphere for a few pixels that can be easily labeled manually (e.g., direct and one-bounce), and fit a sphere to the reconstructed point cloud. With this initial sphere fitting result, we can label every pixel using ray tracing, and completely reconstruct the object. Then, we update the extrinsic parameters of the projector and mirrors relative to the camera by minimizing an objective combining triangulation error (distance of a reconstructed point from its corresponding backprojected rays), reprojection error (distance of the projection of a reconstructed point from its corresponding pixel locations), and sphere fitting error of the reconstructed point cloud. After bundle adjustment, we achieve a root-mean-square triangulation error

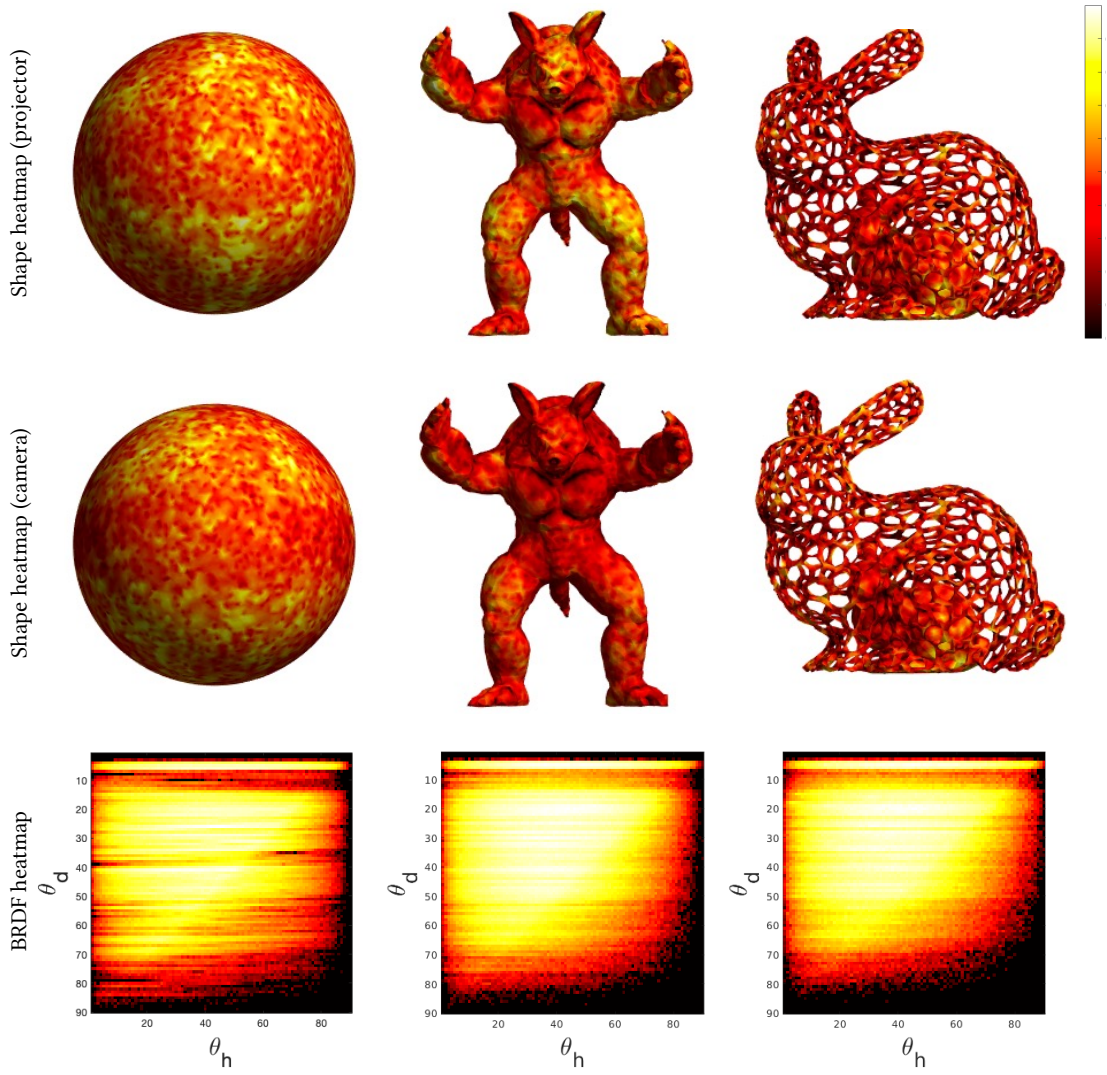
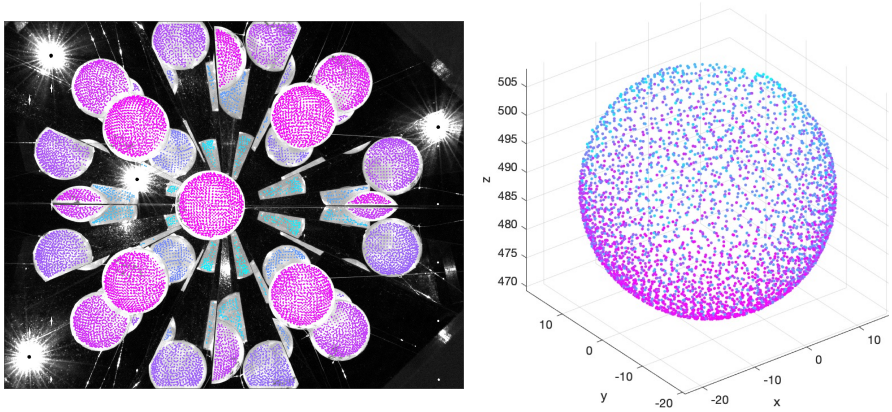


Figure 2.5. **Surface coverage.** Our kaleidoscopic structured light system generates hundreds of virtual projectors and cameras, which densely sample the light-view space. The heatmaps visualize the number of camera and projector pixels that intersect each surface facet (normalized in each object), and the BRDF heatmap shows the number of samples in each light-view angle in bivariate space.

of 33 µm, reprojection error of 1.3 pixels, and sphere fitting error of 361 µm.



(a) Manual label (from initial sphere fitting) (b) Bundle adjustment results

Figure 2.6. **Calibration.** We calibrate the projector, camera, and mirrors using a reference spherical object of known diameter. We manually label pixels using an initial sphere fitting result, and perform bundle adjustment to minimize triangulation error, sphere fitting error, and reprojection error.

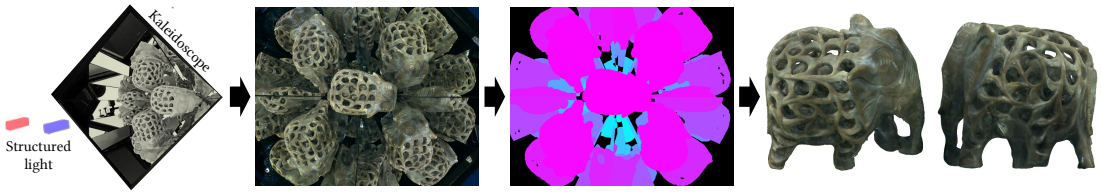
3 Kaleidoscopic Structured Light

In this chapter, we introduce kaleidoscopic structured light, which serves as the kaleidoscopic counterpart to traditional structured light. The kaleidoscope enhances the conventional structured light system by virtually replicating the projector and camera setup. It behaves as a virtual multi-view structured light system, simulating the hundreds of projectors and cameras, which would be challenging to achieve in a real-world setup.

The key challenge when using a kaleidoscope, lies in interpreting the captured image and decoding the numerous views of the object that it provides. Specifically, we need to identify the virtual viewpoint corresponding to each pixel on the captured image. We call this the *labeling* problem. The labeling information allows us to decompose the single captured image into multiple segments, one for each virtual viewpoint. In the absence of this information, we cannot estimate the 3D shape by triangulating from correspondences across different views. The fact that in a kaleidoscope it is common to observe hundreds of virtual views that are interwoven with each other makes the labeling problem particularly challenging.

We propose a full-surround 3D imaging system that we call kaleidoscopic structured light, comprising a projector, a camera, and a kaleidoscope. Our main technical result is to show that we can correctly label the virtual projectors and virtual cameras for arbitrary kaleidoscope configurations, by using their *epipolar geometry* and other physical constraints arising from image formation for this setup. With this result, our kaleidoscopic structured light system can serve as a multi-view structured light system with tens, if not hundreds, of virtual projectors and cameras, which are hard to construct with real devices. Our system allows us to reconstruct shape (in the form of triangular meshes) of highly complex objects with intricate features, by providing correspondences from multiple viewpoints with full-surround coverage.

Contributions. Our work advances the state of the art of 3D scanning, by facilitating the reconstruction of challenging objects, such as textureless or glossy objects, with complex 3D geometry. This is made



(a) Imaging setup (b) Scanning (c) Labeling (d) Full surround 3D reconstruction

Figure 3.1. **Kaleidoscopic structured light.** (a) We propose a system for full-surround 3D imaging using an imaging setup that consists of a projector, a camera, and a kaleidoscope. (b) The camera and projector observe the object from a large number of virtual viewpoints, which unravel the complex geometry of the object. (c) To use the kaleidoscopic image for multi-view stereo, we label each pixel, i.e., identify the sequence of mirrors that the ray backprojected from the pixel encounters before intersecting the object. (d) This labeling allows us to reconstruct the shape of the object using multi-view triangulation. Our system enables us to reconstruct, with high accuracy and full coverage, highly complex objects that have intricate geometric features, including concavities and self-occlusions. Reconstructed point clouds and full-surround videos are available on the project webpage [Ahn *et al.*, 2021b].

possible through the following contributions.

- *Labeling using epipolar geometry.* We develop a labeling technique that takes as input correspondences between a projector pixel illuminating a point on the object, and multiple camera pixels observing that point. Our labeling technique uses constraints specific to the kaleidoscopic setup, and in particular the epipolar geometry across these multiple correspondences, to jointly decode the mirror sequences at the projector and camera pixels.

- *Theoretical guarantees on the recoverability of labels using epipolar geometry.* We prove a theoretical result that establishes the uniqueness and correctness of the labels we decode with our labeling technique. Specifically, we show that the labels we recover from the projector-camera pixel correspondences are accurate, provided the epipolar planes corresponding to the projector-camera geometry are not parallel to the mirror normals. This condition is easy to achieve by placing the camera and projector in asymmetric poses relative to the kaleidoscope.

- *Fast scanning techniques.* We develop heuristics for speeding up our scan, exploiting the fact that the labeling problem is highly over-constrained due to multiple camera pixels mapping to the same projector pixel.

These advances in kaleidoscopic imaging allow us to push the boundaries of 3D scanning in terms of the

complexity of the objects that can be reconstructed. Fig. 3.1 shows an example 3D scan of an object that fully encompasses another smaller object, and is characterized by concavities, a large genus number, and complex self-occlusion. We have released our code and data on the project website [Ahn *et al.*, 2021b], to facilitate reproducibility and follow-up research.

Limitations. The main limitation of the proposed kaleidoscopic structured light technique is the time it requires for scanning an object. The baseline version of our technique relies on point scanning, which can take a prohibitive amount of time. We provide parallelized scanning techniques to speed up acquisition, but our scan times remain considerably longer than those achieved by traditional structured light acceleration techniques. Unfortunately, the complex nature of kaleidoscopic light transport makes these traditional techniques inapplicable to our setting.

3.1 Overview

We provide an overview of the proposed kaleidoscopic structured light system and its associated reconstruction pipeline.

3.1.1 Imaging setup

Hardware. We construct the kaleidoscopic structured light system by combining a projector, a camera, and a kaleidoscope. We create the kaleidoscope by placing four planar mirrors in a pyramidal shape, which Xu *et al.* [2018] report empirically to provide good scanning coverage. We place the projector and camera at the bottom of the pyramid, oriented to look at its tip. We calibrate the intrinsic parameters of the projector and camera, and the extrinsic pose of the projector and mirrors relative to the camera. Then, we place the object inside the mirror system, either by hanging it with strings, or by placing it on the mirrors directly. The kaleidoscopic arrangement provides hundreds of virtual projectors and cameras. We show a schematic of our setup in Fig. 3.1, and provide details about our hardware prototype and calibration procedure in Section 3.4. Fig. 3.1 also outlines the steps of the imaging pipeline—scanning, labeling, and shape reconstruction—which we review next.

Scanning. We first scan the object by turning "on" a number of projector pixels, which illuminate a set of object locations, and thus camera pixels. To simplify exposition, we describe our approach assuming that we activate only a single projector pixel at a time. In Section 3.4, we discuss techniques for speeding up acquisition by simultaneously activating multiple projector pixels. The projector pixel that we activate

illuminates a single point on the object surface, either directly or after one or multiple reflections on the kaleidoscope. The illuminated point is observed at multiple camera pixels, each via a different sequence of mirror reflections, and we save the locations of these pixels. This provides us with a correspondence between a single projector pixel and multiple camera pixels, all of which also correspond to a single object point. This interpretation assumes that interreflections on the object are weak enough to not overwhelm the direct observation of the illuminated point; in our experience, this is generally true except for when the scanned object is a mirror or highly-specular.

Labeling. We now have a correspondence between a single projector pixel and multiple camera pixels. However, we cannot directly use this correspondence for triangulation, because we do not know the mirror sequence encountered by the rays corresponding to the projector and camera pixels. For each projector or camera pixel, we refer to the sequence of mirrors that the ray from the pixel encounters before intersecting the object as the pixel's *label* [Reshetouski *et al.*, 2011]. We refer to the task of determining the labels for all projector and camera pixels as the *labeling problem*. Solving the labeling problem is the core challenge of kaleidoscopic imaging. Labeling and shape reconstruction are interwoven, as labeling requires reasoning about the object shape, and shape reconstruction requires using the labels to perform triangulation. We will define the labeling problem more formally in Section 3.1.2, and show how to solve it using epipolar geometry constraints in Section 3.2.

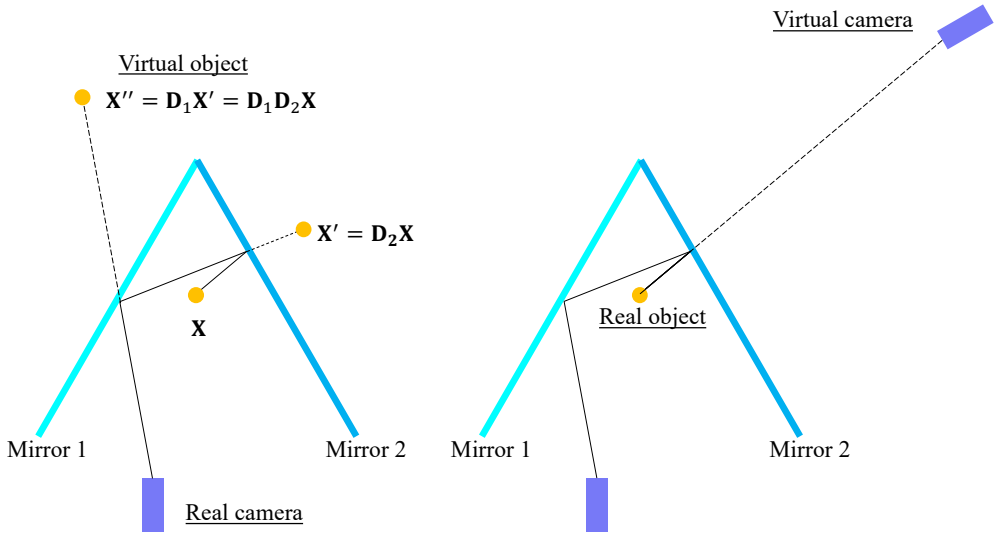
Shape reconstruction. The projector-camera correspondences and labeling information allow us to reconstruct 3D geometry using multi-view triangulation: For each projector pixel, we use its correspondence with multiple camera pixels to estimate a 3D point that comes closest to intersecting all pixel rays (Section 3.3).

3.1.2 Basics of kaleidoscopic imaging

Before we introduce how to solve the labeling problem, we review the transformation of rays and points by planar mirrors, and the epipolar geometry between the virtual projectors and cameras.

Transformation by planar mirrors. To represent the transformation by a single planar mirror m, we define the mirror as a plane with normal $\mathbf{n}$ and distance from origin d (represented in world coordinates). Then, the 4×4 reflection matrix in homogeneous coordinates can be written as

$$\mathbf{D}_m = \begin{bmatrix} \mathbf{I} - 2\mathbf{n}\mathbf{n}^\top & 2d\mathbf{n} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{3.1}$$



(a) Virtual-object interpretation (b) Virtual-camera interpretation

Figure 3.2. **Mirror transformation interpretations.** We can interpret the mirror transformation in two mathematically equivalent ways: (a) *Virtual object*—the location of the *virtual* point in the *real* camera coordinate system, corresponding to unfolding the ray from the object side. (b) *Virtual camera*—the location of the *real* point in the *virtual* camera coordinate system, corresponding to unfolding the ray from the camera side.

Note that $\mathbf{D}_m$ is an involutory matrix (i.e., $\mathbf{D}_m^2 = \mathbf{I}$), as the reflection of a reflected point is the same as the original point.

The transformation by multiple planar mirrors can be represented by multiplying the reflection matrices $\{\mathbf{D}_m\}$ corresponding to each mirror. Consider the example of Fig. 3.2: a ray from a 3D point $\mathbf{X}$ bounces off mirrors 2 and 1 before being observed at a camera pixel, as shown in Fig. 3.2. Then, the intermediate virtual point after the first reflection at mirror 2 equals $\mathbf{X}' = \mathbf{D}_2\mathbf{X}$, and the final virtual point seeing $\mathbf{X}'$ through mirror 1 equals $\mathbf{X}'' = \mathbf{D}_1\mathbf{X}' = \mathbf{D}_1\mathbf{D}_2\mathbf{X}$.

Labels. As mentioned earlier, successful shape recovery requires us to decode the light path between a scene point and the projector or camera pixel observing it. Following Reshetouski *et al.* [2011], we represent this path using a sequence of mirror labels; for a kaleidoscope with M mirrors, and a light path with K reflections, the label sequence is:

$$\ell \equiv (l_k)_{k=1}^{K} = (l_1, l_2, l_3, \ldots, l_K), \tag{3.2}$$

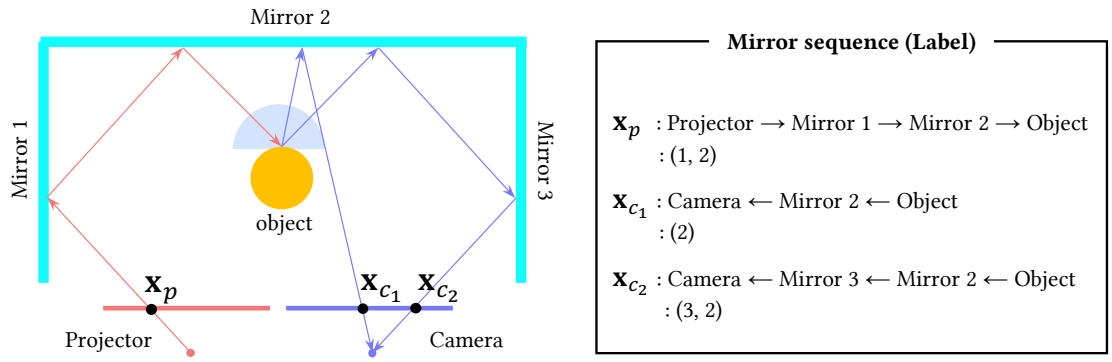


Figure 3.3. **Label definition.** The label of a projector or camera pixel is the sequence of mirrors that the ray backprojected from the pixel reflects off of before reaching the scanned object.

where $l_k \in \{1, 2, ..., M\}$ denotes a mirror label. We use the convention that the labeling starts at the pixel and ends at the object. For instance, in Fig. 3.3, the number of mirrors are $M = 3$, the labels for the light path shown are $\ell_p = (1, 2)$, $\ell_{c_1} = (2)$, and $\ell_{c_2} = (3, 2)$. With this label definition, the mirror transformation matrix $\mathbf{D}(\ell)$ can be written as a function of the label ℓ as

$$\mathbf{D}(\ell) = \prod_{k=1}^{K} \mathbf{D}_{l_k} = \mathbf{D}_{l_1}\mathbf{D}_{l_2}\mathbf{D}_{l_3}\cdots\mathbf{D}_{l_K}. \tag{3.3}$$

Virtual cameras and projectors. As Fig. 3.2 shows, the transformed 3D point $\mathbf{X}'' = \mathbf{D}\mathbf{X}$ can be interpreted in two ways: (i) *virtual object*—the location of the *virtual* point in the *real* camera coordinate system; or (ii) *virtual camera*—the location of the *real* point in the *virtual* camera coordinate system. The first interpretation corresponds to unfolding the ray from the object side, and the second to unfolding the ray from the pixel side. Each of these equivalent interpretations has its own advantages. For example, the virtual-object interpretation is useful when we derive the transformation by multiple mirrors, as it lets us avoid the change of camera coordinates. By contrast, the virtual-camera interpretation is useful when we derive expressions for triangulated 3D points.

We can use the mathematical equivalence of these two interpretations to represent the extrinsic matrix of the virtual camera $\mathbf{T}_{\text{virtual}}$: Let $\mathbf{T}_{\text{real}}$ be the 4×4 extrinsic matrix of the real camera (i.e., camera pose in the world coordinate system). Then, the virtual 3D point $\mathbf{X}_{\text{virtual}} = \mathbf{D}\mathbf{X}_{\text{real}}$ in the real camera coordinate system equals

$$\mathbf{T}_{\text{real}}\mathbf{X}_{\text{virtual}} = \mathbf{T}_{\text{real}}\mathbf{D}\mathbf{X}_{\text{real}} = \mathbf{T}_{\text{virtual}}\mathbf{X}_{\text{real}}. \tag{3.4}$$

This can be interpreted as a real 3D point in the local coordinate system of a virtual camera whose extrinsic matrix equals

$$\mathbf{T}_{\text{virtual}} = \mathbf{T}_{\text{real}}\mathbf{D}. \tag{3.5}$$

The above derivation applies, mutatis mutandis, to projectors, and their virtual counterparts. We skip the derivation as, for the most part, projectors can be treated as cameras.

Epipolar geometry of virtual projectors and cameras Now we can represent the virtual projectors and cameras using the parameters that define the real projector, camera, and mirrors. Let the extrinsic matrix of the real projector and camera be $\mathbf{T}_p$ and $\mathbf{T}_c$, respectively. Then, the extrinsic matrix of the virtual projector with mirror transformation $\mathbf{D}_p$ is $\mathbf{T}'_p = \mathbf{T}_p\mathbf{D}_p$, and that of the virtual camera with mirror transformation $\mathbf{D}_c$ is $\mathbf{T}'_c = \mathbf{T}_c\mathbf{D}_c$. Thus, a 3D scene point $\mathbf{X}$ observed from the virtual projector and the virtual camera equals

$$\begin{cases} \mathbf{X}_p = \mathbf{T}'_p\mathbf{X} = \mathbf{T}_p\mathbf{D}_p\mathbf{X}, \\ \mathbf{X}_c = \mathbf{T}'_c\mathbf{X} = \mathbf{T}_c\mathbf{D}_c\mathbf{X}, \end{cases} \tag{3.6}$$

where $\mathbf{X}_p$ and $\mathbf{X}_c$ are the representations of the point in the local coordinates of the virtual projector and virtual camera, respectively.

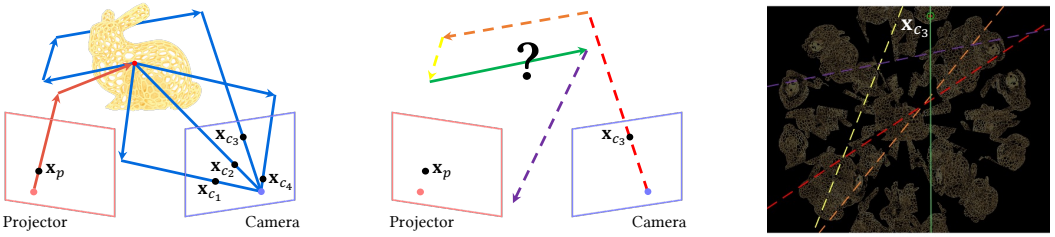
To mathematically describe the epipolar geometry between the virtual projector and virtual camera, we can express the relative transformation between them as

$$\mathbf{X}_p = \mathbf{T}_p\mathbf{D}_p\mathbf{X} = \mathbf{T}_p\mathbf{D}_p\mathbf{D}_c^{-1}\mathbf{T}_c^{-1}\mathbf{X}_c = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}\mathbf{X}_c, \tag{3.7}$$

where $\mathbf{R}$ and $\mathbf{t}$ are the relative rotation and translation between the virtual projector and virtual camera, respectively. Finally, to express the fundamental matrix, we denote by $\mathbf{K}_p$ and $\mathbf{K}_c$ the intrinsic matrices of the virtual projector and virtual camera, respectively (i.e., $\mathbf{x}_p \sim \mathbf{K}_p\mathbf{X}_p$, and $\mathbf{x}_c \sim \mathbf{K}_c\mathbf{X}_c$). Then, we can express epipolar constraints between the virtual projector and virtual camera using the fundamental matrix $\mathbf{F} = \mathbf{K}_p^{-\top}[\mathbf{t}]_\times\mathbf{R}\mathbf{K}_c^{-1}$, which satisfies $\mathbf{x}_p^\top\mathbf{F}\mathbf{x}_c = 0$.

3.2 Labeling

We now present our main technical results, detailing the conditions under which the label sequence can be uniquely determined using the epipolar constraints between virtual projectors and cameras.



(a) Correspondence from scanning (b) Labeling problem (c) Epipolar labeling

Figure 3.4. **Epipolar labeling.** We label how each pixel is reflected using epipolar labeling. (a) We obtain correspondences between a projector pixel and multiple camera pixels during scanning. (b) Labeling such a correspondence is equivalent to determining the number of reflections for each projector and camera pixel given the mirror geometry. We visualize the possible labels for the pixel $\mathbf{x}_{c_3}$. (c) We show on the camera image the epipolar lines corresponding to the possible labels, using matching colors. The green line passes through the pixel $\mathbf{x}_{c_3}$, which satisfies the epipolar constraint, whereas other lines do not. We prove that our epipolar labeling method can correctly determine the label for generic mirror configurations.

3.2.1 Problem setup

Suppose that we illuminate a pixel $\mathbf{x}_p$ on the projector and observe a set of pixels $\{\mathbf{x}_{c_i}\}$ on the camera. Then, we formulate the labeling problem as follows: given the correspondence of $\mathbf{x}_p \leftrightarrow \{\mathbf{x}_{c_i}\}$, find the label, or mirror sequence, ℓ_p associated with the projector pixel $\mathbf{x}_p$, and the label set $\{\ell_{c_i}\}$ for each of the camera pixels in $\{\mathbf{x}_{c_i}\}$.

We approach this problem as one of finding label for the projector pixel and each of the camera pixels, such that the projector-camera pixel correspondences satisfy the epipolar constraints implied by the labels. Let the mirror transformation of the projector pixel $\mathbf{x}_p$ be $\mathbf{D}_p = \mathbf{D}(\ell_p)$, and that of the camera pixel $\mathbf{x}_{c_i}$ be $\mathbf{D}_{c_i} = \mathbf{D}(\ell_{c_i})$, as defined in (3.3). Then, the relative transformation between the virtual projector and camera is $\mathbf{T}_p\mathbf{D}_p\mathbf{D}_{c_i}^{-1}\mathbf{T}_c^{-1} = \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \\ \mathbf{0} & 1 \end{bmatrix}$. The fundamental matrix can now be written as a function of the labels ℓ_p and ℓ_{c_i} as $\mathbf{F}(\ell_p, \ell_{c_i}) = \mathbf{K}_p^{-\top}[\mathbf{t}_i]_\times \mathbf{R}_i \mathbf{K}_c^{-1}$. The epipolar distance due to the labels ℓ_p, ℓ_{c_i} and the correspondence $\mathbf{x}_p \leftrightarrow \mathbf{x}_{c_i}$ can be defined as

$$d(\ell_p, \ell_{c_i}; \mathbf{x}_p, \mathbf{x}_{c_i}) \equiv |\mathbf{x}_p^\top \mathbf{F}(\ell_p, \ell_{c_i})\mathbf{x}_{c_i}|. \tag{3.8}$$

We refer to (3.8) as the *virtual epipolar distance*, and our goal is to find projector and camera labels such that the virtual epipolar distance is zero for each of the projector-camera correspondences. Note that all the correspondences share the same projector label. In the presence of noisy correspondences, we seek

to minimize the total virtual epipolar distance given as

$$\min_{\ell_p, \{\ell_{c_i}\}} \sum_i d(\ell_p, \ell_{c_i}; \mathbf{x}_p, \mathbf{x}_{c_i}). \tag{3.9}$$

3.2.2 Epipolar labeling

Optimizing (3.9) is not trivial, because there are exponentially many possible labels for both projector and camera; hence, exhaustive search is computationally intractable. The size of the search space for each projector or camera pixel is $O(M^{K_{\max}})$, where M is the number of mirrors and $K_{\max}$ is the maximum number of possible reflections in the kaleidoscope. In our prototype, $K_{\max} = 10$ (calculated by ray tracing using the calibration results). However, we can reduce the search space using the fact that the true label at a projector or camera pixel is a pre-subsequence of the label when there is no object in the kaleidoscope. In particular, we can precompute a pixel's *empty label* for an object-free kaleidoscope simply by ray tracing using the calibration results. The ray will be repeatedly reflected by the mirrors until it escapes the kaleidoscope. When we put an object in the kaleidoscope, the ray is truncated when it intersects the object. Thus, if we denote the empty label of a pixel $\mathbf{x}$ as

$$\ell^{\text{empty}}(\mathbf{x}) = (l_k)_{k=1}^{K_{\max}} = (l_1, l_2, l_3, \ldots, l_{K_{\max}}), \tag{3.10}$$

the label with an object will be a pre-subsequence of $\ell^{\text{empty}}(\mathbf{x})$,

$$\ell = (l_k)_{k=1}^{K} = (l_1, l_2, l_3, \ldots, l_K) \subset \ell^{\text{empty}}(\mathbf{x}). \tag{3.11}$$

Then, given the empty label, the labeling problem reduces to finding for each pixel the number of mirror reflections till intersecting the object. Let the number of reflections for $\mathbf{x}_p$ and $\mathbf{x}_{c_i}$ be K_p and K_{c_i}, respectively. We can rewrite (3.9) as

$$\min_{K_p, \{K_{c_i}\}} \sum_i d(\ell_p, \ell_{c_i}; \mathbf{x}_p, \mathbf{x}_{c_i}). \tag{3.12}$$

Now the search space size for each pixel reduces from $O(M^{K_{\max}})$ to $O(K_{\max})$, and we can solve the optimization problem (3.12) by linearly searching the number of reflections for $\mathbf{x}_p$ and $\{\mathbf{x}_{c_i}\}$. Importantly, the fact that this search couples a single projector pixel label with multiple camera pixel labels greatly enhances the robustness of the labeling procedure. Fig. 3.4 shows an example of epipolar labeling for one camera pixel, where our procedure finds the label that minimizes the virtual epipolar distance from the empty label.

3.2.3 Correctness of epipolar labeling

In Section 3.2.2, we described an efficient procedure for estimating labels for projector-camera pixel correspondences. We now analyze the correctness of this labeling procedure. Our analysis establishes two key facts: First, minimizing (3.12) determines labels up to a certain ambiguity. Second, resolving this ambiguity is possible by simply using the fact that the scanned object must lie in the physical space enclosed by the mirrors in front of the projector-camera system. Along the way, we derive conditions on the projector-camera-mirror geometry necessary for the correctness of our labeling procedure.

We define a *mirrored label* $M(\ell, m)$ of a label $\ell = (l_1, l_2, \ldots)$ as

$$M(\ell, m) \equiv (l_1, l_2, \ldots, m), \tag{3.13}$$

which is the label followed by an additional reflection by mirror m. We can generalize this to a multiply mirrored label, that is

$$M(\ell, \ell') \equiv (l_1, l_2, \ldots, l'_1, l'_2, \ldots), \tag{3.14}$$

This mirrored label is useful because all possible labels can be represented as mirrored labels of the true label. In particular, let the true label for $\mathbf{x}_p$ and $\mathbf{x}_{c_i}$ be ℓ^*_p and $\ell^*_{c_i}$, respectively, that is, $d(\ell^*_p, \ell^*_{c_i}; \mathbf{x}_p, \mathbf{x}_{c_i}) = 0$. Given that all possible labels are also pre-subsequence of the empty label, we can represent them as

$$\ell_p = M(\ell^*_p, \ell'_p), \quad \ell_c = M(\ell^*_{c_i}, \ell'_c), \tag{3.15}$$

where ℓ'_p and ℓ'_c are arbitrary labels contained in the empty label. We can now prove the following proposition.

Proposition 1 (Virtual epipolar distance of identically mirrored label.)**.** *The virtual epipolar distance between ℓ_p and ℓ_c is the same as that between $M(\ell_p, \ell')$ and $M(\ell_c, \ell')$ for any label ℓ'.*

Proof. From (3.3), we can write the mirror transformation matrix of the mirrored label $\mathbf{D}(M(\ell, \ell'))$ as

$$\mathbf{D}(M(\ell, \ell')) = \mathbf{D}(\ell)\mathbf{D}(\ell'). \tag{3.16}$$

Then, the relative transformation between identically mirrored virtual projector and virtual camera becomes

$$\mathbf{T}_p\mathbf{D}(M(\ell_p, \ell'))\mathbf{D}(M(\ell_{c_i}, \ell'))^{-1}\mathbf{T}_c^{-1} \tag{3.17}$$

$$=\mathbf{T}_p\mathbf{D}(\ell_p)\mathbf{D}(\ell')\mathbf{D}(\ell')^{-1}\mathbf{D}(\ell_{c_i})^{-1}\mathbf{T}_c^{-1} \tag{3.18}$$

$$=\mathbf{T}_p\mathbf{D}(\ell_p)\mathbf{D}(\ell_{c_i})^{-1}\mathbf{T}_c^{-1}. \tag{3.19}$$

Therefore, the effect of the additional reflection cancels out, and the relative transformation does not change by the identically mirrored label. Thus, the epipolar distance does not change either. □

Remark. Proposition 1 implies that, given the true projector-camera label pair ℓ_p^* and $\ell_{c_i}^*$, any mirrored label pair of the form $M(\ell_p^*, \ell')$ and $M(\ell_{c_i}^*, \ell')$ satisfies

$$d(M(\ell_p^*, \ell'), M(\ell_{c_i}^*, \ell'); \mathbf{x}_p, \mathbf{x}_{c_i}) = d(\ell_p^*, \ell_{c_i}^*; \mathbf{x}_p, \mathbf{x}_{c_i}) = 0. \tag{3.20}$$

Therefore, given the true label, mirroring both the projector and the camera pixel with the same sequence produces a valid solution. This raises the question: can there be other ambiguous solutions to the labeling problem? That is, could a differently mirrored label be used for the projector and camera and still satisfy the epipolar constraints? We eliminate this possibility next.

Proposition 2 (Virtual epipolar distance of differently mirrored label.)**.** *Given the true labels, ℓ_p^* and $\ell_{c_i}^*$, for the projector and camera pixels, the rays corresponding to the mirrored labels $M(\ell_p^*, \ell_p')$ and $M(\ell_{c_i}^*, \ell_c')$ never meet for $\ell_p' \neq \ell_c'$, i.e,*

$$d(M(\ell_p^*, \ell_p'), M(\ell_{c_i}^*, \ell_c'); \mathbf{x}_p, \mathbf{x}_{c_i}) > 0, \tag{3.21}$$

provided that the kaleidoscope and the projector-camera pair are in a generic configuration where the epipolar planes, both real and virtual, and mirror normals are not co-planar.

Proof. Our proof relies on the intuition that the probability of two arbitrary lines in 3D being co-planar is zero. We explain the proof when the mirrored label introduces a single additional bounce. Without loss of generality, we assume that the true 3D point is at the origin, and that the last two mirror bounces before hitting the object are at points $\mathbf{p}_1$ and $\mathbf{p}_2$, on mirrors 1 and 2, respectively. If we have the true labels, the rays for triangulation are $\mathbf{p}_1 + t_1(-\mathbf{p}_1)$ and $\mathbf{p}_2 + t_2(-\mathbf{p}_2)$ and intersect at the origin. If we have the wrong labels, which include additional bounces on mirrors 1 and 2, the rays are $\mathbf{p}_1 + t_1(\mathbf{I} - 2\mathbf{n}_1\mathbf{n}_1{}^\top)\mathbf{p}_1$ and $\mathbf{p}_2 + t_2(\mathbf{I} - 2\mathbf{n}_2\mathbf{n}_2{}^\top)\mathbf{p}_2$. The two rays are co-planar if and only if

$$\det\left(\begin{bmatrix}\mathbf{p}_1 - \mathbf{p}_2 & (\mathbf{I} - 2\mathbf{n}_1\mathbf{n}_1{}^\top)\mathbf{p}_1 & (\mathbf{I} - 2\mathbf{n}_2\mathbf{n}_2{}^\top)\mathbf{p}_2\end{bmatrix}\right) = 0, \tag{3.22}$$

which is possible only when $\mathbf{p}_1, \mathbf{p}_2, \mathbf{n}_1$, and $\mathbf{n}_2$ are co-planar. □

There exist degenerate cases when the conditions of Proposition 2 do not hold. One example is when mirrors are perfectly symmetric, as in Fig. 3.5, and consequently a reflected ray is on the same plane as the epipolar plane of the true label. However, placing the projector-camera pair in an asymmetric configuration relative to the mirrors avoids such degeneracies. Note that Lanman *et al.* [2009] used a system that is in a perfectly symmetric configuration by design. Their focus was on mitigating interference, with labeling done manually.

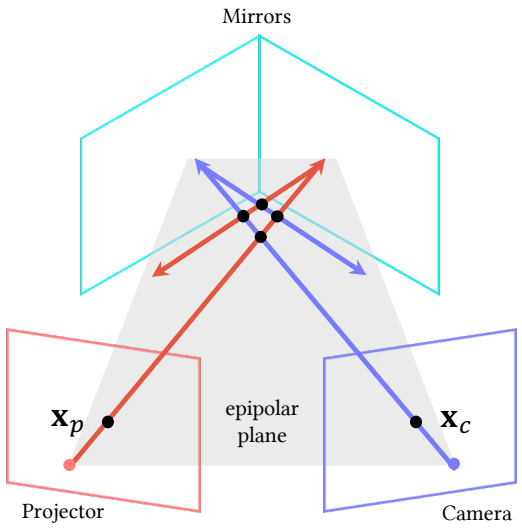


Figure 3.5. **Degenerate configuration.** When the mirrors and the rays from the projector and camera are perfectly symmetric, the reflected rays can lie on the epipolar plane of corresponding projector-camera pixels.

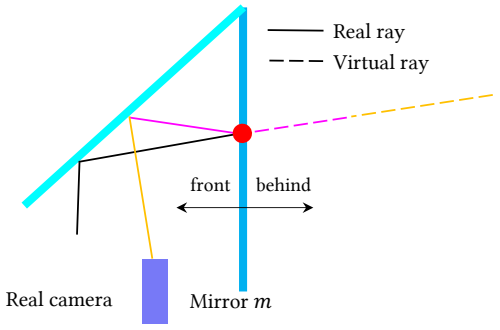


Figure 3.6. **Observation.** For a physically feasible traced ray involving mirror reflections, the virtual ray that is unfolded from the real ray before hitting mirror m is always behind the mirror m.

From Propositions 1 and 2, we know that minimizing (3.12) provides the true labels for projector and camera up to mirroring by a shared third label. We will resolve the remaining ambiguity with Proposition 3 below. We first introduce the following observation.

Observation. For a physically feasible traced ray involving mirror reflections, the virtual ray that is unfolded from the real ray before hitting mirror m is always behind the mirror m. Figure 3.6 shows an example: the virtual rays (dotted line segments) unfolded from the real rays (solid line segments) are

always behind the mirror m. This is true because the mirror reflection transforms a real or virtual point in front of the mirror to a virtual point behind the mirror.

With this observation in mind, we can now resolve the ambiguity due to identically mirrored labels.

Proposition 3 (Triangulation from identically mirrored labels.)**.** *The triangulated point from identically mirrored labels is always outside the mirror system.*

Proof. We denote by $\mathbf{X}^*$ the true 3D point triangulated from $\mathbf{x}_p$ and $\mathbf{x}_{c_i}$ given the true labels ℓ_p^* and $\ell_{c_i}^*$. When we have incorrect labels $M(\ell_p^*, \ell')$ and $M(\ell_{c_i}^*, \ell')$ that are identically mirrored with a label $\ell' = (l'_1, l'_2, ..., l'_{K'})$, the corresponding virtual projectors and cameras are transformed by the label ℓ' relative to the virtual projectors and cameras of the true labels. Therefore, triangulation reconstructs the transformed point $\mathbf{X}_t = \mathbf{D}(\ell')^{-1}\mathbf{X}^*$. We will show that $\mathbf{X}_t$ is always outside the mirror system, and specifically, behind the mirror $l'_{K'}$ corresponding to the last element of ℓ'.

We can prove this directly using the above observation because: 1) the empty label is from the object-free traced ray, which is physically feasible; 2) $\mathbf{X}^*$ is the true 3D point, which is on the real ray before hitting mirror $l'_{K'}$; and 3) $\mathbf{X}_t = \mathbf{D}(l'_{K'})^{-1} \cdots \mathbf{D}(l'_1)^{-1}\mathbf{X}^*$ corresponds to a point on the unfolded virtual ray from mirror $l'_{K'}$. Thus, $\mathbf{X}_t$ is behind the mirror $l'_{K'}$, and therefore outside the mirror system. □

In summary, Propositions 1–3 establish that we can determine the labels associated with a projector pixel and its corresponding camera pixels by searching over the empty labels associated with each pixel for the labels that have zero (or smallest) total virtual epipolar distance, and produce a triangulated point inside the mirror system. In the absence of noise, this procedure provably produces correct labels.

Remark. We note that, even though we only discuss *adding* labels to the true label, our derivation automatically covers *removing* labels as well. This is a consequence of the fact that the matrix $\mathbf{D}_m$, which describes the mirror transformation, is involutory. Hence, adding the trailing end of a mirror sequence in reverse is equivalent to deleting the trailing end from the sequence.

3.2.4 Comparison to other labeling methods

Visual hull. Reshetouski *et al.* [2011] propose solving the labeling problem using the visual hull, approximated through space carving from background pixels. For objects with simple shapes that are predominantly convex, the labeling from a crude visual hull is often a good approximation to the true labeling. However, for objects that have complex geometry and self-occlusions, the visual hull invariably fails to capture concavities, especially when there are not enough background pixels. Fig. 3.7 shows

an example of such a failure case. (For the visualization of labels, we sorted the mirror sequence in ascending order and used the "cool" MATLAB colormap—magenta-to-cyan linearly.) For such objects, the resulting labeling can significantly deviate from the correct labeling. The labeling accuracy for projector and camera is 76.13 % and 83.47 % for the visual hull method, and 99.96 % and 99.99 % for ours. The visual hull results also depend on the quality of background segmentation and the initial shape for carving. In the simulation for Fig. 3.7, we used the ground-truth background segmentation and set the initial shape to be a cube that is 20 % larger than the ground-truth shape in each axis. Ihrke *et al.* [2012] used a structured light system with a kaleidoscope as we do, but relied on the inaccurate labeling from the visual hull, resulting in artifacts as we discuss in Section 3.5.

Pulsed ToF. Xu *et al.* [2018] combined a kaleidoscope with a pulsed ToF camera, with the source and detector collocated. The ToF measurement provides a simple solution to the labeling problem: As the source and detector are collocated, simply ray tracing from the detector pixel for a distance equal to half the measured ToF provides the location of the 3D point. However, this technique requires a high-cost pulsed ToF camera for precise ToF measurements.

3.3 Surface Reconstruction

In the previous section, we have shown how to establish *labeled* correspondences $\mathbf{x}_p \leftrightarrow \{\mathbf{x}_{c_i}\}$ between a projector pixel $\mathbf{x}_p$ and multiple camera pixels $\{\mathbf{x}_{c_i}\}$. We now explain how to recover a 3D point $\mathbf{q}$ from a labeled correspondence, as well as how to reconstruct the surface of the scanned object from multiple such 3D points.

Multi-view triangulation. A naive approach for reconstructing 3D points $\mathbf{q}$ from a labeled correspondence $\mathbf{x}_p \leftrightarrow \{\mathbf{x}_{c_i}\}$ would be to apply the classical two-view triangulation procedure [Hartley and Zisserman, 2004] to each projector-camera pixel pair $\mathbf{x}_p \leftrightarrow \mathbf{x}_{c_i}$ separately. However, this approach does not take into account the information that each of these projector-camera pixel pairs is an observation of the same underlying 3D point $\mathbf{q}$. These multiple observations of the same point correspond to different viewpoints and baselines, and thus taking them into account jointly can greatly improve the robustness of the reconstruction of the unique 3D point $\mathbf{q}$. The naive approach produces multiple perturbed versions of this point, resulting in a noisy and redundant point cloud that impedes subsequent surface reconstruction procedures.

We adopt a *multi-view triangulation* approach, which uses all the geometric information from the one-to-multiple labeled correspondence $\mathbf{x}_p \leftrightarrow \{\mathbf{x}_{c_i}\}$, to reconstruct a single 3D point $\mathbf{q}$. We obtain this

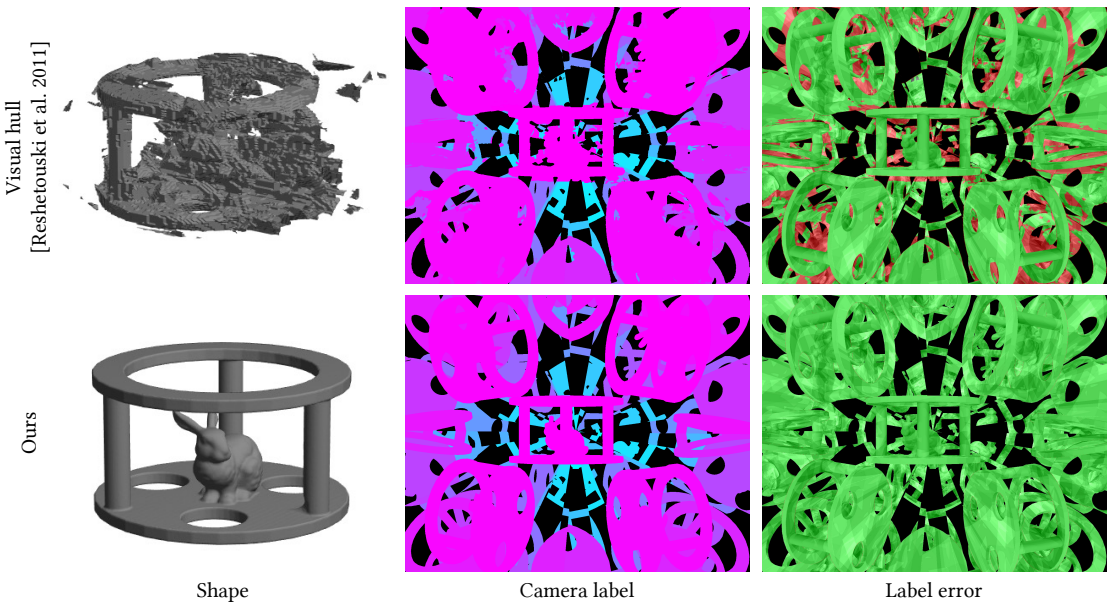


Figure 3.7. **Comparison to visual hull.** Reshetouski *et al.* [2011] solved the labeling problem using the visual hull. Their approach fails when there are insufficient background pixels, or for non-convex objects. Green and red colors in "label error" indicate correct and incorrect labels, respectively.

point by solving a linear least-squares problem:

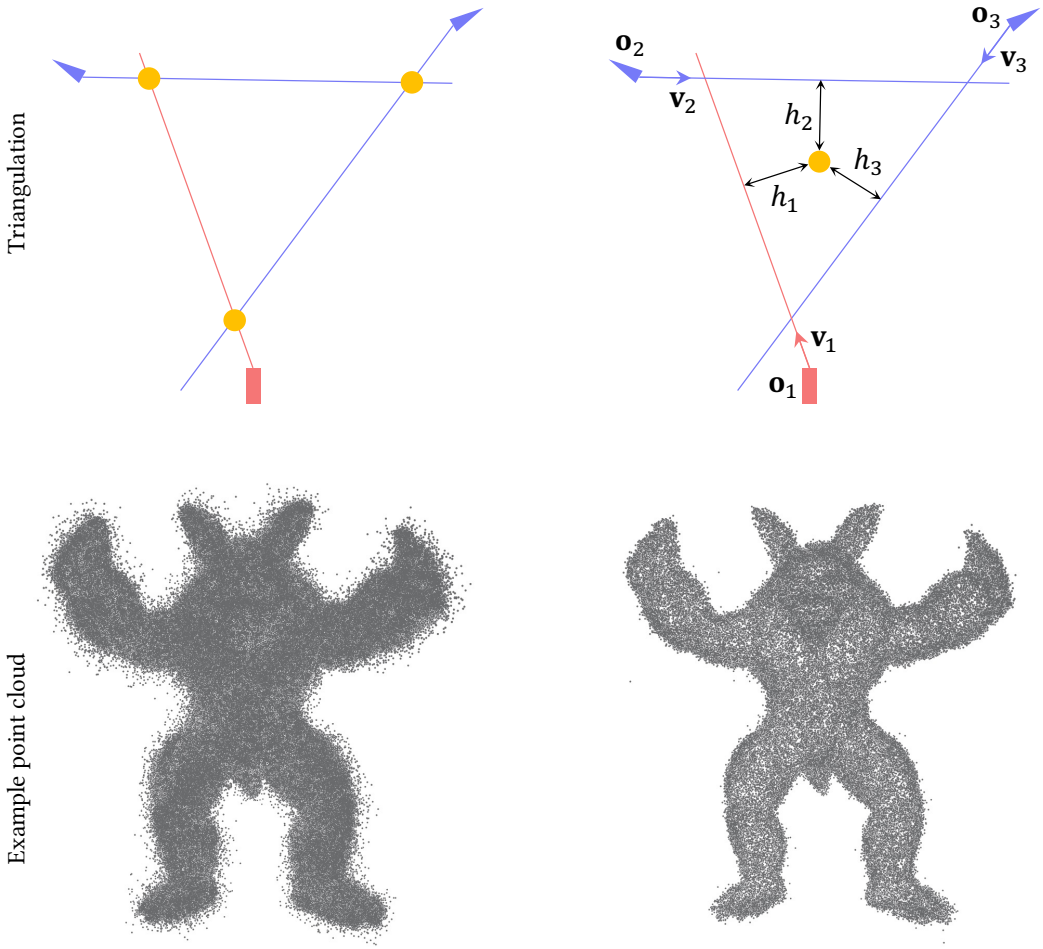
$$\mathbf{q} = \underset{\mathbf{q}}{\operatorname{argmin}} \sum_i h_i^2 \tag{3.23}$$

$$= \underset{\mathbf{q}}{\operatorname{argmin}} \sum_i \| \left(\mathbf{I} - \mathbf{v}_i \mathbf{v}_i{}^\top\right)(\mathbf{q} - \mathbf{o}_i) \|^2 \tag{3.24}$$

$$= \underset{\mathbf{q}}{\operatorname{argmin}}\, \mathbf{q}^\top \mathbf{A}\mathbf{q} - 2\mathbf{b}^\top \mathbf{q} + c \tag{3.25}$$

$$= \mathbf{A}^{-1}\mathbf{b}, \tag{3.26}$$

where h_i is the distance from the 3D point $\mathbf{q}$ to each ray, $\mathbf{o}_i$ is the ray origin, $\mathbf{v}_i$ is the ray direction, and we use $\mathbf{A} \equiv \sum_i \left(\mathbf{I} - \mathbf{v}_i \mathbf{v}_i{}^\top\right)$, $\mathbf{b} \equiv \sum_i \left(\mathbf{I} - \mathbf{v}_i \mathbf{v}_i{}^\top\right) \mathbf{o}_i$. and $\mathbf{c} \equiv \sum_i \mathbf{o}_i^\top \left(\mathbf{I} - \mathbf{v}_i \mathbf{v}_i{}^\top\right) \mathbf{o}_i$. Fig. 3.8 compares point clouds produced using the naive two-view and our multi-view triangulation procedures. For this comparison, we use simulated data where we perturb measurements of camera pixel locations with Gaussian noise of variance 5 pixels. We observe that the point cloud from multi-view triangulation is less noisy than that from two-view triangulation. We empirically found this to be the case across all our



(a) Two-view triangulation (b) Multi-view triangulation

Figure 3.8. **Triangulation.** Given the correspondence between a projector pixel and multiple camera pixels, we can perform either (a) two-view triangulation for all pairs in the correspondence; or (b) multi-view triangulation. Multi-view triangulation reconstructs a single 3D point that is closest to all the rays, making it more robust than two-view triangulation. Multi-view triangulation produces less noisy results than two-view triangulation when there is noise in the measurements.

experiments. Therefore, we use multi-view triangulation throughout this section.

Table 3.1. **Components used in our hardware prototype.**

Description	Company	Model name
laser projector	Sony	MP-CL1A
camera	FLIR	GS3-U3-91S6M
camera lens	Nikon	AF Nikkor 24mm f/2.8D
mirror	Edmund Optics	46-656 (custom)

Outlier rejection. The multi-view triangulation estimate of (3.26) is susceptible to outlier rays due to incorrect pixel detections (e.g., pixels illuminated due to direct illumination of the camera from a virtual projector, or indirect illumination effects). Such outliers can cause severe errors in the estimation of the unknown 3D point $\mathbf{q}$. We address this issue using RANSAC [Fischler and Bolles, 1981]: We repeatedly perform two-view triangulation between projector-camera pixel pairs $\mathbf{x}_p \leftrightarrow \mathbf{x}_{c_i}$ randomly selected from the labeled correspondence $\mathbf{x}_p \leftrightarrow \{\mathbf{x}_{c_i}\}$. For each reconstructed 3D point, we choose as inliers the pixels in $\{\mathbf{x}_{c_i}\}$ whose corresponding rays are close enough (0.5 mm) to the reconstructed point. Finally, we perform multi-view triangulation between $\mathbf{x}_p$ and the largest inlier set to get a robust estimate of the 3D point.

Surface reconstruction. Our triangulation procedure produces a 3D point for each labeled correspondence $\mathbf{x}_p \leftrightarrow \{\mathbf{x}_{c_i}\}$. The last step in our reconstruction pipeline is to use the resulting 3D point cloud to reconstruct the object surface. For this, we first compute PCA normals [Hoppe *et al.*, 1992] for each point in the point cloud. We then reconstruct a mesh representation of the scanned object by using screened Poisson surface reconstruction [Kazhdan and Hoppe, 2013], which estimates an implicit surface from the oriented point cloud, and extracts an isosurface.

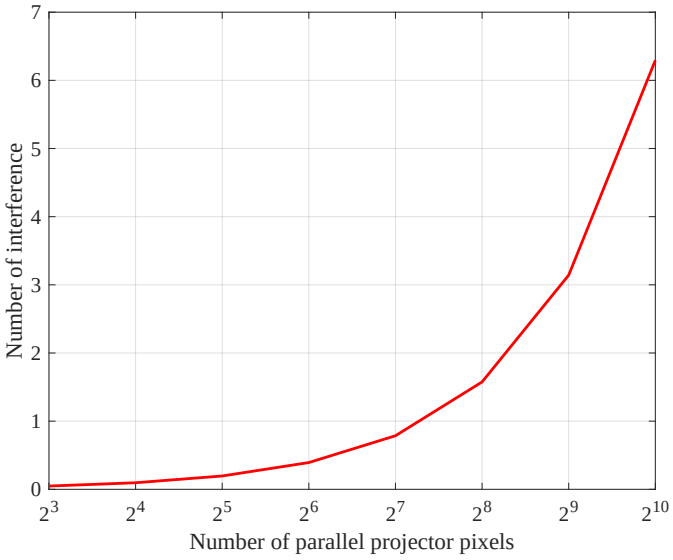
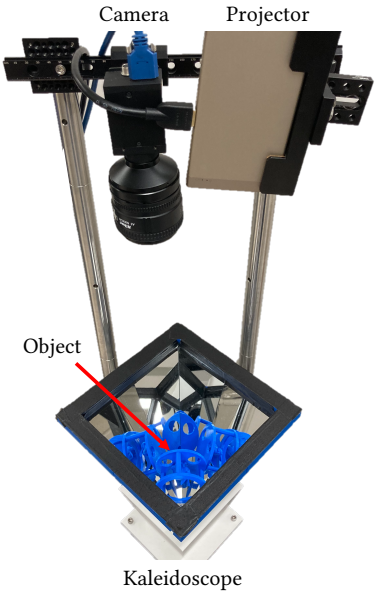


Figure 3.9. **Interference in parallel scanning of a spherical object.** The graph quantifies the trade-off between interference and acquisition speed, by plotting the number of points interfered by mirror reflection against the number of simultaneously illuminated projector pixels.

3.4 Implementation

We have developed a hardware prototype for a kaleidoscopic structured light system, comprising a laser projector, a monochrome CCD camera, and a kaleidoscope. The inset shows a photograph of our prototype. For the kaleidoscope, we use four planar metal-coated mirrors (surface flatness 4–6λ, dimensions 200 mm × 307 mm) that we cut to be shaped as isosceles triangles. Table 3.1 lists the exact parts used in our prototype. In the rest of this section, we describe how we calibrate our system, and how we accelerate scanning.



Parallel scanning. Up to this point, we have assumed that scanning works by sequentially illuminating all projector pixels, one at a time. However, sequential scanning makes acquisition times impractically long. We now describe a parallel scanning technique for faster acquisition.

Similar to traditional structured light techniques, our parallel scanning

technique operates by simultaneously illuminating multiple projector pixels with a temporal code, and decoding the measurements made at a pixel to obtain projector-camera correspondences. However, a key difference in the kaleidoscopic setting is the complexity of the epipolar geometry (and thus, interference patterns) produced by the multiple mirrors, which makes traditional column-scanning schemes inapplicable. Instead, we *randomly* sample groups of $2^8 - 1$ projector pixels, encode each pixel with an 8-bit binary code, and project the sequence of binary images (and their inverses for improved robustness) that correspond to each bit of the binary code. Then, we obtain the camera pixels corresponding to each projector pixel by decoding the binary code from the captured image.

This parallel scanning technique accelerates acquisition, but also exacerbates interference: multiple projector pixels may illuminate the same 3D point, which could result in erroneous decoding. The likelihood of this happening increases as we increase the number of projector pixels we simultaneously illuminate, which creates a trade-off between acquisition acceleration and interference.

To empirically quantify this trade-off, we simulated parallel scanning of a spherical object. Fig. 3.9 plots the number of points having interference in each scan, as a function of the number of simultaneously illuminated projector pixels. Based on this plot, we chose to illuminate $2^8 - 1$ projector pixels, corresponding to an average of one pixel with interference in each scan. This results in few decoding errors, which are easily handled during triangulation via RANSAC.

3.5 Results

We evaluate our method using simulated and real experiments. Our code and data are available on the project page [Ahn *et al.*, 2021b].

3.5.1 Simulated experiments

We use a ray tracing implementation (customized to handle multiple specular-specular reflections) to simulate measurements from our kaleidoscopic structured light system. We use these simulated measurements to evaluate the results of our labeling and shape reconstruction procedures against known ground truth. The diameter of our simulated objects is 60 mm.

Labeling accuracy. Fig. 3.10 and Table 3.2 show simulated labeling results for three synthetic objects, with and without Gaussian noise in the camera pixel measurements ($\sigma = 5$ px). Our labeling provides almost 100% accuracy in all cases.

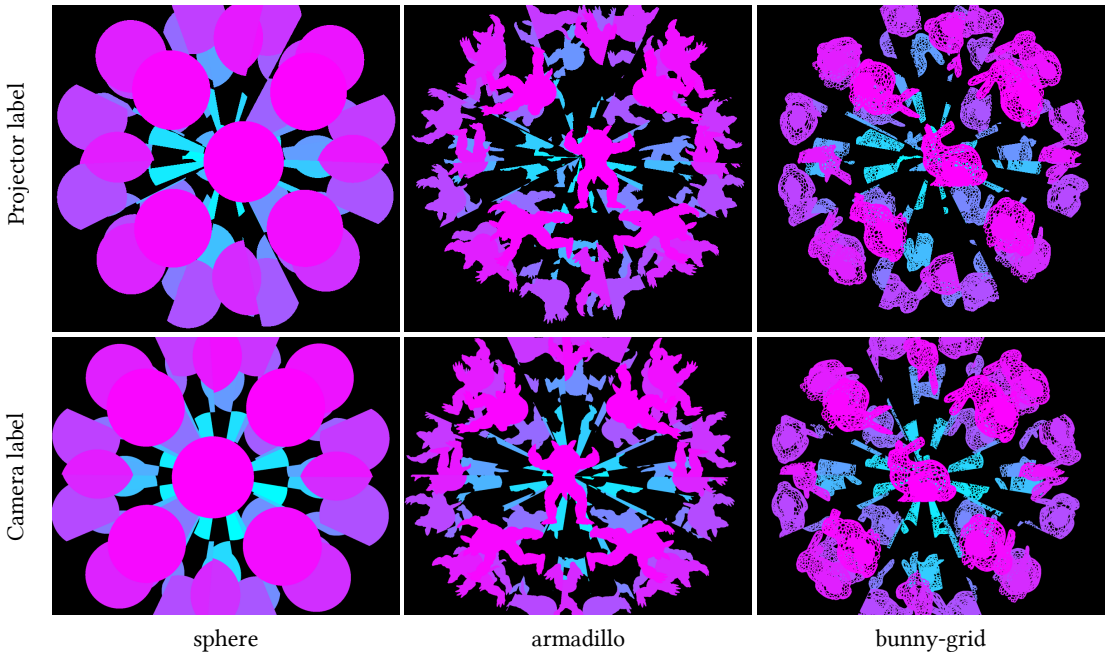


Figure 3.10. **Labeling accuracy for synthetic data.** We visualize the labeling results under Gaussian noise for several simulated objects.

Table 3.2. **Labeling accuracy statistics.** We report labeling accuracy metrics, with and without adding Gaussian noise ($\sigma = 5$ px) to camera pixel measurements. The labeling is robust to measurement noise.

Labeling accuracy	sphere	armadillo	bunny-grid
Projector (w/o noise)	100.00%	100.00%	100.00%
Projector (Gaussian noise)	99.69%	99.69%	99.43%
Camera (w/o noise)	100.00%	99.99%	100.00%
Camera (Gaussian noise)	99.99%	99.99%	99.98%

Reconstruction accuracy. Fig. 3.11 shows simulated shape reconstruction results, and average distance between reconstruction and ground truth. We note that the multiple virtual views of our system make it possible to reconstruct the severely-occluded inner bunny.

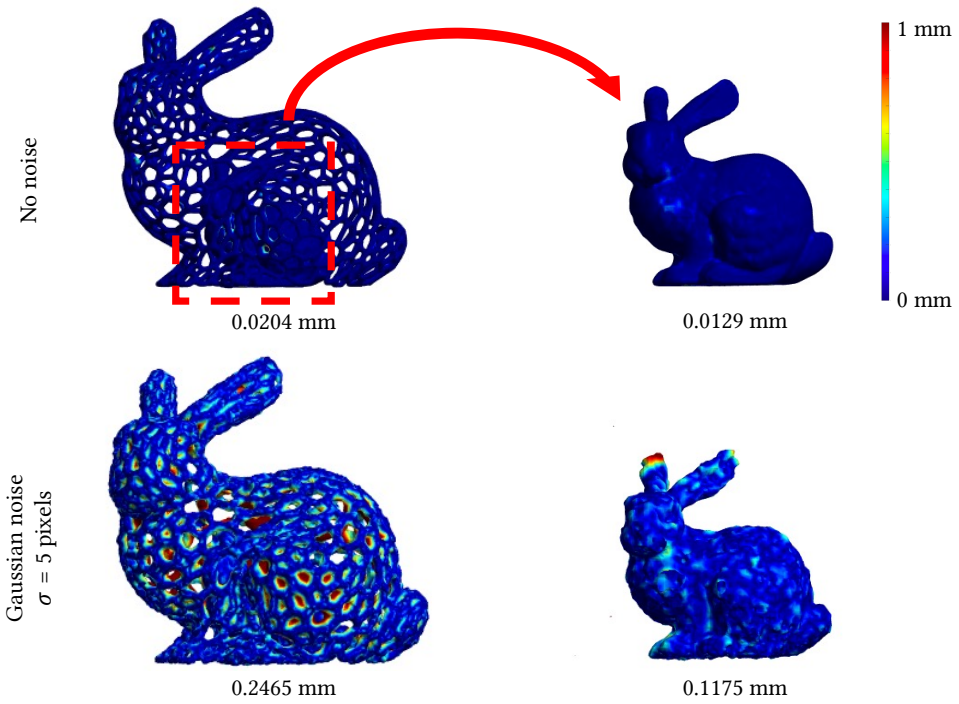


Figure 3.11. **Reconstruction accuracy for synthetic data.** We simulate scanning of an object with diameter 60 mm using the same imaging system as our prototype, with and without noise. Our system makes it possible to accurately reconstruct the severely occluded inner bunny.

Comparisons with other kaleidoscopic imaging methods. Fig. 3.12 shows simulated comparisons to other kaleidoscopic imaging methods. We used pixel binning and depth binning to simulate the finite resolution of the sensor used in each method. We compare the performance of different approaches using two metrics: The first metric is *accuracy*, which we define as the average distance of the vertices of the reconstructed mesh from the ground-truth mesh; this metric quantifies how the reconstruction is to the ground truth. The second metric is *coverage*, which we define as the average distance of the vertices of the ground-truth mesh from the nearest point of the reconstructed point cloud; this metric quantifies how well the reconstruction covers each part of the ground-truth shape.

- Fig. 3.12(a) shows results from the visual hull technique proposed by Reshetouski *et al.* [2011]. We observe that this technique does not recover the inner bunny, because of the limited background area

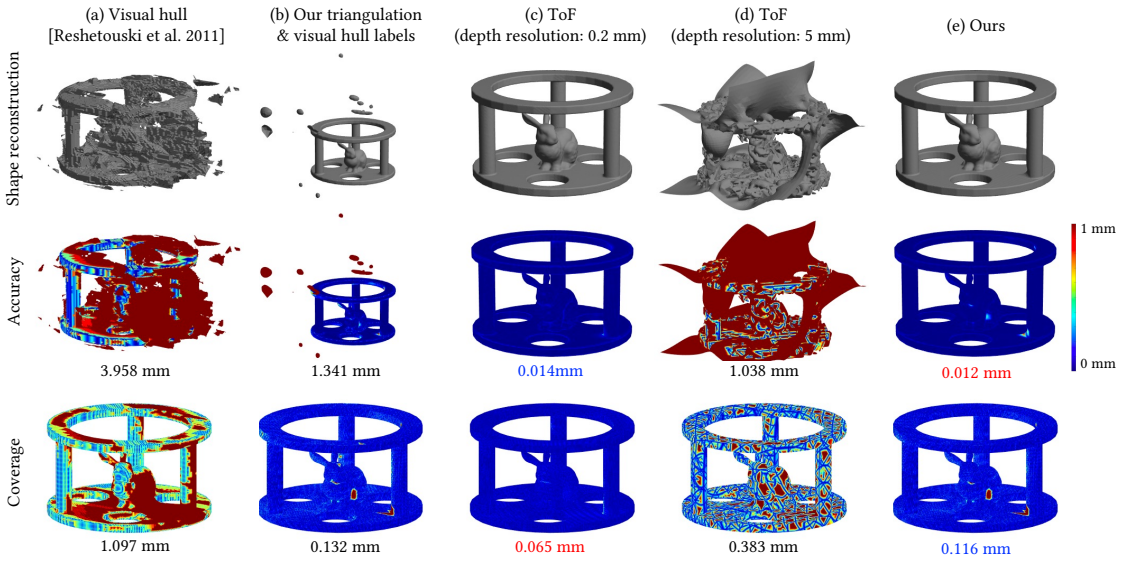


Figure 3.12. **Simulated comparison of kaleidoscopic imaging methods.** We use simulation to quantitatively evaluate the performance of various kaleidoscopic imaging methods. Our method provides the best reconstruction accuracy.

available for space carving.

- Fig. 3.12(b) shows results from a variation of our technique inspired by Ihrke *et al.* [2012], where we perform multi-view triangulation with RANSAC, but using labels from the visual hull (Fig. 3.7). We observe that, even though RANSAC mitigates the effect of incorrect labels, there are still some artifacts remaining.

- Fig. 3.12(c) and (d) show results from the ToF-based technique of Xu *et al.* [2018]. We set the spatial resolution of the ToF camera to be the same as that of our camera, and simulate two different ToF depth resolutions: Fig. 3.12(c) uses a depth resolution of 0.2 mm, corresponding to the calibration error reported by Xu *et al.* [2018] using a costly high-end lidar (Leica ScanStation P40 3D Laser Scanner). Fig 3.12(d) uses a depth resolution of 5 mm, corresponding to a low-end lidar of cost comparable to our setup (Intel Realsense LiDAR Camera L515). We observe that lowering the resolution results in a severe increase in noise in the reconstructed shape.

- Fig. 3.12(e) shows results from our method. We observe that, compared to the visual hull technique (a), our use of triangulation enables reconstructing the concave and occluded parts of the shape. Additionally, using accurate labels from our labeling technique reduces artifacts and improves accuracy

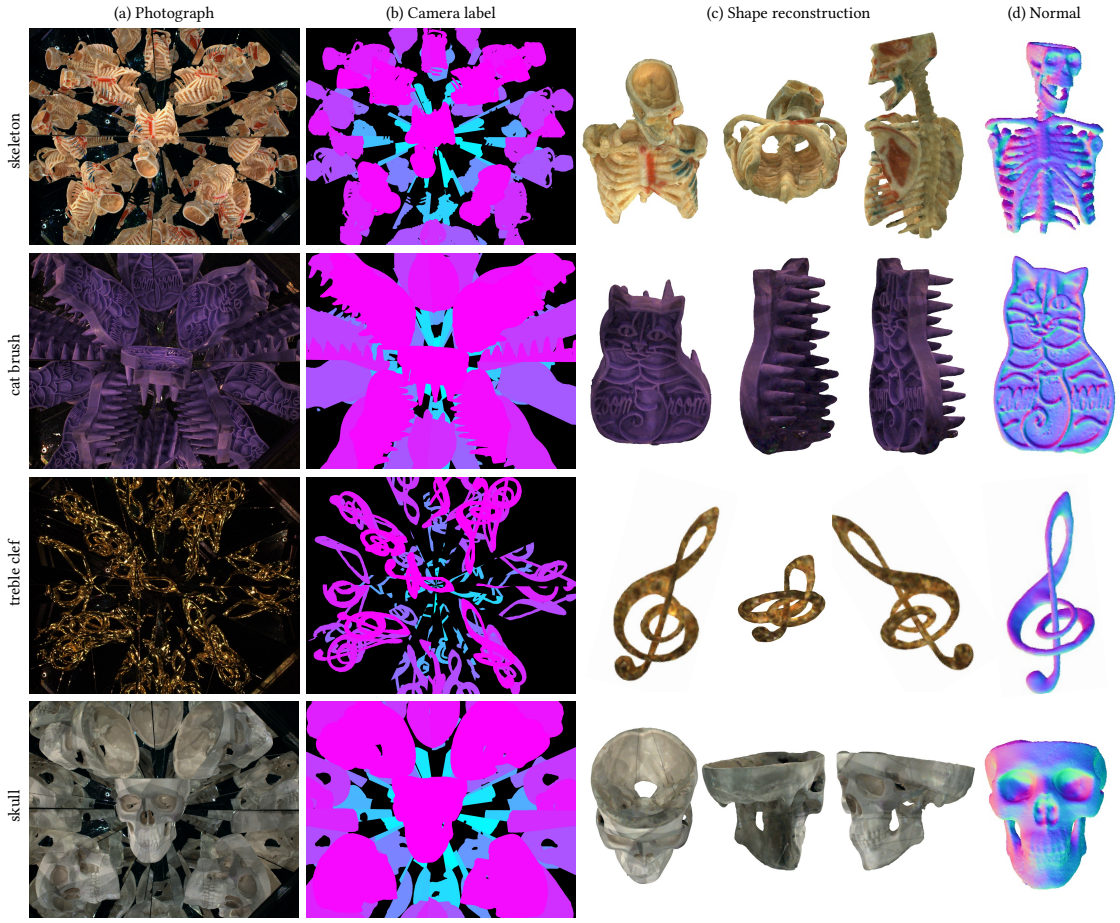


Figure 3.13. **Real object scans from our prototype.** Reconstructed point clouds and full-surround videos are available on the project webpage [Ahn *et al.*, 2021b].

by two orders of magnitude compared to using visual hull labels (b). Our technique has a similar accuracy improvement when compared to ToF using a lidar of cost comparable to our setup (d). Lastly, our technique achieves comparable accuracy with the ToF setup of Xu *et al.* [2018], while using a much more affordable imaging setup.

3.5.2 Real experiments

Scanned objects. Fig. 3.13 shows reconstructions of a variety of real objects obtained using the kaleidoscopic structured light prototype of Sec. 3.4. For each object, we show a kaleidoscopic image under

Table 3.3. **Effective number of per-vertex projector and camera views.** The number is generally smaller for larger objects because of occlusion.

#views	elephant	skeleton	cat brush	treble clef	skull
projector	5.2	6.3	4.8	9.1	4.7
camera	5.4	6.9	4.5	10.2	4.3
pair	31.8	51.5	24.3	96.5	22.2

uniform projector illumination, camera labels, and reconstructed mesh surface. Our setup allows scanning objects of size up to about 10 cm (e.g., the skull has dimensions 5 cm $\times$ 10 cm $\times$ 7.5 cm). To visualize the appearance of the reconstructed surface, we use a simple texture mapping procedure by projecting each vertex to all visible (virtual) cameras, and computing the average intensity of the corresponding pixel values. As our setup uses a monochrome camera, we obtain per-pixel color by projecting color channels sequentially from our RGB projector. We additionally visualize PCA vertex normals, to help better assess the quality of the reconstructed mesh. Fig. 3.1 shows an additional scanned object. We observe that our kaleidoscopic structured light system produces high-quality reconstructions for a variety of objects with complex visibility and reflectance properties.

Visibility. Fig. 3.14 and Table 3.3 report the effective average number of projector views, camera views, and unique projector-camera pairs per mesh vertex, for the scanned objects of Fig. 3.1 and Fig. 3.13. These numbers are strongly affected by the size and location of the object inside the kaleidoscope. For example, the skull has an average of 22.2 projector-camera pairs per vertex, whereas the smaller treble clef has an average of 96.5 pairs per vertex.

Quantitative evaluation. To quantify the reconstruction accuracy and coverage of our technique, we used our setup to scan a 3D-printed object for which a ground-truth mesh is available. The object had a width of 8 cm, and was 3D-printed at a layer resolution of 0.17 mm. Fig. 3.15 shows the results. We aligned the reconstructed mesh with the ground-truth one using the iterative closest point algorithm [Besl and McKay, 1992]. By comparing the two meshes, we estimated an accuracy of 0.235 mm and coverage of 0.305 mm for our method. By comparing these numbers with those in Fig. 3.12(e), where we used the same ground-truth mesh for a simulated experiment, we can also quantitatively assess the impact of calibration errors and other hardware imperfections on reconstruction quality.

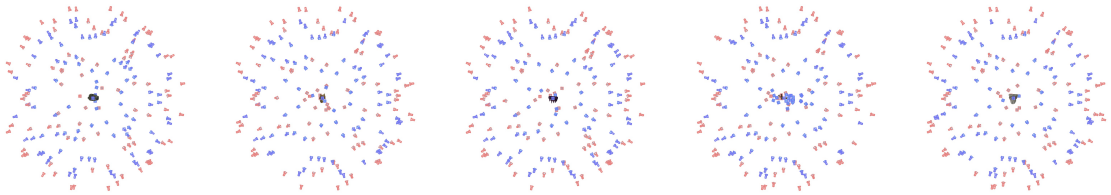
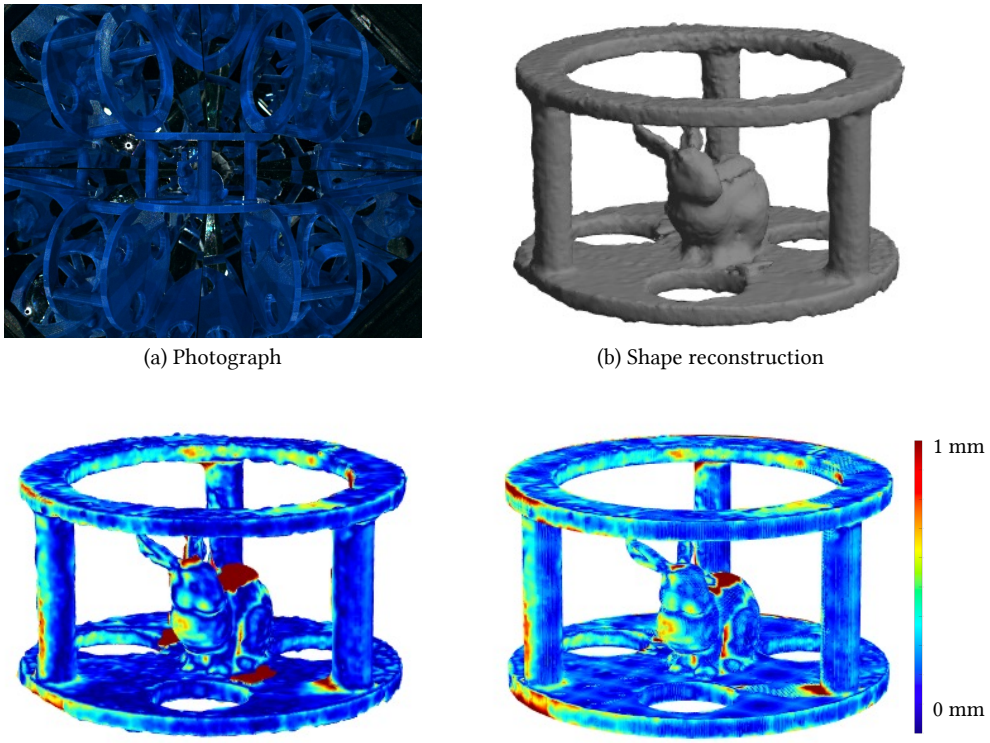


Figure 3.14. **Effective projector and camera views for different objects.** We visualize and report the number of the effective projectors and cameras around the object. Overall, there are almost 100 virtual projectors and cameras surrounding each object.

Comparison with kaleidoscopic ToF system. We performed an experiment where we replaced the projector-camera pair in our prototype with a commercial lidar of cost comparable to our setup (Intel Realsense LiDAR Camera L515, same depth resolution as that used for the simulated experiments of Fig 3.12(d)). We used this modified system to scan the same spherical calibration object as in Fig. 2.6. We show the results in Fig. 3.16. We note that, because of the low depth resolution and noisy ToF measurements of our lidar, we were unable to obtain accurate calibration information for the modified imaging setup using the calibration procedure of Xu *et al.* [2018]. In turn, the lack of accurate calibration meant we could not produce a meaningful shape reconstruction using their proposed ray folding procedure. We show, instead, our reconstructed *unfolded* point cloud, which we observe to be a lot noisier than the one obtained by our technique (Fig. 2.6). As Xu *et al.* [2018] and our simulations in Fig. 3.12 both have shown, using a high-end lidar can alleviate these issues and achieve the same reconstruction accuracy as our structured light technique, albeit at a much higher hardware cost.

3.6 Discussion

PCA normals. The combined use of our labeled correspondences and multi-view triangulation with RANSAC robustly reconstructs accurate point clouds. However, the screened Poisson surface reconstruction [Kazhdan and Hoppe, 2013] algorithm we use to create the final mesh reconstructions requires as input *oriented* point clouds. We produce those by assigning PCA normals [Hoppe *et al.*, 1992] to our reconstructed unoriented point clouds. Unfortunately, PCA normals can be inaccurate for shapes with complex topology, resulting in inaccurate meshes. As an example, in Fig. 3.17 we show scan results for a slinky: The reconstructed point cloud accurately represents the object's interwoven thin parts; however, the reconstructed mesh has strong artifacts near the center, because of the inaccurate PCA



(a) Photograph (b) Shape reconstruction

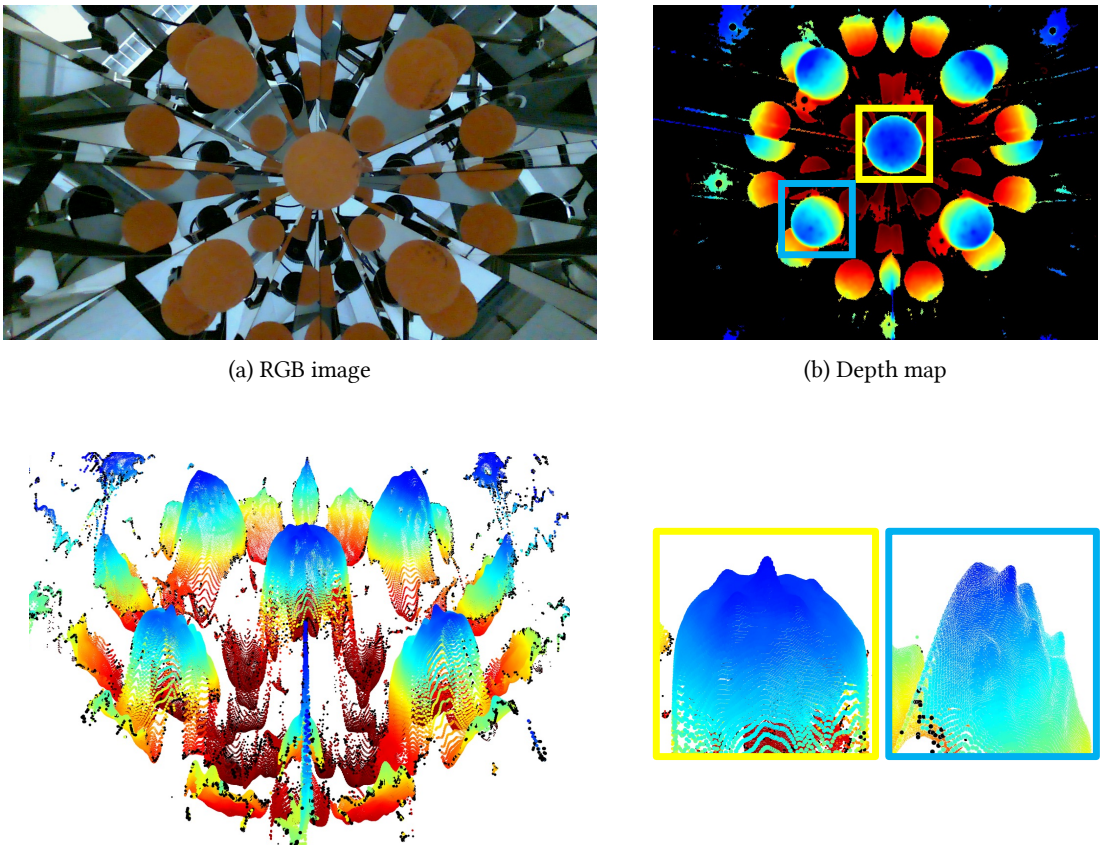
(c) Accuracy: 0.235 mm (d) Coverage: 0.305 mm

Figure 3.15. **Real scan of a 3D-printed mesh.** We 3D-printed and scanned the mesh used in the simulation results of Fig. 3.12. We report performance metrics comparing the reconstructed and ground-truth meshes.

normals. Combining our technique with more accurate normal estimation procedures, including ones using shading information, can help improve the accuracy of the final mesh reconstructions.

Pose of the object in the kaleidoscope. Fig. 3.18 shows results from scans of the same object from two different poses: in pose 1, we placed the object directly on the kaleidoscope, whereas in pose 2, we hung it using strings. In the former case, the parts on the mirrors (e.g., head, tail) are visible from few viewpoints, and thus are not reconstructed. By contrast, in the latter case, these parts are well reconstructed. This example highlights the strong impact object pose can have on the final reconstruction, and suggests object pose optimization as an important future research direction.

Comparison to neural rendering. 3D reconstruction by moving a flash-camera pair around an object, in the style of IDR [Yariv *et al.*, 2020] and NeRF [Mildenhall *et al.*, 2020], has recently seen immense



(a) RGB image

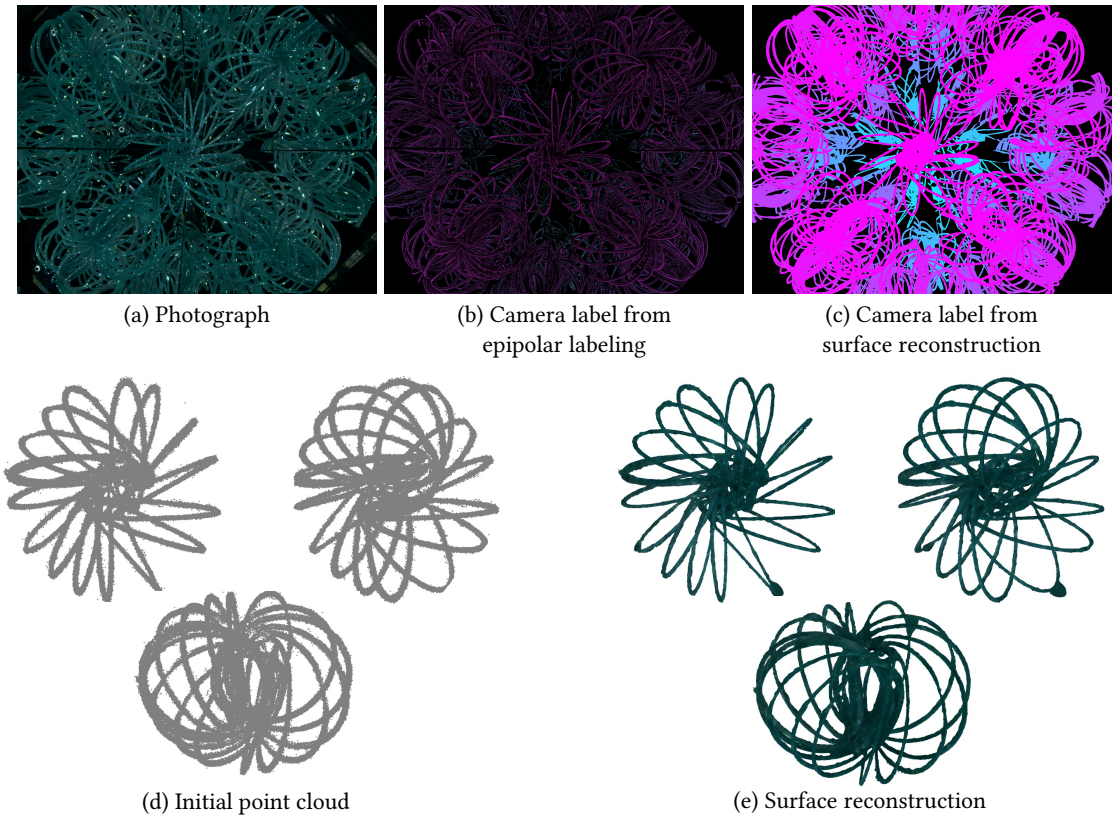
(b) Depth map

(c) Point cloud (unfolded)

(d) Zoom in (noisy surface)

Figure 3.16. **Kaleidoscopic time-of-flight experiment.** Unfolded point cloud obtained from the depth map of a commercial low-resolution lidar. The depth measurements are a lot noisier than those obtained using our kaleidoscopic structured light system for the same object (Fig. 2.6).

success. This is advantageous over our technique in terms of cost, object size, acquisition time, and overall user convenience. By contrast, our structured light technique can handle textureless objects where passive techniques fail. Our technique can also handle very complex shapes where, due to visibility, flash photography methods can produce poor results, unless the number of views becomes impractically large. In the future, it is possible that using neural rendering algorithms to process measurements from kaleidoscopic structured light setups can lead to scanning technologies that combine the complementary advantages of the two approaches.



(a) Photograph (b) Camera label from epipolar labeling (c) Camera label from surface reconstruction

(d) Initial point cloud (e) Surface reconstruction

Figure 3.17. **Effect of PCA normals.** We show an example of how of failures in normal estimation impact reconstruction quality: Our system estimates an accurate point cloud of a very thin object, for which the PCA normal estimates are inaccurate. As a result, the reconstructed mesh has strong artifacts, especially at the center where different rings intersect.

Summary. We introduced a full-surround 3D imaging technique that combines a projector-camera pair with a kaleidoscope, to produce a virtual multi-view structured light system. We derived an algorithm that uses the epipolar geometry between virtual projectors and virtual cameras to produce provably correct labels for pixels in the kaleidoscopic image, in terms of which virtual projector and virtual camera these pixels correspond to. By combining these labels with multi-view triangulation, we showed that our system can achieve high reconstruction accuracy and full coverage, even when scanning objects with complex geometry and reflectance.

(a) Camera view (b) Reconstructed point cloud

Figure 3.18. **Effect of object pose inside the kaleidoscope.** Changing the object pose impacts reconstruction quality. Having enough space between the object and mirrors can reduce occlusion and improve the reconstruction.

4 Kaleidoscopic Neural Rendering

In this chapter, we introduce a technique for full-surround 3D reconstruction with a single kaleidoscopic image in the framework of inverse rendering using the neural surface representation. Our key insight is that a single pixel in a kaleidoscopic image is equivalent to multiple such pixels in its multi-camera counterpart. For example, the pre-image of the background pixel, which is the collection of 3D points that map to that pixel, in a kaleidoscope does not intersect with the object; this implies that is also a background pixel in all virtual views associated with it. Similarly, even a foreground pixel that intersects with the object can be used to carve out space since all of the light path prior to a ray's intersection with the object can be considered as background.

Armed with these insights on the nature of information encoded in a kaleidoscopic image, we propose a technique that we call *kaleidoscopic space sculpting*; sculpting sets up an optimization problem that updates a neural implicit surface [Yariv *et al.*, 2020] using a collection of cost functions that encode background information (to remove regions) and foreground regions (to add regions), as well as the texture of the object. Interestingly, our technique does not explicitly calculate the label information. Despite this, it provides robust single-shot full-surround 3D reconstructions. For dynamic objects, we apply our technique separately on each frame of a kaleidoscopic video, to obtain full-surround 3D videos. Figure 4.1 shows a gallery of objects placed beside their 3D printed counterparts, obtained using neural kaleidoscopic space sculpting.

4.1 Related Work

Neural rendering. The proposed technique relies on recent advances in neural rendering and representations, which have found immense success in shape estimation [Yariv *et al.*, 2020] and novel viewpoint synthesis [Barron *et al.*, 2021; Mildenhall *et al.*, 2020; Verbin *et al.*, 2022]), especially for scenes that are largely static. There have been some recent works towards extending these techniques for

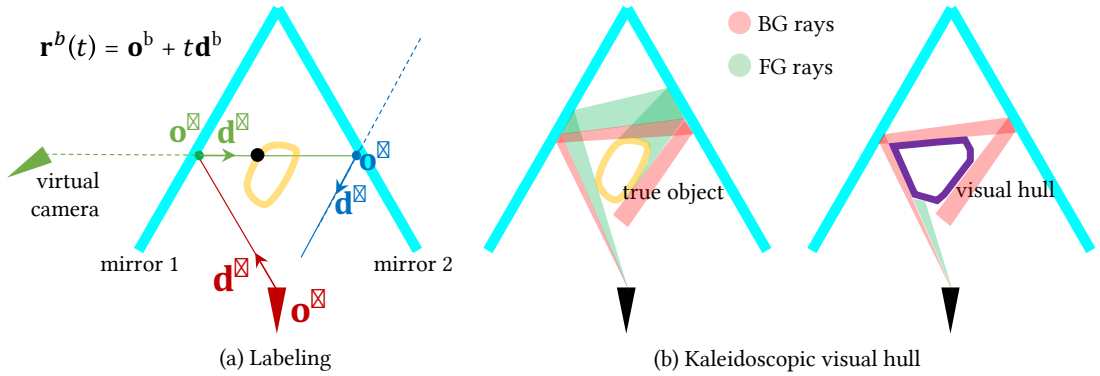
Figure 4.1. **3D printing of shape reconstructions.** The proposed neural kaleidoscopic space sculpting can generate replicas of real objects with a range of shapes and reflectances.

dynamic scenes [Park *et al.*, 2021a,b; Pumarola *et al.*, 2021]. The most successful among these work use class-specific priors for faces and the human body [Liu *et al.*, 2021; Peng *et al.*, 2021a,b; Wang *et al.*, 2021; Xu *et al.*, 2021]. In contrast, our technique provides a low-cost (vs. multi-view) solution for *general* objects (non-class specific). As our technique can reconstruct the scene frame by frame from a video, it can handle topology and color changes that cannot be modeled as rigid deformations.

4.2 Overview

Our problem formulation is as follows: given a kaleidoscopic image $\mathbf{I}$ and its silhouette mask $\mathbf{M}$, we seek to reconstruct a 3D surface that is photo- and silhouette-consistent. There are many solutions to the multi-view version of this problem, where the input is multiple images instead of a single kaleidoscopic image, including neural-rendering approaches [Mildenhall *et al.*, 2020; Yariv *et al.*, 2020] that have recently found great success. However, multi-view approaches do not directly apply to our problem, because the kaleidoscopic image has an additional challenge—the *labeling* problem. We discuss labeling and other concepts related to our approach.

Labels. We define a label as the sequence of mirrors that the ray backprojected from a pixel encounters before intersecting the object [Ahn *et al.*, 2021a; Reshetouski *et al.*, 2011]. With the label information, and



(a) Labeling (b) Kaleidoscopic visual hull

Figure 4.2. **Background.** (a) Labeling is needed to utilize a kaleidoscopic imaging system as a virtual multi-view imaging system. (b) The kaleidoscopic visual hull [Reshetouski *et al.*, 2011] has the same silhouette as the true object but may have different labels because of the lack of label information in the silhouette.

given the mirror geometry, we can decompose the kaleidoscopic image into the corresponding virtual camera images, by unfolding backprojected rays and assign the virtual viewpoints for each pixel. In particular, given the mirror geometry, we can determine the label from the number of bounces b for each pixel $\mathbf{x}$. As the label of an empty kaleidoscope (i.e., without object) can be precomputed, $b \in \{0..B_{\mathbf{x}}^{\emptyset}\}$ can determine the label and thereby the pose of the virtual camera, where $B_{\mathbf{x}}^{\emptyset}$ is the length of the empty label [Ahn *et al.*, 2021a]. Figure 4.2(a) shows a labeling example where the empty label $(1, 2)$ is the sequence of mirror numbers. If we can figure out the number of bounces before hitting the object being $b = 1 \in \{0, 1, 2\}$ ($B_{\mathbf{x}}^{\emptyset} = 2$), then the resulting label becomes (1), implying that this ray is equivalent to the ray from the virtual camera generated by the label (1). Thus, labeling allows kaleidoscopic imaging to serve as a virtual multi-view imaging system and to reconstruct the object geometry via multi-view stereo (MVS). However, it is challenging to obtain the label since it itself is a function of the unknown object shape.

Kaleidoscopic visual hull. Visual hull [Kutulakos and Seitz, 2000; Laurentini, 1994], initially used for multi-view images, is obtained by "carving" the voxels that are not consistent with the silhouette of the input images. A visual hull can provide a good approximation of 3D shape, if we have ample multi-view images and if the object under consideration is not overly complex.

Visual hulls have been utilized for kaleidoscopic images [Reshetouski *et al.*, 2011] where the background rays are reflected multiple times. To be specific, this method backprojects a ray from a pixel $\mathbf{x}$ and reflects it with mirrors, and computes the rays $\mathbf{r}_{\mathbf{x}}^b(t) = \mathbf{o}_{\mathbf{x}}^b + t\mathbf{d}_{\mathbf{x}}^b$ for each bounce b. $\mathbf{o}_{\mathbf{x}}^0$ is the real camera

location, $\mathbf{d}_{\mathbf{x}}^0$ is the backprojected direction, $\mathbf{o}_{\mathbf{x}}^{b+1}$ is the intersection point between $\mathbf{r}_{\mathbf{x}}^b(t)$ and the mirrors, and $\mathbf{d}_{\mathbf{x}}^{b+1}$ is the reflected direction. Once the visual hull is obtained, the label can also be obtained simply by ray tracing to the visual hull inside the kaleidoscope. The key idea underlying this approach is that it requires only background pixels, which are easy to label as the background labels are independent of the object shape.

A useful property of the visual hull is that it is the maximal shape among silhouette-consistent shapes [Kutulakos and Seitz, 2000]. Based on this property, the possible labels can be narrowed down by ignoring the bounces that do not intersect with the visual hull [Reshetouski *et al.*, 2011]. To be specific, for a pixel $\mathbf{x}$, if we define the set of bounces hitting the visual hull as

$$\mathcal{H}_{\mathbf{x}}^{\text{vh}} = \{b \mid \mathbf{r}_{\mathbf{x}}^b(t) \text{ intersects with the visual hull}\}, \tag{4.1}$$

then the true bounce should satisfy $b^{\text{true}} \in \mathcal{H}_{\mathbf{x}}^{\text{vh}}$. Furthermore, when $|\mathcal{H}_{\mathbf{x}}^{\text{vh}}| = 1$, implying there is only one bounce intersecting with the visual hull, the label of the pixel can be uniquely determined, which is refered to as *a reliable pixel*.

However, there is ambiguity in determining the equivalence of shapes under silhouette consistency, as the visual hull cannot resolve these inherent ambiguities in silhouette information. First, it cannot reconstruct concave objects, as concavity cannot be captured by silhouette. Second, kaleidoscopic visual hull cannot disambiguate shapes with the same silhouette but different label maps. We refer to this effect as *virtual occlusion*, as it is caused by the visual hull not being carved when the background is occluded by a virtual object. The ambiguity causes errors in decomposing the images into multi-view images, which results in an incorrect shape (Figure 4.2(b)). To resolve these ambiguities and obtain better labels, it is essential to use texture information from foreground pixels, which requires labeling, setting up a chicken-and-egg problem.

Neural surface representation. We represent the surface using neural signed distance function (SDF) [Park *et al.*, 2019; Yariv *et al.*, 2020]. In particular, we optimize for the surface using IDR [Yariv *et al.*, 2020], which we briefly review here. In IDR, shape and texture are represented by two separate neural networks θ and ϕ, which are updated to achieve a silhouette-consistent and photo-consistent shape based on input mask images $\{\mathbf{M}\}$ and RGB images $\{\mathbf{I}\}$. To update these networks, IDR backprojects a ray $\mathbf{r}_{\mathbf{x}}(t)$ from a pixel $\mathbf{x}$, checks if it intersects with the current shape, and computes a rendered mask $\widehat{\mathbf{M}}(\mathbf{x}, \theta)$. The intersection is computed using the sphere tracing algorithm [Hart, 1996] since the surface is represented as SDF $f(\cdot; \theta)$. If the ray hits the surface (i.e., $\widehat{\mathbf{M}}(\mathbf{x}) = 1$), it computes an RGB value $\widehat{\mathbf{I}}(\mathbf{x}; \theta, \phi)$ by giving the intersection point, surface normal, and view direction to the texture network. To be specific,

it computes the point along the ray that has the minimum SDF, which is represented by

$$\mathbf{p}_{\mathbf{x}}(\theta) = \text{argmin}_{\mathbf{p}\in\mathbf{r}_{\mathbf{x}}(t)} f(\mathbf{p};\theta). \tag{4.2}$$

IDR updates the shape and texture with the following loss:

$$\text{loss}(\theta,\phi) = \text{loss}_{\text{tex}}(\theta,\phi) + \text{loss}_{\text{mask}}(\theta) + \text{loss}_{\text{eik}}(\theta). \tag{4.3}$$

The texture loss $\text{loss}_{\text{tex}}(\theta,\phi)$ compares the rendered image and the input image as

$$\text{loss}_{\text{tex}}(\theta,\phi) = {}^{\lambda_{\text{tex}}}/_{|\mathcal{X}|} \sum_{\mathbf{x}\in\mathcal{X}_{\text{in}}} |\mathbf{I}(\mathbf{x}) - \widehat{\mathbf{I}}(\mathbf{x};\theta,\phi)|, \tag{4.4}$$

where λ_{tex} is the weight for texture loss, $\mathcal{X}$ is a set of pixels in the image, and $\mathcal{X}_{\text{in}} = \left\{\mathbf{x} \mid \mathbf{M}(\mathbf{x}) = 1 \text{ and } \widehat{\mathbf{M}}(\mathbf{x}) = 1\right\}$. The mask loss $\text{loss}_{\text{mask}}(\theta)$ compares the rendered mask and the input mask as

$$\text{loss}_{\text{mask}}(\theta) = {}^{\lambda_{\text{mask}}}/_{|\mathcal{X}|} \sum_{\mathbf{x}\in\mathcal{X}_{\text{out}}} \text{BCE}\left(m(\mathbf{p}_{\mathbf{x}}(\theta)), \mathbf{M}(\mathbf{x})\right), \tag{4.5}$$

where λ_{mask} is the weight for mask loss, BCE is the binary cross-entropy function, $m(\cdot)$ is the soft binarization function for SDF defined as

$$m(\mathbf{p}) = \text{sigmoid}\left(-\alpha f(\mathbf{p})\right), \tag{4.6}$$

to make the loss differentiable, α is a parameter that controls the softness, and $\mathcal{X}_{\text{out}} = \mathcal{X} - \mathcal{X}_{\text{in}}$. The eikonal loss $\text{loss}_{\text{eik}}(\theta)$ enforces the gradient of the SDF to have a unit norm as

$$\text{loss}_{\text{eik}}(\theta) = \lambda_{\text{eik}} \mathbb{E}\left(\|\nabla_{\mathbf{p}} f(\mathbf{p};\theta)\| - 1\right)^2, \tag{4.7}$$

where λ_{eik} is the weight for eikonal loss, and $\mathbf{p}$ is uniformly distributed in a bounding box of the scene.

4.3 Method

Our goal is to jointly solve the labeling and geometry reconstruction from a kaleidoscopic image. To solve this problem, we propose a method that we call *kaleidoscopic space sculpting*, which updates the shape both in additive and subtractive ways without the necessity of labels.

Our method uses two observations, which follow from the definitions of background and foreground pixels.

Observation 1 (Background rays)**.** *Background rays do not intersect with the object for all bounces, that is, for a background pixel* $\mathbf{x}$*,*

$$\forall b\, \forall t,\ f(\mathbf{r}_{\mathbf{x}}^{b}(t)) > 0, \tag{4.8}$$

where f *is the object SDF, and* $\mathbf{r}_{\mathbf{x}}^{b}(t)$ *is the pre-image of the pixel including mirror reflections.*

Observation 2 (Foreground rays)**.** *Foreground rays intersect with the object for at least one bounce, that is, for a foreground pixel* $\mathbf{x}$*,*

$$\exists b,\ \min_t |f(\mathbf{r}_{\mathbf{x}}^b(t))| = 0, \tag{4.9}$$

where $\mathbf{r}_{\mathbf{x}}^b(t)$ *is a ray bounced* b *times in a kaleidoscope after being backprojected from pixel* $\mathbf{x}$*, and* f *is the object SDF.*

From these observations, we can update the 3D shape by *carving* the points on the background rays, as they will never intersect the object; and *modeling* a point on the foreground rays, as there will be at least one intersection with the object. To do so, we need to answer the following questions: (1) How do we select points on the rays to carve and model? (2) how do we represent the shape and update it?

Point selection and labeling. The point selection for carving is straightforward, as all bounces can be used for carving in background rays. However, this is problematic for foreground rays, as we do not know which foreground ray bounce $\mathbf{r}_{\mathbf{x}}^b(t)$ will intersect with the object among all the possible bounces $b \in \{0..B_{\mathbf{x}}^{\emptyset}\}$. We need to select a point on the ray with the correct bounce b to avoid erroneous solutions. We can expect the current shape to be close to the true shape as the iteration proceeds, especially since an initial shape estimates can be determined by the background rays. Thus, we pick the foreground bounce with the minimum distance value in each iteration for selecting the modeling points. This turns out to be a simple yet effective approach. We now proceed to introduce how to update the shape with the silhouette constraint and texture constraint.

4.3.1 Silhouette constraint

To obtain a silhouette-consistent shape from a kaleidoscopic silhouette mask $\mathbf{M}$, we use the following loss function to update the neural SDF $f(\cdot;\theta)$ inspired by the IDR [Yariv *et al.*, 2020]:

$$\begin{aligned}\text{loss}_{\text{sil}}(\theta) =& \text{loss}_{\text{carve}}(\theta; \mathcal{P}_{\text{carve}}(\theta)) + \\ & \text{loss}_{\text{model}}(\theta; \mathcal{P}_{\text{model}}(\theta)) + \text{loss}_{\text{eik}}(\theta).\end{aligned} \tag{4.10}$$

$\mathcal{P}_{\text{carve}}(\theta)$ is the set of points for carving, $\mathcal{P}_{\text{model}}(\theta)$ is the set of points for modeling, $\text{loss}_{\text{carve}}(\theta)$ is the *carving loss*,

$$\text{loss}_{\text{carve}}(\theta) \equiv {}^{\lambda_{\text{carve}}}/_{|\mathcal{X}|} \sum\nolimits_{\mathbf{p}\in\mathcal{P}_{\text{carve}}(\theta)} \text{BCE}(m(\mathbf{p}), 0), \tag{4.11}$$

and $\text{loss}_{\text{model}}(\theta)$ is the *modeling loss*,

$$\text{loss}_{\text{model}}(\theta) \equiv {}^{\lambda_{\text{model}}}/_{|\mathcal{X}|} \sum\nolimits_{\mathbf{p}\in\mathcal{P}_{\text{model}}(\theta)} \text{BCE}(m(\mathbf{p}), 1), \tag{4.12}$$

where $m(\mathbf{p})$ is the soft binarization function defined in Equation (4.6), and λ_{carve} and λ_{model} are the weights for the carving loss and modeling loss, respectively. The difference in our method compared to IDR is that we use different point selection methods because rays are multiply reflected in the kaleidoscope with unknown labels. Next, we describe the point selection methods in detail for carving and modeling. Figure 4.3 illustrates the point selection methods.

Carving. Selecting the points for the carving $\mathcal{P}_{\text{carve}}(\theta)$ from background rays is straightforward as all the points $\mathbf{p}$ on the background rays should have positive SDF values (i.e., Observation 1) and thereby $m(\mathbf{p})$ should be zero. However, instead of using all the points on the ray (e.g., stratified sampling), we use a single point per each ray that has the minimum SDF value following IDR, which is more memory efficient and provides promising results. For a background pixel $\mathbf{x}$, these points can be expressed as

$$\mathcal{P}_{\mathbf{x}}^{\text{bg}}(\theta) = \left\{ \mathbf{p} \mid \underset{\mathbf{p} \in \mathbf{r}_{\mathbf{x}}^{b}(t)}{\operatorname{argmin}} f(\mathbf{p};\theta),\ \forall b \in \{0..B_{\mathbf{x}}^{\emptyset}\} \right\}. \tag{4.13}$$

We collect $\mathcal{P}_{\mathbf{x}}^{\text{bg}}(\theta)$ for all background pixels $\mathbf{x} \in \mathcal{X}_{\text{bg}}$ and obtain $\{\mathcal{P}_{\mathbf{x}}^{\text{bg}}(\theta)\}_{\mathbf{x} \in \mathcal{X}_{\text{bg}}}$. We use $\mathcal{X}_{\text{bg}}$, which is different from $\mathcal{X}_{\text{in}}$ in IDR [Reshetouski *et al.*, 2011]. Then, the point set for carving becomes

$$\mathcal{P}_{\text{carve}}(\theta) = \{\mathcal{P}_{\mathbf{x}}^{\text{bg}}(\theta)\}_{\mathbf{x} \in \mathcal{X}_{\text{bg}}}. \tag{4.14}$$

Figure 4.3(a-b) illustrates the point selection for background pixels for carving the shape.

Modeling. For the modeling, we select a point on the foreground ray with the minimum distance to the current shape without knowing the label. As we will select the point with the minimum SDF value again on this ray, the selected point will have the minimum value on all foreground rays $\{\mathbf{r}_{\mathbf{x}}^{b}(t)\}_{b=0}^{B_{\mathbf{x}}^{\emptyset}}$. We select points only when the ray does *not hit* the object (i.e., $\widehat{\mathbf{M}}(\mathbf{x}) = 0$) because the ray already hitting the object should not model the space further. Thus, the points for modeling $\mathcal{P}_{\mathbf{x}}^{\text{fg,nh}}(\theta)$ from a foreground pixel $\mathbf{x}$ is

$$\mathcal{P}_{\mathbf{x}}^{\text{fg,nh}}(\theta) = \begin{cases} \underset{\mathbf{p} \in \{\mathbf{r}_{\mathbf{x}}^{b}(t)\}_{b=0}^{B_{\mathbf{x}}^{\emptyset}}}{\operatorname{argmin}} f(\mathbf{p};\theta) & \text{if } \widehat{\mathbf{M}}(\mathbf{x}) = 0, \\ \emptyset & \text{if } \widehat{\mathbf{M}}(\mathbf{x}) = 1. \end{cases} \tag{4.15}$$

We collect $\mathcal{P}_{\mathbf{x}}^{\text{fg,nh}}(\theta)$ for all foreground pixels $\mathbf{x} \in \mathcal{X}_{\text{fg}}$ and compute $\{\mathcal{P}_{\mathbf{x}}^{\text{fg,nh}}(\theta)\}_{\mathbf{x} \in \mathcal{X}_{\text{fg}}}$. Then,

$$\mathcal{P}_{\text{model}}(\theta) = \{\mathcal{P}_{\mathbf{x}}^{\text{fg,nh}}(\theta)\}_{\mathbf{x} \in \mathcal{X}_{\text{fg}}}. \tag{4.16}$$

Figure 4.3(c) shows the point selection for foreground pixels for modeling the shape.

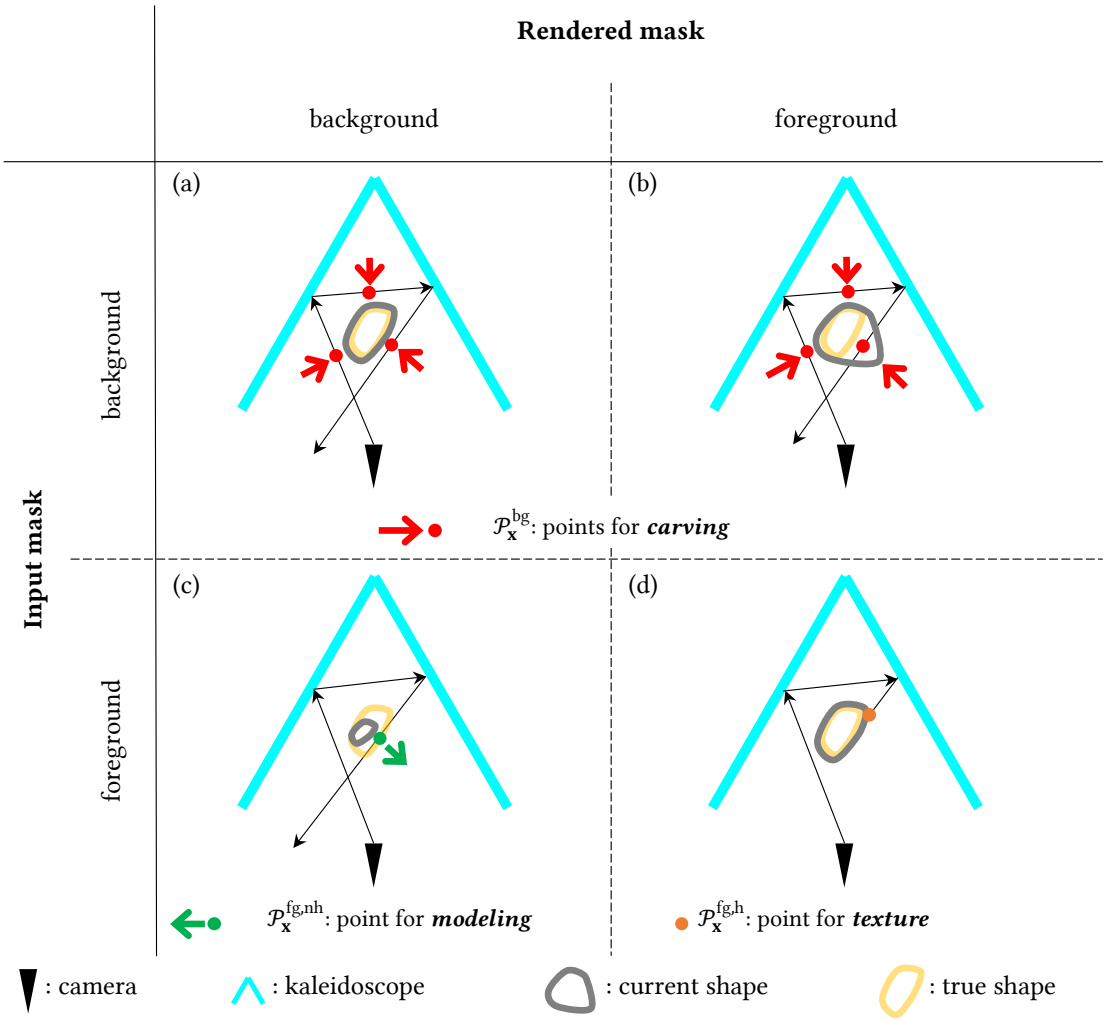


Figure 4.3. **Point selection for sculpting.** We select the points for sculpting differently in each case: (a-b) Background pixel (Equation (4.13)). (c) Foreground pixel not hitting the current shape (Equation (4.15)). (d) Foreground pixel hitting the current shape.

4.3.2 Visual-hull constraint

Although the point selection method described above works promisingly, we introduce another constraint that can be naturally incorporated into our method—the visual hull constraint based on Equation (4.1). As the true bounce b^{true} satisfies $b^{\mathrm{true}} \in \mathcal{H}_{\mathbf{x}}^{\mathrm{vh}}$, we can exclude the bounces that are not in $\mathcal{H}_{\mathbf{x}}^{\mathrm{vh}}$ (i.e., outside the visual hull) from the modeling. The points on these bounces are represented as

$$\mathcal{P}_{\mathbf{x}}^{\mathrm{ovh}} = \{\mathbf{p} \mid \underset{\mathbf{p} \in \mathbf{r}_{\mathbf{x}}^{b}(t)}{\operatorname{argmin}} f(\mathbf{p}; \theta),\ \forall b \notin \mathcal{H}_{\mathbf{x}}^{\mathrm{vh}}\}, \tag{4.17}$$

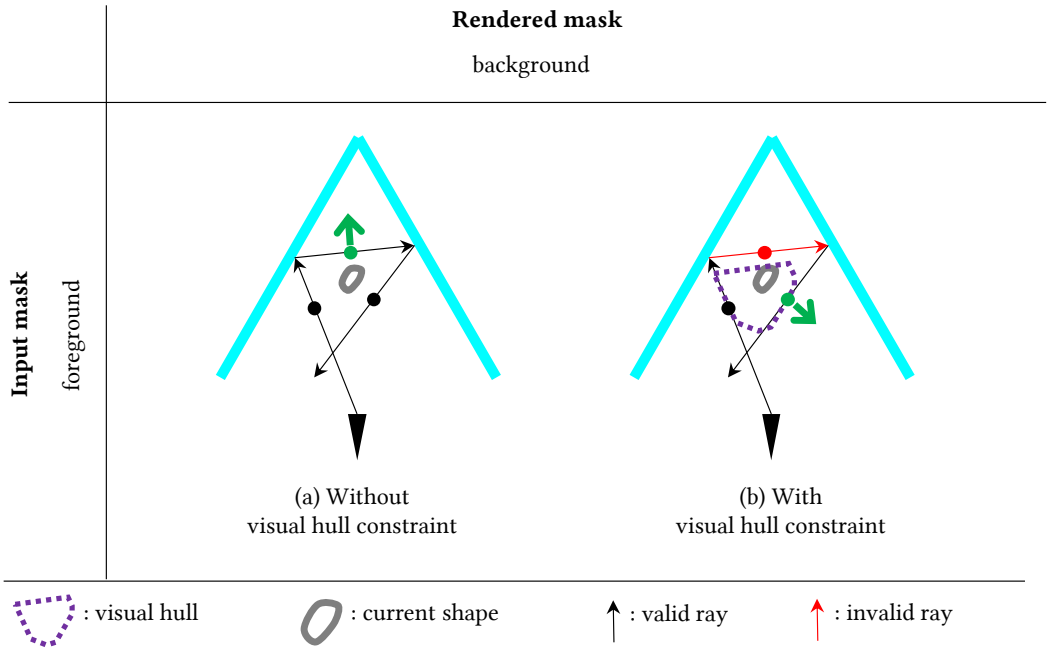


Figure 4.4. **Visual-hull constraint.** The rays not intersecting with the visual hull never intersect with the true shape, so we can exclude the points on these rays from modeling.

and these points should be carved rather than modeled. We compute $\{\mathcal{P}_{\mathbf{x}}^{\text{ovh}}\}_{\mathbf{x}\in\mathcal{X}_{\text{fg}}}$ for all foreground pixels, then the point sets for carving and modeling becomes

$$\mathcal{P}_{\text{carve}}(\theta) = \{\mathcal{P}_{\mathbf{x}}^{\text{bg}}(\theta)\}_{\mathbf{x}\in\mathcal{X}_{\text{bg}}} \cup \{\mathcal{P}_{\mathbf{x}}^{\text{ovh}}(\theta)\}_{\mathbf{x}\in\mathcal{X}_{\text{fg}}}, \tag{4.18}$$

and

$$\mathcal{P}_{\text{model}}(\theta) = \{\mathcal{P}_{\mathbf{x}}^{\text{fg,nh}}(\theta) - \mathcal{P}_{\mathbf{x}}^{\text{ovh}}(\theta)\}_{\mathbf{x}\in\mathcal{X}_{\text{fg}}}. \tag{4.19}$$

Figure 4.4 illustrates the visual hull constraint. Note that our kaleidoscopic sculpting technique naturally handles unreliable pixels as well as reliable pixels.

4.3.3 Texture constraint

One of the main advantages of our technique compared to the visual hull technique [Reshetouski *et al.*, 2011] is that ours is designed to utilize foreground pixels as well as background pixels, and thereby can take advantage of texture information on the foreground pixels. On the other hand, the visual hull technique relies only on the background pixels, ignoring foreground pixels that provide the information of correspondence, and thus it cannot capture concavity, virtual occlusion, or high-frequency details on

Table 4.1. **Comparison of the kaleidoscopic visual hull [Reshetouski *et al.*, 2011] and the proposed neural kaleidoscopic space sculpting.**

	Visual hull [Reshetouski *et al.*, 2011]	Space sculpting (ours)
Background pixels	✓	✓
Foreground pixels	✗	✓
Texture	✗	✓
Shape representation	voxel	neural SDF
Shape update	subtract	add & subtract

the surface. The texture loss is the same as the texture loss in IDR (Equation (4.4)), but we use the first intersection points along all the bounces as there can be multiple bounces intersecting with the object. We collect these points from all foreground pixels and use them for the texture loss as

$$\mathcal{P}_{\text{tex}} = \{\mathcal{P}_{\mathbf{x}}^{\text{fg,h}}(\theta)\}_{\mathbf{x}\in\mathcal{X}_{\text{fg}}}. \tag{4.20}$$

Figure 4.3(d) shows the point selection for foreground pixels for the texture constraint.

4.4 Information in a kaleidoscopic image

A kaleidoscopic image encodes information in a more complicated way compared to an image without any reflection—a kaleidoscopic pixel $\mathbf{x}$ is associated with multiple rays $\{\mathbf{r}_{\mathbf{x}}^{b}(t)\}_{b=0}^{B_{\mathbf{x}}^{\emptyset}}$, whereas a regular pixel is associated with just a single ray (i.e., backprojected ray).

Conventional kaleidoscopic imaging methods [Ahn *et al.*, 2021a; Xu *et al.*, 2018] chose the ray out of all the bounces that is intersecting with the object (i.e., labeling) and it could achieve the benefit of the redistribution of the ray in the kaleidoscope. Ours not only achieves this benefit but also achieves another benefit of using multiple rays $\{\mathbf{r}_{\mathbf{x}}^{b}(t)\}_{b=0}^{B_{\mathbf{x}}^{\emptyset}}$ per pixel either for carving or modeling. Interestingly, not only the background rays but also the foreground pixels can be multiply used by making use of the rays not hitting the visual hull for carving.

Comparison to kaleidoscopic visual hull [Reshetouski *et al.*, 2011]. Table 4.1 summarizes the differences between the visual hull and ours. They use the same number of rays for background pixels but ours additionally uses foreground pixels and thereby texture information. Also, we have more rays to carve as we collect points for carving $\mathcal{P}_{\mathbf{x}}^{\text{ovh}}$ from foreground pixels.

4.5 Implementation

Hardware and calibration. Figure 4.5 shows our hardware prototype used for the the experiments, comprising an RGB camera (FLIR Blackfly S BFS-U3-200S6C) and four triangular mirrors (Edmund Optics 46-656 customized). The mirrors are coated with metal, with the size of 200 mm × 307 mm and the surface flatness of 4–6λ. The image size is 5472 px × 3648 px. We calibrated the kaleidoscope following our kaleidoscopic structured light [Ahn *et al.*, 2021a], which uses a reference sphere object of diameter 40 mm. The calibration error is 346 µm in sphere fitting error and 1.327 px in reprojection error.

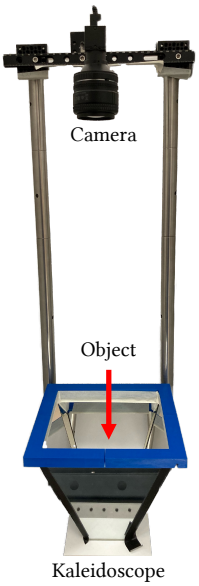


Figure 4.5. **Prototype.**

Algorithm. Algorithm 1 shows how the points are selected and used for the optimization in the proposed neural kaleidoscopic space sculpting.

Implementation Details. We obtain the input mask by computing the difference between the images with and without the object, and manually refining the mask using Adobe Photoshop for some pixels. For the shape and texture network, we use the same architecture as IDR [Yariv *et al.*, 2020]. The shape network is initialized to be an SDF of an approximate unit sphere [Atzmon and Lipman, 2020]. We provide the results with a different initialization in the supplemental where the network is pretrained to represent the visual hull using [Gropp *et al.*, 2020]. For the final mesh results, we ran a marching cube from $f(\cdot;\theta)$ and extracted the largest connected part for the final results [Yariv *et al.*, 2020]. The weights used for the loss are $\lambda_{\text{tex}} = 1.0$ $\lambda_{\text{eik}} = 0.1$ $\lambda_{\text{model}} = 100.0$ $\lambda_{\text{carve}} = 20.0$. Note that λ_{carve} is smaller than λ_{model} by 5, which is about the average number of rays per pixel.

Mask processing for visual hull. For the implementation of the baseline visual hull [Reshetouski *et al.*, 2011], we dilate the input mask to handle the false background pixels caused by the incorrect masking or the gap between mirrors. Figure 4.6 shows the visual hull results with different dilations. We use 8 pixels of dilation for the baseline results used in comparisons, and conservatively use 16 pixels of dilation ignoring mirror boundary for obtaining $\mathcal{H}_{\mathbf{x}}^{\text{vh}}$ for the visual hull constraint. The voxel resolution of the visual hull is 128 on each axis.

Input: Kaleidoscopic image $\mathbf{I}$ and mask $\mathbf{M}$

Output: Neural network θ, ϕ for shape and texture

Initialize $\mathcal{P}_{\text{carve}} = \emptyset$, $\mathcal{P}_{\text{model}} = \emptyset$;

for $b = 0 : B_{\mathbf{x}}^{\emptyset}$ **do**

  for $\mathbf{x} \in \mathcal{X}_{\text{batch}}$ **do**

    Compute intersection b/w $\mathbf{r}_{\mathbf{x}}^{b}(t)$ and $f(\cdot;\theta)$;

    if $\mathbf{M}(\mathbf{x}) = 0$ **then**

      `// BG (Eq.(14))`

      $\mathcal{P}_{\text{carve}} \leftarrow \mathcal{P}_{\text{carve}} \cup \mathcal{P}_{\mathbf{x}}^{\text{bg}}$;

    else

      `// FG ((Eqs.(17-19)))`

      $\mathcal{P}_{\text{model}} \leftarrow \mathcal{P}_{\text{model}} \cup \mathcal{P}_{\mathbf{x}}^{\text{fg,nh}} - \mathcal{P}_{\mathbf{x}}^{\text{ovh}}$;

      $\mathcal{P}_{\text{carve}} \leftarrow \mathcal{P}_{\text{carve}} \cup \mathcal{P}_{\mathbf{x}}^{\text{ovh}}$;

      $\mathcal{P}_{\text{tex}} \leftarrow \mathcal{P}_{\text{tex}} \cup \mathcal{P}_{\mathbf{x}}^{\text{fg,h}}$;

    end

  end

end

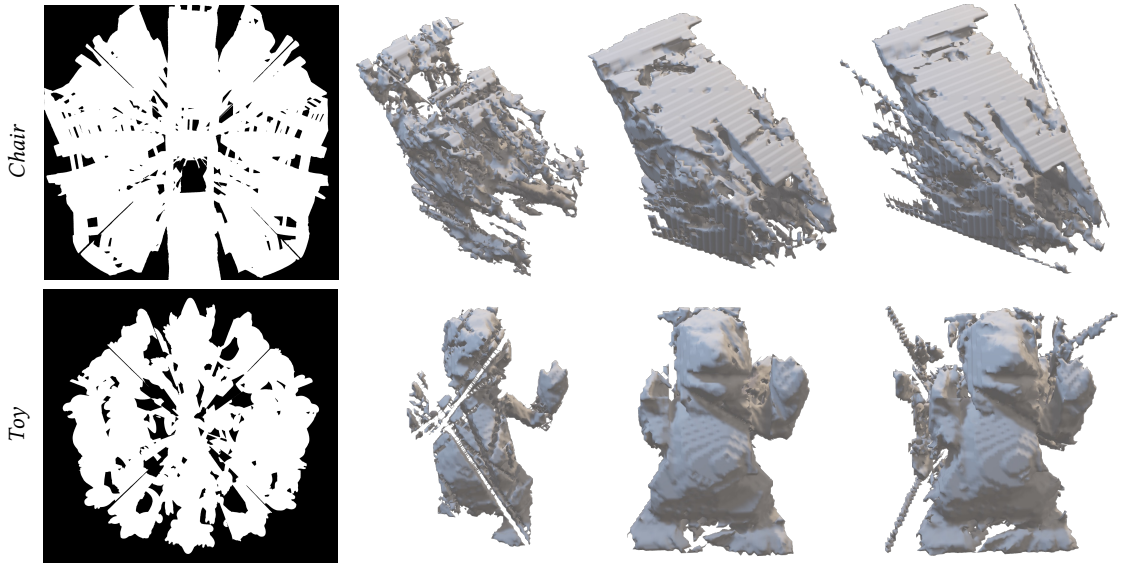
Update θ and ϕ with $\text{loss}(\theta, \phi; \mathcal{P}_{\text{carve}}, \mathcal{P}_{\text{model}}, \mathcal{P}_{\text{tex}})$;

Repeat with different batches;

Algorithm 1: Algorithm for neural kaleidoscopic space sculpting

Manual mask refinement. We obtain the input mask first by computing the difference between the images with and without the object, and then manually refining the mask using Adobe Photoshop for correcting some pixels. Figure 4.7 shows the effect of the manual mask refinement. Although they provide similar results overall, the manual refinement provides better results in the regions where the difference between the images with and without the object is not clear (e.g., the tail of *Toy*).

Initial shape. We use an approximate unit sphere as an initial shape [Atzmon and Lipman, 2020] in the experiments. Figure 4.8 shows the effect of the initial shape by comparing the result with another initialization obtained by fitting the SDF to the visual hull using IGR [Gropp *et al.*, 2020]. We observe the initial shape does not affect to the final shape significantly.



(a) Input mask (b) 0 pixels (c) 8 pixels (d) 16 pixels + mirror boundary

Figure 4.6. **Mask processing for the baseline method (kaleidoscopic visual hull [Reshetouski *et al.*, 2011]).** We dilate the input mask to compute the visual hull to handle the incorrect masking and the gap between mirrors.

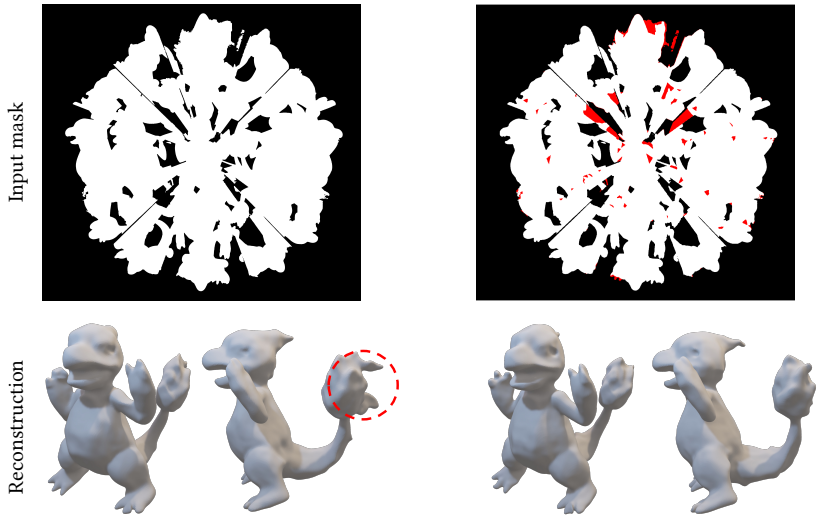
4.6 Results

We evaluate the proposed method through both simulated and real experiments. Our implementation and data are available on our project page [Ahn *et al.*, 2023].

4.6.1 Simulated experiments

We simulate a kaleidoscopic image of an armadillo of height 60 mm, and Figure 4.9 and Table 4.2 show the qualitative and quantitative results for the simulated experiment.

Comparisons to baseline methods. We compare our method with other kaleidoscopic imaging methods: kaleidoscopic visual hull [Reshetouski *et al.*, 2011], and IDR after decomposing the kaleidoscopic image into virtual multi-view images. For the virtual multi-view decomposition, we test several variations in the use of background rays while using the foreground label from the visual hull: (1) use only first background rays, (2) use only final background rays, and (3) use all background rays. Also, we have: (4) use only reliable pixels, and (5) use the ground-truth labels with all background rays. Figure 4.9(a-e) shows the results for the baselines (1–5), and Figure 4.9(f) shows the kaleidoscopic visual hull.



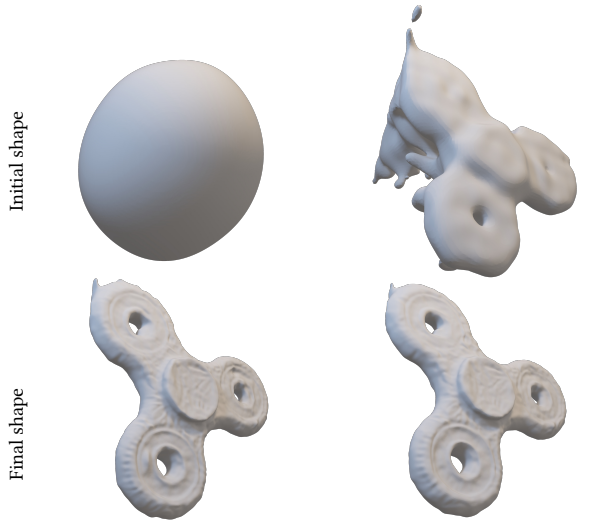
(a) Auto mask (b) Manual mask

Figure 4.7. **Effect of manual mask refinement.** (a) Mask automatically generated from the difference between the images with and without the object. (b) Mask additionally refined using Adobe Photoshop manually.

Ours with all constraints performs best overall qualitatively and quantitatively, with the exception of PSNR where it ranks second. There are artifacts near the left ear of the armadillo in (Figure 4.9(c-d)) because of the incorrect label from the visual hull. This is caused by the virtual occlusion as the background near the left ear is occluded by another virtual object and observed as a foreground, and thereby the space is not carved. Our method does not have this problem as we are jointly solving the labeling and 3D reconstruction.

Figure 4.10 shows the comparison to the baseline method of kaleidoscopic visual hull [Reshetouski *et al.*, 2011] with more synthetic kaleidoscopic images. We observe the proposed method produces superior results than the baseline method.

Ablation study. We conduct an ablation study with the silhouette constraint, visual hull constraint, and texture constraint as shown in Figure 4.9(g-j) and Table 4.2. Silhouette constraint provides the result without details, and texture constraint captures the detail on the surface. Adding the visual hull constraint additionally improves the result.



(a) Sphere initialization (b) Visual hull initialization

Figure 4.8. **Effect of initial shape.** We observe the initial shape (i.e., approximate unit sphere or visual hull) does not affect to the final shape significantly.

Table 4.2. **Quantitative results for simulated experiments.**

Method	**PSNR** [dB] ↑	**chamfer** [μm] ↓	**mask err** [%] ↓	**label err** [%] ↓
visual hull [Reshetouski *et al.*, 2011]	-	9.11	0.93	4.12
IDR w/ VH label (first ray)	6.92	2.3×10^3	26.8	37.71
IDR w/ VH label (last ray)	12.21	1.8×10^3	21.63	20.67
IDR w/ VH label (all rays)	21.84	4.42	0.80	2.74
IDR w/ VH label (reliable)	21.89	4.39	0.78	2.64
IDR w/ GT label	22.53	1.87	0.62	1.42
ours, SIL	13.73	5.53	0.95	3.11
ours, SIL+VH	13.95	6.22	0.80	2.45
ours, SIL+TEX	23.04	2.56	0.59	1.66
ours, SIL+VH+TEX	22.85	1.87	0.57	1.39

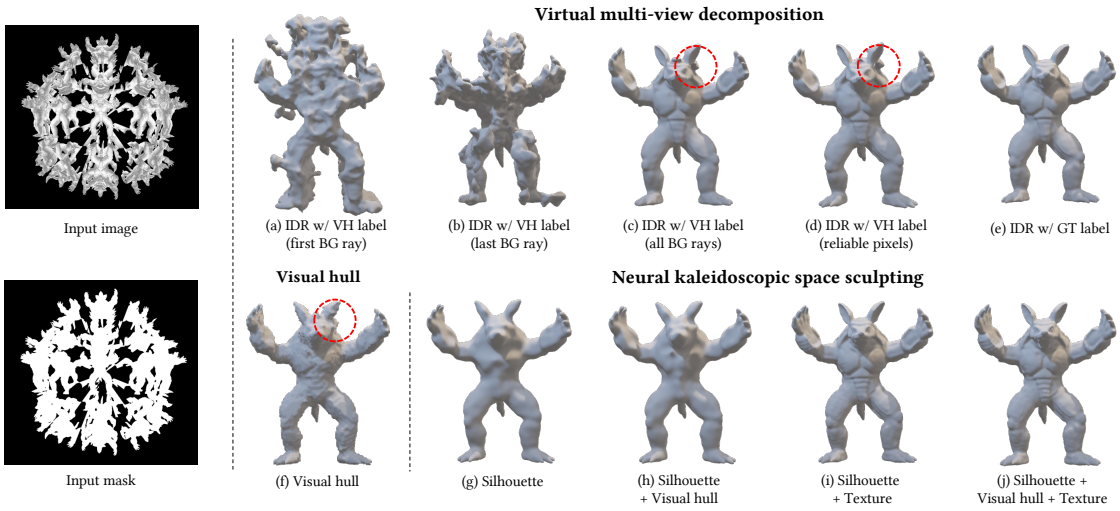


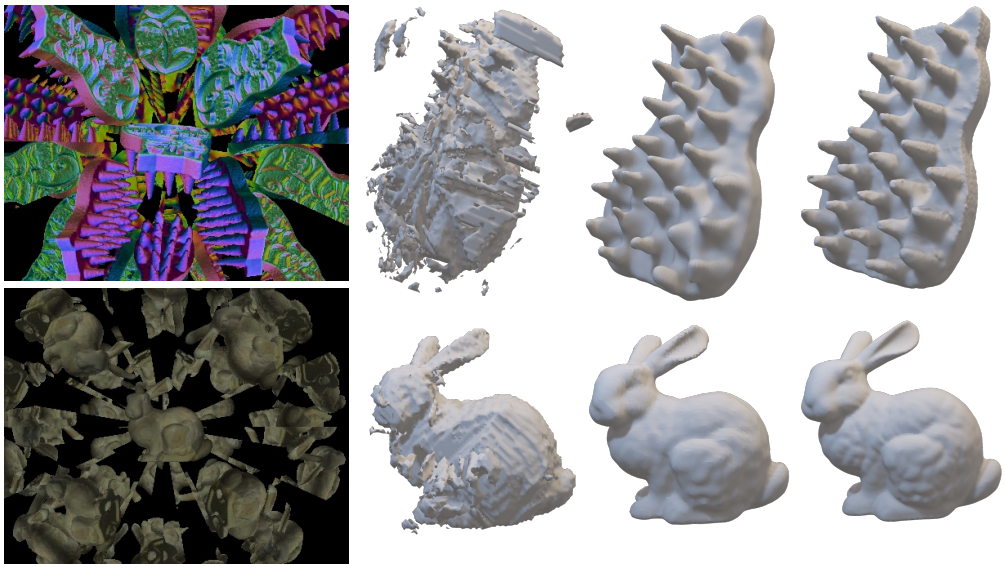
Figure 4.9. **Simulated experiments.** We compare our method with other kaleidoscopic imaging methods on the simulated image of the armadillo. (a-e) Virtual multi-view decomposition followed by IDR, where the kaleidoscopic image is decomposed into virtual multi-view images either with the visual hull label or ground-truth labels. (f) Kaleidoscopic visual hull. (g-j) Ablation study of our method.

4.6.2 Real experiments

We capture a kaleidoscopic image with our hardware prototype and reconstruct the 3D shape. The objects are placed inside the kaleidoscope either directly on the mirror or hung with strings. The size of the object is within 100 mm (e.g., *Treble clef* has the height 100 mm).

Scanned objects. Figure 4.15 shows the reconstruction results on the real objects exhibiting a range of shapes with different visibilities and reflectances with different textures and materials. The label map is visualized by sorting the label in ascending order and using the "rainbow" Matplotlib colormap. We observe that our technique produces high-quality results for this variety of objects.

Statistics. Table 4.3 shows the statistics of the results for each object. The number of viewpoints (i.e., the number of different labels) is about 100 viewpoints for all objects, which shows the light direction is redistributed well by the kaleidoscope. The number of rays per pixel is $5.67 - 5.80$ for foreground pixels and 5.97–6.34 for background pixels, which shows each pixel in a kaleidoscope produces about 6 times more information compared to a regular pixel without any reflection. PSNR and mask error shows the reconstructed shape is photo-consistent and silhouette-consistent.



(a) Kaleidoscopic image (b) Visual hull (c) Ours (d) Ground truth

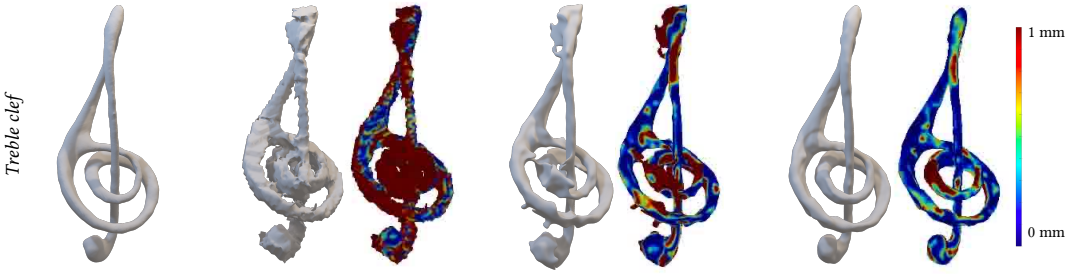
Figure 4.10. **Comparison to visual hull [Reshetouski *et al.*, 2011] on synthetic data.**

Table 4.3. **Statistics for real experiments.**

object	**#views**	**FG #rays / #pixels**	**BG #rays / #pixels**	**PSNR** [dB] ↑	**mask err** [%] ↓
Toy	107	5.68	6.08	20.47	1.55
Chair	112	5.77	6.34	22.19	1.38
Treble clef	123	5.72	5.97	15.91	3.73
Venus	88	5.72	6.02	25.83	1.07
Monkey	84	5.80	5.99	21.77	2.42
Spinner	111	5.67	6.06	21.62	0.75

Comparison to kaleidoscopic visual hull. Figure 4.14 show the comparison to the baseline method of kaleidoscopic visual hull [Reshetouski *et al.*, 2011] for real data. We observe the proposed method produces superior results than the baseline method.

Comparison to kaleidoscopic structured light. Figure 4.11 shows the comparison to the kaleidoscopic structured light [Ahn *et al.*, 2021a], which uses a projector additionally for labeling and better correspondences. We compare the results from our method and baseline methods to the result of kaleido-



(a) Kaleidoscopic structured light | (b) Visual hull (Avg. dist: 4.01mm) | (c) IDR w/ VH label (Avg. dist: 1.92mm) | (d) Ours (Avg. dist: 0.36mm)

Figure 4.11. **Comparison to kaleidoscopic structured light [Ahn *et al.*, 2021a].** Our method produces a comparable result to kaleidoscopic structured light that uses an additional projector for active illumination.

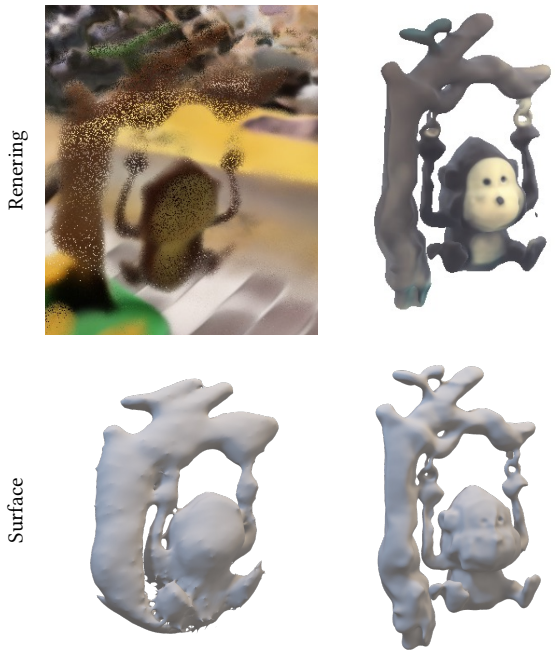
scopic structured light for *Treble clef* object. For each result, we compute the distance of each vertex from the kaleidoscopic structured light mesh, and visualize it as the vertex color. Our technique produces high-quality results both qualitatively and quantitatively.

Comparison to Nerfies [Park *et al.*, 2021a]. Figure 4.12 shows the comparison with Nerfies, which can handle dynamic objects. We captured the video of *Monkey* data where the monkey playing on the swings using a smartphone camera (iPhone SE 2) from surrounding views. Nerfies cannot capture the deformation of this dynamic object and fail to reconstruct the object. By contrast, ours can capture multiple viewpoints in a single shot and obtain the full-surround reconstruction. Note that both methods use different inputs, where our method captures multiple viewpoints of a dynamic object in *a single frame*, whereas Nerfies reconstructs dynamic objects from *multiple frames* by optimizing for a deformation field.

Ablation study. Figure 4.13 shows the ablation study on the real chair data. The result using only silhouette constraint produces an over-carved result, partly because of the imperfect input mask. Adding the visual hull constraint improves the over-carved parts. Adding the texture constraint greatly improves the result, and applying the visual hull constraint additionally improves some artifacts on the armrest.

4.7 Discussion

High-genus topology with neural implicit representations. In this chapter, our focus did not include handling objects with high-genus topology, unlike our approach in the kaleidoscopic structured light. This limitation stems from the restricted representational capacity of neural surface representations.



(a) Nerfies (b) Ours

Figure 4.12. **Comparison to Nerfies [Park *et al.*, 2021a].** Nerfies fail to model the large deformation between the different views of the swinging monkey, whereas ours does not suffer from the dynamic movement since we capture multiple viewpoints in a single shot.

For instance, complex high-genus objects like the elephant could not be effectively reconstructed using neural representations, even with real multi-view images that provide far more pixel information than a kaleidoscopic image. Future advancements could explore improvements through topological derivatives, as suggested in recent studies Mehta *et al.* [2022, 2023], which are methods of incorporating discrete topology changes into the modeling of complex shapes and surfaces.

Summary. We introduce a single-shot full-surround 3D reconstruction method producing a silhouette-consistent and photo-consistent shape from a single kaleidoscopic image. Based on neural SDF representation, our method carves and models the space by selecting adequate points without the necessity of labels, and jointly solves the labeling and 3D reconstruction problems. We show that our method takes the advantage of the information in the kaleidoscope by reusing the pixels multiple times by the number of reflections.

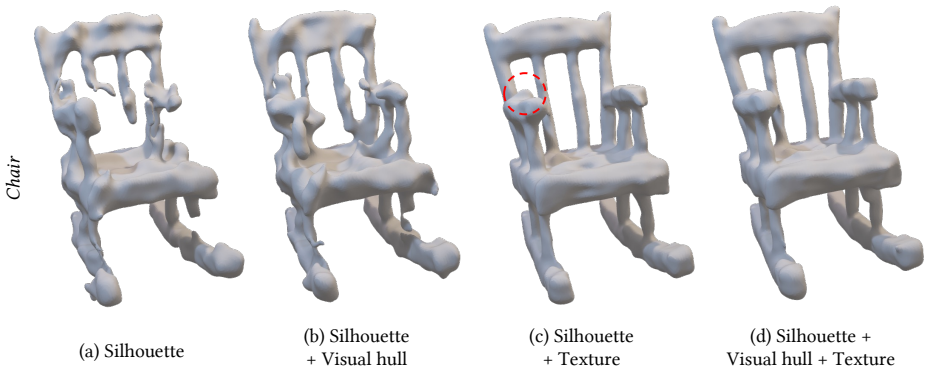


Figure 4.13. **Ablation study on *Chair* data.**

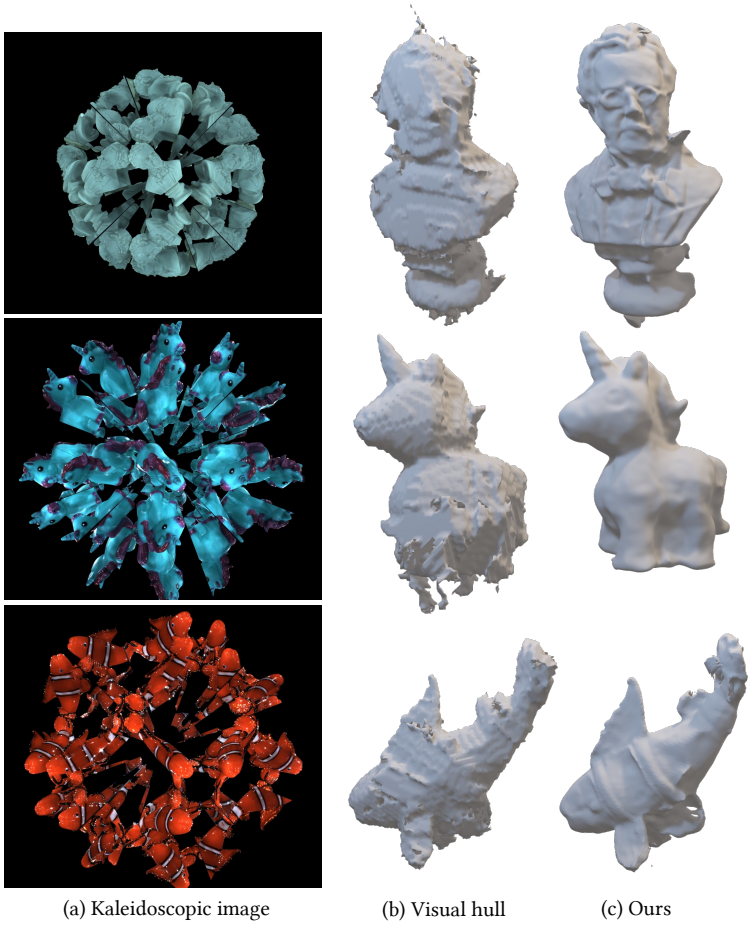


Figure 4.14. **Comparison to visual hull [Reshetouski *et al.*, 2011] on real data.**

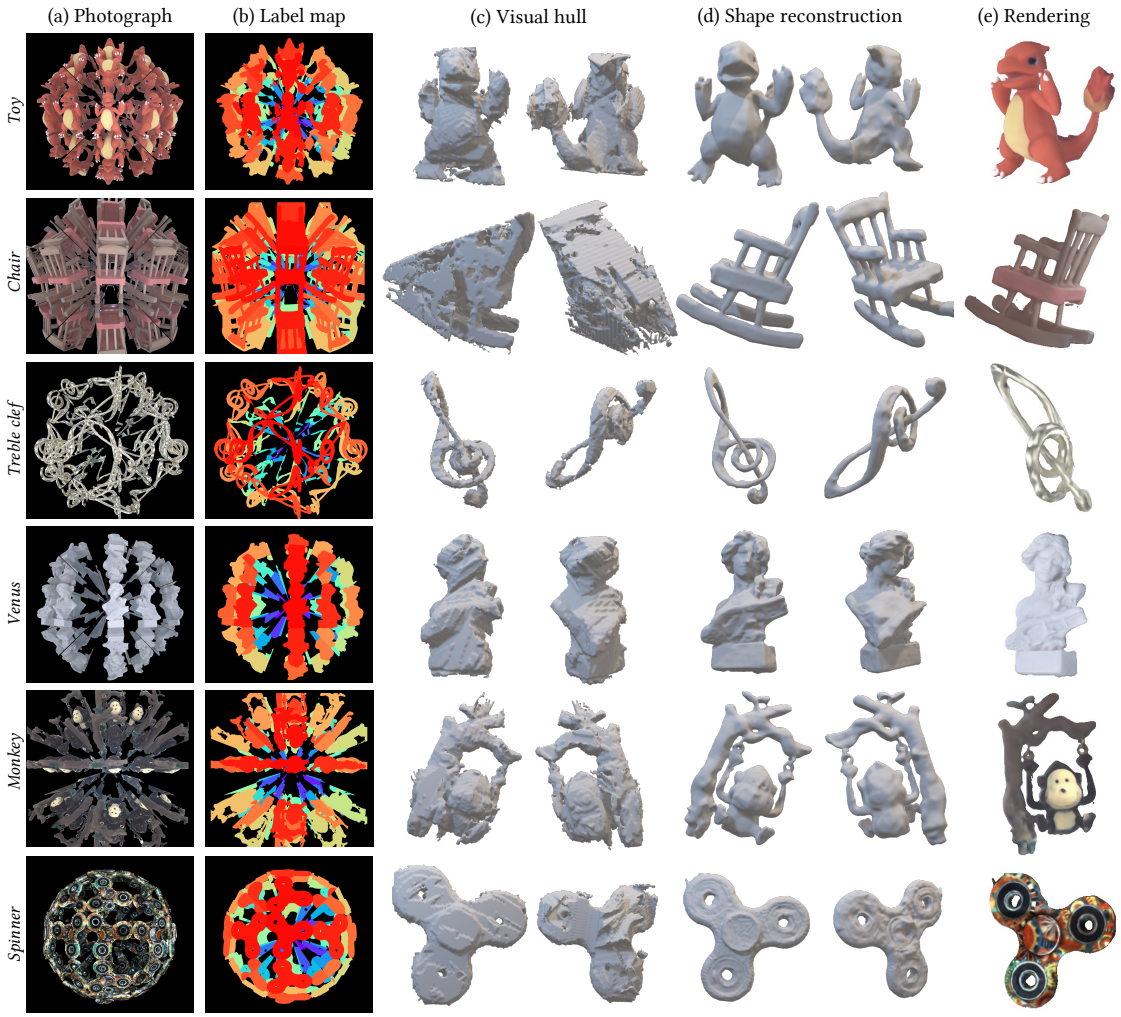


Figure 4.15. **Real object reconstructions from neural kaleidoscopic space sculpting.** Please zoom in to see the differences.

5 Conclusion

Throughout this thesis, we have established a foundation for the theoretical and practical application of kaleidoscopic techniques in full-surround 3D reconstruction. Our focus has been primarily on reconstructing highly complex objects, particularly those with intricate geometries and self-occlusions.

From a theoretical standpoint, this thesis has formulated a kaleidoscopic parallel to the multiple view geometry in classical computer vision. On a practical level, we have introduced a system that is straightforward to build, calibrate, and deploy, making advanced 3D reconstruction techniques accessible to the average consumer of 3D technology.

Chapter 1 of the thesis delved into the motivation for using kaleidoscopic imaging, providing a comparative analysis between traditional methods of full-surround 3D reconstruction and the kaleidoscopic approach. This comparison highlighted the enhanced capabilities and efficiencies brought about by the kaleidoscopic method.

Chapter 2 delved into the design and calibration of kaleidoscopes. Here, we outlined crucial design considerations, including the quantity and configuration of mirrors, and developed quantitative metrics to evaluate these designs. This chapter demonstrated how a design, considering factors like the number and size of mirrors and their configuration, can impact the efficacy of the kaleidoscopic system.

Chapter 3 introduced kaleidoscopic structured light, a novel adaptation of traditional structured light. Key in this chapter was the development of epipolar labeling and its theoretical implications. Our labeling technique, using constraints specific to the kaleidoscopic setup, particularly the epipolar geometry, allows for the decoding of mirror sequences at projector and camera pixels. This method offers theoretical guarantees on the recoverability of labels, enhancing the practicality and accuracy of 3D reconstruction.

Chapter 4 presented kaleidoscopic neural rendering, integrating neural rendering within a kaleidoscopic framework. This approach not only improves the handling of reflections but also enables single-shot full-surround 3D reconstruction. This capability is particularly crucial for dynamic objects as each frame contains all necessary information for complete surround coverage. Theoretically, this

chapter is significant for solving labeling and 3D reconstruction jointly, thereby removing the need for explicit labeling.

5.1 Limitations

Our technique has a number of limitations, which are inherent in the use of a kaleidoscope. First, the size of objects we can scan is restricted by the kaleidoscope; for our lab setup, this constrains our technique to objects that fit in a sphere of diameter 4 inches. Second, the total number of pixels that we have at our disposal is limited to that of a single image sensor; divvying this pixel budget across the many (virtual) views results in lower resolution imagery, especially when we consider multi-view alternatives where the total pixel count grows linearly with the number of cameras. Third, it is challenging to reconstruct transparent objects or objects with refraction and reflection effects. Reconstruction of such objects is challenging for many 3D reconstruction techniques that need to establish correspondences across viewpoints, and not just ours. Compared to non-kaleidoscopic techniques, our kaleidoscopic method further needs the assumption that the object is opaque in order to be able to establish unique correspondences between each pixel and a single virtual camera (i.e., a unique label for each pixel).

5.2 Future directions

Despite the kaleidoscopic 3D reconstruction techniques developed in this thesis, both in theoretical foundation and practical applications, there are still several aspects of the kaleidoscopic imaging problem that remain unsolved and provide several exciting future research directions:

- **Uncalibrated kaleidoscopic imaging.** Development of methods for kaleidoscopic 3D reconstruction using *uncalibrated* kaleidoscopic images could significantly enhance the technology's accessibility. Our current technique depends on a calibrated kaleidoscopic system, requiring a reference object with initial manual labeling. An intriguing future direction would be the kaleidoscopic equivalent of structure-from-motion techniques, such as the five-point method for calculating the essential matrix, termed as 'structure from mirrors'. This advancement could allow kaleidoscopic methods to be applied in a more versatile, in-the-wild manner, such as in gyms with mirrored walls or even in Yayoi Kusama's Infinity Mirror Rooms.

- **Non-planar mirrors.** Exploring the use of non-planar mirrors may offer a more diverse distribution of rays, potentially capturing more complex geometries and textures [De Zeeuw and Sankaranarayanan, 2022]. Non-planar mirrors could provide continuously varying viewpoints due to the different angles

at which each ray intersects with the mirror surface. The construction and calibration of non-planar mirrors, particularly their pose relative to the camera, is challenging but an interesting future direction.

- **Arbitrary mirror reconstruction.** Expanding 3D reconstruction to include mirrors of arbitrary shapes and configurations in natural settings, such as the reflective surfaces of Chicago's Cloud Gate ('The Bean'), could significantly broaden the scope of kaleidoscopic 3D reconstruction. Although the production of non-planar mirrors typically incurs higher costs, which diverges from the initial aim of creating a cost-effective 3D scanner, the capability to reconstruct any mirror shape and position in real-world environments offers vast potential. This advancement would not only facilitate the use of common reflective surfaces like windows in 3D scanning but also open up the possibility of the extraction of reflective data from these surfaces, thereby enriching the kaleidoscopic 3D reconstruction in the wild.

- **Mirror detection.** Identifying mirrors in uncontrolled, or in-the-wild environments represents a promising research direction. Combined with arbitrary mirror reconstruction, mirror detection could transform the use of reflective surfaces in 3D scanning, leading to the natural kaleidoscope in unknown environments. Key to this detection are cues like edge discontinuity or the correlation between real and mirrored images, including changes in orientation (i.e., left-handed and right-handed). Such identification opens up the possibility for leveraging any reflective surface for 3D scanning, broadening its application scope, especially in forensic imaging where it offers novel information cue from limited observation.

- **Material property acquisition.** Kaleidoscope can provide rich information for the acquisition of material property, such as BRDF or subsurface scattering, by offering a large range of diverse set of illuminating and imaging directions. Although this kaleidoscopic setup involves a key challenge of interreflections, which mix the effects of multiple illumination directions [Shem-Tov *et al.*, 2020], it offers a richer set of constraints on the material property compared to the equivalent setup without interreflections, which can be resolved computationally.

- **Diffuse kaleidoscope.** Beyond extracting reflections from glossy surfaces, utilizing diffuse reflections presents a novel research area. With the advancements in non-line-of-sight imaging, which treats diffuse walls as mirrors, exploring the use of multiple diffuse surfaces (i.e., diffuse kaleidoscope) for complete 3D scanning is an interesting direction. This would require tackling the labeling problem in conjunction with non-line-of-sight imaging techniques.

- **Kaleidoscope for non-optical signals.** Another intriguing possibility lies in employing non-optical signals, such as sound, in kaleidoscopic imaging. Due to its longer wavelength, sound inherently offers more specular reflection, providing cues for 3D reconstruction both acoustically and visually. Combining optical and acoustic signals could yield synergistic benefits for applications like source separation, detection, dereverberation, as well as 3D reconstruction.

Bibliography

Kfir Aberman, Oren Katzir, Qiang Zhou, Zegang Luo, Andrei Sharf, Chen Greif, Baoquan Chen, and Daniel Cohen-Or. 2017. Dip transform for 3D shape reconstruction. *ACM Transactions on Graphics (TOG)* 36, 4 (2017).

Byeongjoo Ahn, Ioannis Gkioulekas, Michael De Zeeuw, and Aswin C. Sankaranarayanan. 2023. Project page: Neural Kaleidoscopic Space Sculpting. `https://imaging.cs.cmu.edu/neural_kaleidoscopic_space_sculpting`.

Byeongjoo Ahn, Ioannis Gkioulekas, and Aswin C. Sankaranarayanan. 2021a. Kaleidoscopic structured light. *ACM Transactions on Graphics (TOG)* 40, 6 (2021).

Byeongjoo Ahn, Ioannis Gkioulekas, and Aswin C. Sankaranarayanan. 2021b. Project page: Kaleidoscopic structured light. `https://imaging.cs.cmu.edu/kaleidoscopic_structured_light`

Matan Atzmon and Yaron Lipman. 2020. Sal: Sign agnostic learning of shapes from raw data. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Shaun Bangay and Judith D Radloff. 2004. Kaleidoscope configurations for reflectance measurement. In *International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*.

Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. 2021. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *IEEE International Conference on Computer Vision*.

Paul J Besl and Neil D McKay. 1992. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 14, 2 (1992).

David Brewster. 1858. *The kaleidoscope, its history, theory and construction: with its application to the fine and useful arts*. J. Murray.

Yan Cui, Sebastian Schuon, Derek Chan, Sebastian Thrun, and Christian Theobalt. 2010. 3D shape scanning with a time-of-flight camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Michael De Zeeuw and Aswin C Sankaranarayanan. 2022. Wide-Baseline Light Fields using Ellipsoidal Mirrors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2022).

Martin A Fischler and Robert C Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 6 (1981).

Keith Forbes, Fred Nicolls, Gerhard De Jager, and Anthon Voigt. 2006. Shape-from-silhouette with two mirrors and an uncalibrated camera. In *European Conference on Computer Vision (ECCV)*.

Martin Fuchs, Markus Kächele, and Szymon Rusinkiewicz. 2013. Design and fabrication of faceted mirror arrays for light field capture. In *Computer Graphics Forum*, Vol. 32.

Gaurav Garg, Eino-Ville Talvala, Marc Levoy, and Hendrik PA Lensch. 2006. Symmetric Photography: Exploiting data-sparseness in reflectance fields. In *Symposium on Rendering*.

Abhijeet Ghosh, Graham Fyffe, Borom Tunwattanapong, Jay Busch, Xueming Yu, and Paul Debevec. 2011. Multiview face capture using polarized spherical gradient illumination. *ACM Transactions on Graphics (TOG)* 30, 6 (2011).

Joshua Gluckman and Shree K Nayar. 2001. Catadioptric stereo using planar mirrors. *International Journal of Computer Vision (IJCV)* 44, 1 (2001).

Joshua Gluckman and Shree K Nayar. 2002. Rectified catadioptric stereo sensors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 24, 2 (2002).

Ardeshir Goshtasby and William A Gruver. 1993. Design of a single-lens stereo camera system. *Pattern Recognition* 26, 6 (1993).

Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. 2020. Implicit Geometric Regularization for Learning Shapes. In *International Conference on Machine Learning*.

Jefferson Y Han and Ken Perlin. 2003. Measuring bidirectional texture reflectance with a kaleidoscope. *ACM Transactions on Graphics (TOG)* 22, 3 (2003).

John C Hart. 1996. Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer* 12, 10 (1996).

Richard Hartley and Andrew Zisserman. 2004. *Multiple view geometry in computer vision* (2nd ed.). Cambridge University Press.

Michael Holroyd, Jason Lawrence, and Todd Zickler. 2010. A coaxial optical scanner for synchronous acquisition of 3D geometry and surface reflectance. *ACM Transactions on Graphics (TOG)* 29, 4 (2010).

Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. 1992. Surface Reconstruction from Unorganized Points. *Computer Graphics (SIGGRAPH'92 proceedings)* 26, 2 (1992).

Bo Hu, Christopher Brown, and Randal Nelson. 2005. *Multiple-view 3-D reconstruction using a mirror*. Technical Report. University of Rochester, Department of Computer Science.

Po-Hao Huang and Shang-Hon Lai. 2006. Contour-based structure from reflection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ivo Ihrke, Ilya Reshetouski, Alkhazur Manakov, Art Tevs, Michael Wand, and Hans-Peter Seidel. 2012. A kaleidoscopic approach to surround geometry and reflectance acquisition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Godisart, Bart Nabbe, Iain Matthews, *et al*. 2017. Panoptic studio: A massively multiview system for social interaction capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 41, 1 (2017).

Kaizhang Kang, Cihui Xie, Chengan He, Mingqi Yi, Minyi Gu, Zimin Chen, Kun Zhou, and Hongzhi Wu. 2019. Learning efficient illumination multiplexing for joint capture of reflectance and shape. *ACM Transactions on Graphics (TOG)* 38, 6 (2019).

Michael Kazhdan and Hugues Hoppe. 2013. Screened poisson surface reconstruction. *ACM Transactions on Graphics (TOG)* 32, 3 (2013).

Kalin Kolev, Petri Tanskanen, Pablo Speciale, and Marc Pollefeys. 2014. Turning mobile phones into 3D scanners. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kiriakos N Kutulakos and Steven M Seitz. 2000. A theory of shape by space carving. *International Journal of Computer Vision (IJCV)* 38, 3 (2000).

Douglas Lanman, Daniel Crispell, and Gabriel Taubin. 2009. Surround structured lighting: 3-D scanning with orthographic illumination. *Computer Vision and Image Understanding (CVIU)* 113, 11 (2009).

Aldo Laurentini. 1994. The visual hull concept for silhouette-based image understanding. 16, 2 (1994).

Daniel Lichy, Jiaye Wu, Soumyadip Sengupta, and David W Jacobs. 2021. Shape and material capture at home. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. 2021. Neural actor: Neural free-view synthesis of human actors with pose control. *ACM Transactions on Graphics* 40, 6 (2021).

Ishit Mehta, Manmohan Chandraker, and Ravi Ramamoorthi. 2022. A level set theory for neural implicit evolution under explicit flows. In *European Conference on Computer Vision (ECCV)*.

Ishit Mehta, Manmohan Chandraker, and Ravi Ramamoorthi. 2023. A Theory of Topological Derivatives for Inverse Rendering of Geometry. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2020. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*.

Hiroshi Mitsumoto, Shinichi Tamura, Kozo Okazaki, Naoki Kajimi, and Yutaka Fukui. 1992. 3-D reconstruction using mirror images based on a plane symmetry recovering method. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 14, 09 (1992).

Daniel Moreno and Gabriel Taubin. 2012. Simple, accurate, and robust projector-camera calibration. In *IEEE International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*.

David W Murray. 1995. Recovering range using virtual multicamera stereo. *Computer Vision and Image Understanding (CVIU)* 61, 2 (1995).

Giljoo Nam, Joo Ho Lee, Diego Gutierrez, and Min H Kim. 2018. Practical svbrdf acquisition of 3d objects with unstructured flash photography. *ACM Transactions on Graphics (TOG)* 37, 6 (2018).

Sameer A Nene and Shree K Nayar. 1998. Stereo with mirrors. In *IEEE International Conference on Computer Vision (ICCV)*.

Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. 2011. KinectFusion: Real-time dense surface mapping and tracking. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*.

NextEngine. 2000. `http://www.nextengine.com` [Accessed Sep. 9, 2021].

Shohei Nobuhara, Takashi Kashino, Takashi Matsuyama, Kouta Takeuchi, and Kensaku Fujii. 2016. A single-shot multi-path interference resolution for mirror-based full 3D shape measurement with a correlation-based ToF camera. In *International Conference on 3D Vision (3DV)*.

Peter Ondrúška, Pushmeet Kohli, and Shahram Izadi. 2015. Mobilefusion: Real-time volumetric surface reconstruction and dense tracking on mobile phones. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 21, 11 (2015).

Jaesik Park, Sudipta N Sinha, Yasuyuki Matsushita, Yu-Wing Tai, and In So Kweon. 2016. Robust multiview photometric stereo using planar mesh parameterization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 39, 8 (2016).

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. 2019. Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. 2021a. Nerfies: Deformable neural radiance fields. In *IEEE International Conference on Computer Vision*.

Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. 2021b. HyperNeRF: A Higher-Dimensional Representation for Topologically Varying Neural Radiance Fields. *ACM Transactions on Graphics* 40, 6 (2021).

Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. 2021a. Animatable neural radiance fields for modeling dynamic human bodies. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. 2021b. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. 2021. D-NeRF: Neural radiance fields for dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Ilya Reshetouski, Alkhazur Manakov, Hans-Peter Seidel, and Ivo Ihrke. 2011. Three-dimensional kaleidoscopic imaging. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Christopher Schwartz, Ralf Sarlette, Michael Weinmann, and Reinhard Klein. 2013. DOME II: a parallelized BTF acquisition system. In *Eurographics Workshop on Material Appearance Modeling: Issues and Acquisition*.

Kfir Shem-Tov, Sai Praveen Bangaru, Anat Levin, and Ioannis Gkioulekas. 2020. Towards reflectometry from interreflections. In *IEEE International Conference on Computational Photography (ICCP)*.

Yuichi Taguchi, Amit Agrawal, Srikumar Ramalingam, and Ashok Veeraraghavan. 2010a. Axial light field for curved mirrors: Reflect your perspective, widen your view. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yuichi Taguchi, Amit Agrawal, Ashok Veeraraghavan, Srikumar Ramalingam, and Ramesh Raskar. 2010b. Axial-cones: Modeling spherical catadioptric cameras for wide-angle light field rendering. *ACM Transactions on Graphics (TOG)* 29, 6 (2010).

Tomu Tahara, Ryo Kawahara, Shohei Nobuhara, and Takashi Matsuyama. 2015. Interference-free epipole-centered structured light pattern for mirror-based multi-view active stereo. In *International Conference on 3D Vision (3DV)*.

Kosuke Takahashi, Akihiro Miyata, Shohei Nobuhara, and Takashi Matsuyama. 2017. A linear extrinsic calibration of kaleidoscopic imaging system from single 3d point. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kosuke Takahashi and Shohei Nobuhara. 2021. Structure of multiple mirror system from kaleidoscopic projections of single 3d point. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2021).

Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. 2022. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Ziyan Wang, Timur Bagautdinov, Stephen Lombardi, Tomas Simon, Jason Saragih, Jessica Hodgins, and Michael Zollhofer. 2021. Learning compositional radiance fields of dynamic human heads. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Hongzhi Wu, Zhaotian Wang, and Kun Zhou. 2015. Simultaneous localization and appearance estimation with a consumer RGB-D camera. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 22, 8 (2015).

Hongzhi Wu and Kun Zhou. 2015. AppFusion: Interactive appearance acquisition using a kinect sensor. *Computer Graphics Forum* 34, 6 (2015).

Rui Xia, Yue Dong, Pieter Peers, and Xin Tong. 2016. Recovering shape and spatially-varying surface reflectance under unknown illumination. *ACM Transactions on Graphics (TOG)* 35, 6 (2016).

Hongyi Xu, Thiemo Alldieck, and Cristian Sminchisescu. 2021. H-NeRF: Neural radiance fields for rendering and temporal reconstruction of humans in motion. In *Advances in Neural Information Processing Systems*.

Ruilin Xu, Mohit Gupta, and Shree K Nayar. 2018. Trapping light for time of flight. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. 2020. Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. In *Neural Information Processing Systems (NeurIPS)*.

Xianghua Ying, Kun Peng, Ren Ren, and Hongbin Zha. 2010. Geometric properties of multiple reflections in catadioptric camera with two planar mirrors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhengyou Zhang. 2000. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 22, 11 (2000).

Zhenglong Zhou, Zhe Wu, and Ping Tan. 2013. Multi-view photometric stereo with spatially varying isotropic materials. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.