

# LookUp3D: Data-Driven 3D Scanning

GIANCARLO PEREIRA\*, New York University, USA

YIDAN GAO\*, New York University, USA

YURII PIADYK\*, New York University, USA

DAVID FOUHEY, New York University, USA

CLAUDIO T. SILVA, New York University, USA

DANIELE PANOZZO, New York University, USA

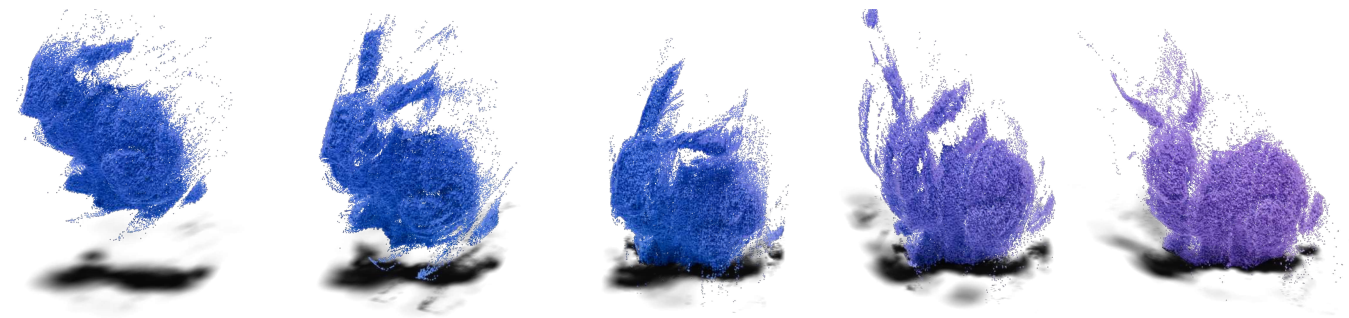


Fig. 1. **Reconstructing Dynamic Motion at 450 fps.** We visualize point cloud reconstructions of a silicone bunny dropping, with consecutive frames being 22 milliseconds apart. *LookUp3D* reconstructs dense 3D geometry at each frame, accurately capturing both deformation and motion despite moderate blur and 1-millisecond exposure. We further validate reconstruction fidelity by estimating gravitational acceleration at  $9.60 \text{ m/s}^2$  (versus its ground truth value of  $9.80 \text{ m/s}^2$ ) from the recovered trajectories, demonstrating the utility of our method for downstream tasks such as physical inference and inverse design.

High speed, high-resolution, and accurate 3D scanning would open doors to many new applications in graphics, robotics, science, and medicine by enabling the accurate scanning of deformable objects during interactions. Past attempts to use structured light, time-of-flight, and stereo in high-speed settings have usually required tradeoffs in resolution or inaccuracy. In this paper, we introduce a method that enables, for the first time, 3D scanning at 450 frames per second at 1 Megapixel, or 1,450 frames per second at 0.4 Megapixel in an environment with controlled lighting. The key idea is to use a per-pixel lookup table that maps colors to depths, which is built using a linear stage. Imperfections, such as lens-distortion and sensor defects are baked into the calibration. We describe our method and test it on a novel hardware prototype. We compare the system with both ground-truth geometry as well as commercially available dynamic sensors like the Microsoft Kinect and Intel Realsense. Our results show the system acquiring geometry of objects undergoing high-speed deformations and oscillations and demonstrate the ability to recover physical properties from the reconstructions.

CCS Concepts: • **Computing methodologies** → **3D imaging; Active vision; Computational photography; Reconstruction**; • **Hardware** → *Emerging optical and photonic technologies; Scanners.*

Additional Key Words and Phrases: 3D Scanning, Geometry Acquisition, Structured Light, Data-Driven, Active Illumination, High-Speed

## 1 INTRODUCTION

3D scanning is ubiquitous in modern technology, enabling industrial metrology, face recognition, cultural heritage preservation, and immersive VR. Various solutions have been introduced to tackle this problem, including touch probes [Dobosz and Woźniak 2005], time-of-flight [Tang et al. 2010], structured light [Salvi et al. 2004],

polarized light [Deschaintre et al. 2021], and event cameras [Rebecq et al. 2018]. Among these, structured light (SL) stands out as a non-contact, high-resolution technique that delivers consistent, repeatable results across diverse materials and object sizes. SL systems typically reconstruct 3D geometry using a calibrated projector-camera pair to project time-varying patterns (e.g., phase-shifted sinusoids or Gray Codes) and analyze their deformations. Through epipolar constraints and triangulation, pixel correspondences are established, achieving submillimeter precision. Despite six decades of advancements in coding schemes [Caspi et al. 1998a; Mirdehghan et al. 2018], hardware synchronization [Koppal et al. 2012; Zhang et al. 2010], and hybrid sensing [Bajestani and Beltrame 2023; Muglikar et al. 2021], two fundamental challenges persist: (1) SL accuracy is bounded by triangulation, which in turn is bounded by the calibration quality of the projector-camera setup; (2) SL capture speed is constrained by the need to project and decode multiple patterns per scan to compute correspondences for triangulation.

In this work, we introduce *LookUp3D*, a novel SL scanning approach that eliminates the need for triangulating correspondences in a calibrated camera-projector system. This shift enables reconstruction with fewer projected patterns and supports higher-speed acquisition for dynamic scenes. *LookUp3D* treats the projector as a simple light source rather than a generator of precise codes, operating with as few as one RGB pattern, and performs depth reconstruction through a per-ray lookup table (LUT) obtained by actual recordings during calibration. By avoiding triangulation between projector and camera, our method eliminates the need to model projector intrinsics or align projector-camera geometry precisely – they all get implicitly baked into the LUT. This allows the system

\*Joint first authors with equal contribution.

to tolerate significant optical distortions or misalignment on the projection side. The recorded LUT not only enables direct, highly parallelizable depth retrieval, but also provides a measure of reconstruction confidence. Each pixel’s match quality – how closely the observed color aligns with the calibration data – serves as an indicator of reliability, allowing for automatic filtering of poorly reconstructed regions. We validate our approach across three hardware configurations: a standard DLP projector, an inexpensive LCD projector, and a custom-built analog projector for high-speed. With the high-speed prototype, we achieve accurate reconstructions at 450 fps and 1 Megapixel resolution. We recover the free-fall of a bunny at 450 fps with relative motion errors below 2% (Figure 1).

Our contributions are: (1) A simple, effective, data-driven algorithm for structured light scanning that does not rely on optical projector calibration or triangulation; and (2) a hardware prototype comprising a digital camera and analog projector capable of acquiring accurate 3D geometry at 450 fps and 1 Megapixel resolution, or 1,450 fps at 0.4 Megapixel, demonstrating our system’s viability.

By rethinking the relationship of illumination, observation, and reconstruction, *LookUp3D* offers a new perspective on 3D scanning that is practical today and extensible for future developments.

## 2 RELATED WORK

While the broader literature on 3D geometry acquisition spans diverse solutions [Lanman and Taubin 2009b], here we focus on SL approaches, which have been extensively surveyed, particularly for high-speed acquisition [Van der Jeught and Dirckx 2016].

*Structured Light Scanning.* Structured light (SL) scanning has a rich history, as reviewed by [Zhang 2018]. Early systems, such as [Posdamer and Altschuler 1982], introduced the idea of projecting  $m$ -bit stripe sequences to encode  $2^m$  patterns. This idea evolved through Gray codes [Hall-Holt and Rusinkiewicz 2001; Inokuchi et al. 1984], linear stripe patterns [Carrihill and Hummel 1985], RGB and color-encoded fringes [Boyer and Kak 1987; Caspi et al. 1998b; Je et al. 2012], phase-shift techniques [Gupta and Nayar 2012; Wust and Capson 1991], and space-filling curves [Gupta and Nakhate 2018]. Although effective for static scenes, these methods face limitations in dynamic, high-framerate settings. Most require projecting and decoding multiple time-varying patterns to establish correspondences for triangulation, which fundamentally restricts temporal resolution. Our system, in contrast, bypasses triangulation and captures depth from as few as one single projected pattern plus a white reference, enabling significantly faster acquisition via a beam-splitter-based optical setup. SL has also seen widespread deployment in commercial products, for example, the original Microsoft Kinect [Kinect 2023], based on PrimeSense technology [PrimeSense 2023]. In our experiments, we compare with three commercial depth sensors capable of dynamic scanning: Microsoft Kinect, Intel RealSense, and Photoneo MotionCam. We note that these sensors provide visibly worse reconstructions of simple dynamic scenes, such as a fan spinning at 5 Hz.

*High-Speed SL Scanning.* Numerous efforts have been made to improve frame rates of SL techniques through hardware modifications. These include adaptations to DLP projectors [Koppal et al. 2012; Zhang et al. 2010], stereo-projector hybrid systems [Hu et al. 2021;

Jones et al. 2006, 2009; TABATA et al. 2016], and event-based cameras [Cossairt et al. 2015; Huang et al. 2021]. However, most of these approaches demonstrate limited dynamic reconstruction, achieving up to 120 fps with a single projector, and often not accurate enough to recover physical properties from the scene.

Their performance is further constrained by the limitations of DLP projectors, which support very high refresh rates only in binary mode. Although DLP hardware can operate at kilohertz speeds, these rates are typically not available for 8-bit or 24-bit color projections, as manufacturers optimize for perceptual frame rates rather than machine vision. In contrast, our proposed method relies only on the ability to repeatedly project a fixed pattern without explicitly modeling or calibrating the projector.

*LUT-Based Calibration.* Lookup tables (LUTs) have been used in SL systems primarily as a tool for calibration rather than reconstruction. [Luo et al. 2014] constructs per-pixel LUTs by sweeping a linear stage and mapping phase values to depth for phase-shifted patterns. Similar pixel-wise LUTs are employed in telecentric SL systems [Chen et al. 2022] to model nonlinear phase-depth mappings with high precision. In underwater imaging, LUT calibration methods account for refraction across glass-water interfaces [Zoraja et al. 2023]. In contrast to these approaches, *LookUp3D* treats LUT not as an intermediate calibration aid, but as the primary mechanism for depth inference, enabling per-ray reconstruction without explicit projector calibration or projector-camera triangulation.

*Summary.* SL scanning methods use calibration to fit an idealized projector model: while compact, the fitted model will unavoidably introduce errors. To the best of our knowledge, ours is the first SL approach sidestepping modeling the projector, replacing it with a purely data-driven LUT.

## 3 LOOKUP3D SCANNING

Our scanning method relies on a per-pixel lookup table (LUT)  $C_i(d)$  from scene depth  $d$  to observed color values at camera pixel  $i$ . Once this unique mapping is established during calibration, depth can be inferred at runtime by value lookup (Section 3.1). We acquire each  $C_i$  using a linear stage and a flat calibration board. By sweeping the stage through a volume while projecting a fixed color pattern, we record the color response at each depth for every pixel, building a per-pixel LUT from depth to color (Section 3.2). Fig. 2 summarizes the full calibration and reconstruction pipeline, and we provide additional technical details in Section 3.3.

### 3.1 Reconstruction

*LookUp3D* reconstruction method relies on decoding depth from a per-pixel pre-recorded lookup table (LUT). We define  $C_i(d) : \mathbb{R}^+ \rightarrow \mathbb{R}^3$ , the mapping from depth  $d$  to color for a given pixel  $i$ . For reconstruction to work without confusion, the calibrated mapping should be unique. The scanning procedure is as follows: we project a color pattern onto an object of unknown geometry and capture an image of intensity  $I$ . Reconstruction compares each pixel in the measured color image  $I$  with a per-pixel LUT. In its unoptimized form, the operation is a simple, embarrassingly parallel lookup to



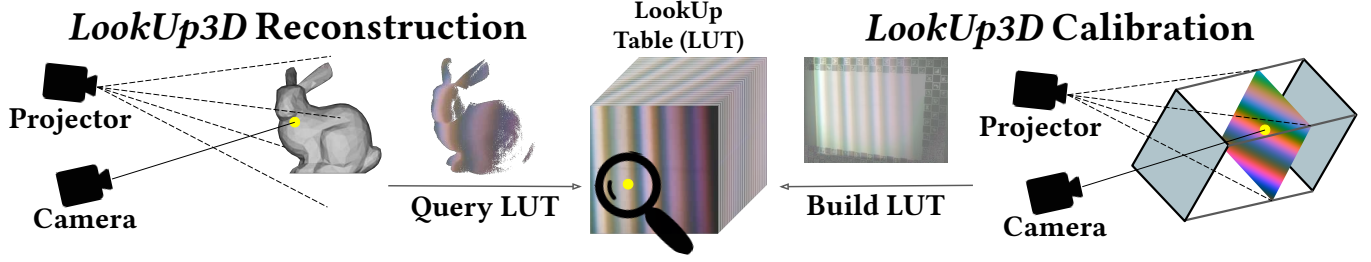


Fig. 2. **Method Overview.** Our pipeline consists of two stages: *LookUp3D Calibration* and *LookUp3D Reconstruction*. During calibration, we sweep a planar target with a linear stage in fine depth increments. At each step, we use normalized pattern image against a white flash to produce per-pixel RGB observations. These are assembled into a per-pixel lookup table (LUT) that encodes the color-to-depth mapping. Once calibrated, we scan unknown geometry by capturing the same pattern and white flash, normalizing them, and querying the LUT to assign depths via nearest-neighbor search in Euclidean distance.

find the depth whose color is closest to the observed pixel  $I_i$ , or

$$d_i^* = \arg \min_{d \in \mathcal{D}} \|C_i(d) - I_i\|_2, \quad (1)$$

where the search is done offline on a discrete set of depths  $\mathcal{D}$ . The residual  $r_i = \|C_i(d_i^*) - I_i\|_2$  provides a per-pixel confidence measure that can be thresholded to filter unreliable points.

### 3.2 Calibration

The goal of calibration is to create the unique dictionary  $C_i(d)$  between depth and color for each pixel  $i$ . Our calibration procedure, then, uses a planar calibration target moved by an off-the-shelf linear stage [Xiao et al. 2015] and a fixed pattern projected onto the calibration target. We record, per-pixel, the color reading and the depth measurement. First, in order to accurately retrieve a depth measurement from a camera pixel, we calibrate the intrinsic camera parameters following [Lanman and Taubin 2009a]. Then, we discretize the frustum in front of the camera as a collection of rays  $\mathbf{r}_i(d) : \mathbb{R}^+ \rightarrow \mathbb{R}^3$ , where  $\mathbf{r}$  departs from camera’s origin and passes through pixel  $i$ , scaled by a positive depth  $d$ . At this point, we deviate from traditional structured light: to recover depth, we rely solely on the calibration board and the linear stage, without the need to model the projector defocusing, vignetting, intrinsic and extrinsic parameters. We show in Section 3.4 a set-up where the quality of the optics of the projector does not alter the results achieved by *LookUp3D*, whereas a SL method reliant on triangulation suffers a lot in quality of reconstruction.

### 3.3 Technical Details

*LookUp3D* is a data-driven method: both calibration and reconstruction rely exclusively on RGB measurements acquired from the camera sensor. While the core idea of lookup scanning is conceptually straightforward, achieving robustly accurate performance in static setting and adapting to dynamic setting require careful design. In this section, we detail the key implementation details and algorithmic strategies employed to improve the quality of the calibrated LUT and of the reconstructed 3D geometry.

**Normalization.** Throughout, rather than work with raw readings, which are susceptible to varying albedo, lens vignetting, object texture, and other nuisance factors, we use normalized color [Caspi

et al. 1998b]. To normalize our readings, we require one image acquired with the color pattern ( $I_P$ ), one image of white flash ( $I_W$ ), and one of ambient capture ( $I_B$ ). The normalized color image is  $I = \frac{I_P - I_B}{I_W - I_B}$ . In dynamic, we forgo the ambient capture. This leads to a frame rate of half of projector and camera in the dynamic setting. We note that this is not equivalent to the superior option of normalizing to an ambient capture, which we use in static case.

**Linear Stage Trajectory Fitting.** During calibration, we detect the markers on the calibration board for every linear stage step, which allows us to register the camera in relation to the board plane. However, such registration is not entirely reliable due to marker detection errors and we benefit from incorporating the known, precise motion of the linear stage. We thus fit a linear trajectory to the calibration board motion using the stride size to acquire a monotonic depth measurement for each pixel. We set up our calibration for 750 linear stage steps of 0.2 mm, which in turn limits the depth precision per pixel at 0.2 mm. The calibration with these settings takes less than 10 minutes and requires no user interaction.

**Denosing.** In low-noise settings, storing the normalized color readings into the LUT is sufficient, but when noise is significant (e.g., short exposures, low-quality camera sensors), denoising becomes a useful tool to improve depth decoding. In our high-dynamic case, where our method is more susceptible to sensor noise, we denoised the LUT via rolling kernel, Fourier transform, low-rank approximation, and spline fitting (refer to Section G.1 for details). In our dynamic experiments, LUT storage is around 7GB uncompressed; with a low-rank approximation the storage reduces to 152MB.

**Residual Thresholding.** The residual measures the consistency between the captured color and what was observed and stored in the LUT during calibration. Although a low residual does not guarantee correctness (e.g., due to LUT aliasing or local ambiguities), a high residual strongly indicates reconstruction failure (see Section 5). Residual thresholding can be used to filter unreliable depth estimates, which is particularly beneficial for downstream tasks requiring high precision, such as physical property inference.

**Restricting Depth Range.** In our static set-up, a naive reconstruction with high channel count yields low-error results. With lower pattern count in both static and dynamic set-ups, however, we can

opt for a coarse-to-fine (C2F) approach to effectively add neighboring information as we decode depth (Fig. 25). C2F helps disambiguate potential confusion from finding the minimum residual in a smaller depth range (refer to Section G.2). In dynamic scenes, we have additional access to the previous depth as a starting point for the lookup search; we see in practice that both C2F and temporal consistency (TC, refer to Section G.3) help disambiguate candidate depths in challenging dynamic cases (Fig. 6).

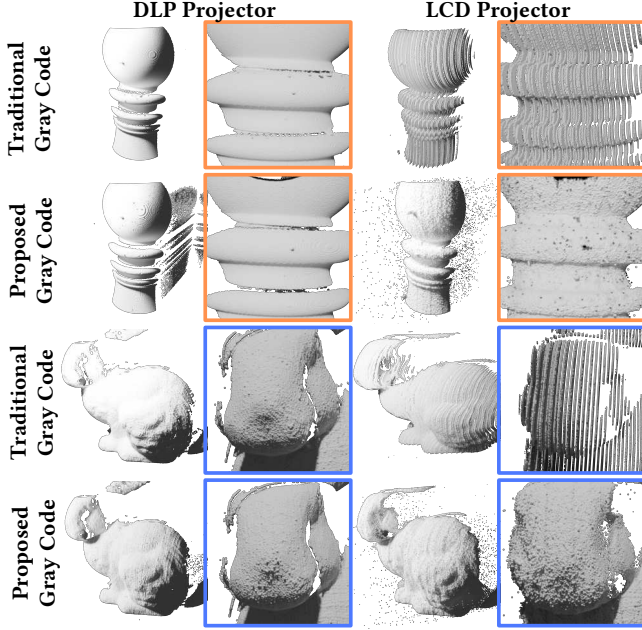


Fig. 3. **Static Reconstructions with Different Hardware.** We reconstruct a pawn and a bunny using a high-end DLP projector and an inexpensive LCD projector, both paired with the same camera. Traditional SL reconstruction with 44 Gray Codes deteriorates significantly with the low-quality projector due to a reliance on accurate triangulation, whereas *LookUp3D* with 11 Gray Code channels remains robust.

### 3.4 Static Validation

We first validate *LookUp3D* in a static setting, where we minimize sources of noise and can more easily compare to traditional SL methods. We explore different hardware configurations, pattern combinations, and ambient light conditions, conducting careful quantitative evaluations against ground-truth geometries that are otherwise difficult to obtain in a dynamic setting.

We conduct validation using two hardware setups, both sharing the same camera – a Lucid Vision Labs Atlas 31.4MP with an Edmund Optics APS-C 50mm lens – but paired with different projectors. The first is a high-quality Texas Instruments DLP4710EVM-LC projector. The second is a low-cost LCD projector (Mini Projector VOPLS 1080P, approximately 50USD) intended for home entertainment with inexpensive optics. We control all other settings and compare the reconstructions across three test objects: the Stanford bunny (approx. 10cm  $\times$  12cm  $\times$  6cm), a precisely machined pawn (approx.

Table 1. Comparison of our proposed *LookUp3D* (*LU3D*) with traditional structured light. We report the root mean squared error (RMSE) in mm of each reconstructed point cloud with respect to the ground-truth geometry.

Method	# Chan.	DLP Projector			LCD Projector		
		Bunny	Pawn	Plane	Bunny	Pawn	Plane
Trad.	44	0.96	1.12	0.17	47.07	67.07	1.81
<i>LU3D</i>	3	1.46	1.86	0.42	3.02	1.42	0.42
<i>LU3D</i>	6	1.12	0.73	0.17	2.31	0.86	0.38
<i>LU3D</i>	9	1.03	0.34	0.13	2.01	0.63	0.34

10cm  $\times$  16cm  $\times$  10cm), and a flat plane (40cm  $\times$  30cm). For the pawn, we use its CAD model as ground truth; for the plane, we assume ideal flatness; and for the bunny, we obtain a reference mesh using a traditional SL scanner and a rotating stage.

We evaluate *LookUp3D* using 11, 9, 6, and 3 channels and compare it to traditional structured light using Gray codes (with 44 binary patterns) with both projectors. We report the median absolute error in Table 1 and visually evaluate the reconstructions (Fig. 3).

The results show that *LookUp3D* achieves reconstruction quality comparable to traditional SL with the DLP projector, and significantly outperforms it with the LCD projector. While the quality of traditional Gray Code reconstruction deteriorates significantly with the low-quality projector – due to its reliance on accurate triangulation – *LookUp3D* remains robust. We calibrated both projectors using the standard method in [Moreno and Taubin 2012], but the calibration fails to sufficiently model the optical distortions in the inexpensive LCD projector. In contrast, *LookUp3D* implicitly embeds projector intrinsics into the LUT, so reconstruction accuracy is largely unaffected by projector quality. This insight motivates our design of the high-speed projector prototype (Section 4.1): decoupling performance from the ability to model the projector optics enables us to build a simple analog projector with LED components.

Additionally, *LookUp3D* reconstruction error decreases with more pattern channels for both projectors. This validates the intuition that ambiguity in depth-to-color mapping decreases with increasing channel count. These results suggest that a 3-channel LUT is sufficiently accurate for our high-speed setting, where we trade off some accuracy for fewer patterns and higher fps. We defer an in-depth analysis of pattern design to prior work [Caspi et al. 1998a; Gupta and Nakhate 2018; Gupta and Nayar 2012; Mirdehghan et al. 2018] and our Supplemental Materials (Section E).

We also reconstruct, with the DLP projector and *LookUp3D*, four challenging objects (Fig. 4): a flat color calibration board (top left), a precisely milled irregular structure, with curves and steps (top right), a mug (bottom left), and a fossil replica (bottom right). For the mug, we recovered a cylindrical diameter of 83.42 mm (versus 83.00 mm measured with a caliper) and RMSE of 0.68 mm; for the milled irregular structure, we got RMSE 1.35 mm from its CAD design. For all four scenes, we also zoom into different parts of the reconstructed clouds to showcase that *LookUp3D* can capture those details like varying textures, varying colors, and difficult shapes.

Finally, we explore how *LookUp3D* performs under different lighting conditions. We calibrate an 11-channel LUT in a darkroom, but

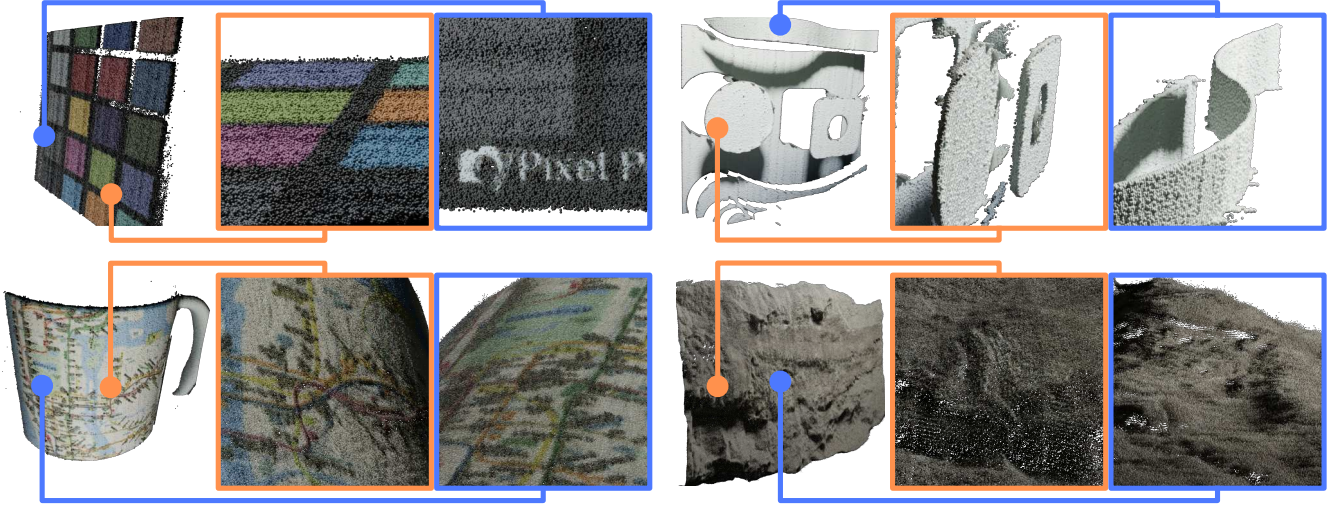


Fig. 4. **Mosaic of Static Scenes.** These scenes were captured with our DLP Projector and Atlas Camera in a static setting with controlled lighting to showcase the quality of reconstructions we can achieve with *LookUp3D*. We show four scenes reconstructed with 9 channels: top left is a color calibration board; top right is a precisely-machined irregular structure; bottom left is a mug; bottom right is a replica of a fish fossil. Each reconstructed cloud is displayed with two zoom-ins of different regions to highlight different shapes, textures, and albedo.

capture the objects with the ceiling lights on; the reconstructed objects maintain its accuracy even at a different lighting condition (Fig. 5). The house is 3D printed, therefore we compare our reconstruction against the ground truth geometry and obtain median



Fig. 5. **Mosaic of Static Scenes with Ceiling Lights On.** These static scenes were captured with our DLP Projector with ceiling lights on. Even though the lookup table was calibrated in a darkroom, *LookUp3D* is able to accurately reconstruct the objects. We show a chess board, a paint brush, a 3D-printed house, and a monkey statue, all reconstructed with 11 channels.

absolute error of 1.57 mm and RMSE of 2.46 mm. We provide the PLY files of twenty reconstructions with our Supplemental Materials.

The design of *LookUp3D* offers robustness and flexibility but introduces trade-offs. Building the lookup table requires a motorized linear stage for calibration, sufficient storage for the table itself, and enough RAM to support runtime lookup operations. However, these are one-time or amortized costs and do not impact the frame rate or simplicity of the scanning process during acquisition.

## 4 HIGH-SPEED 3D SCANNING

*LookUp3D* opens the possibility of achieving high-resolution 3D scanning at high speeds. In this section, we review the hardware choices we made and present a prototype to scan 3D geometry at high speeds (Section 4.1). We also evaluate our method against controlled transformations of objects to which we have ground truth geometries acquired in a static setting with a traditional structured light scan (not *LookUp3D*) and extract physical properties, such as speed and gravitational acceleration, from the dynamic reconstructions (Section 4.2). After establishing the accuracy of *LookUp3D* and our scanning prototype at high-speeds, we explore the new regime of 3D geometry scanning that *LookUp3D* enables. We highlight our capabilities with a handful of high-resolution, high-accuracy scenes reconstructed at 100 fps, 450 fps, and 1,450 fps (Section 4.3). Finally, we compare *LookUp3D* side-by-side against three commercially-available dynamic 3D scanners: Photoneo MotionCAM 3D M+, Intel Realsense D455, and Microsoft Azure Kinect (Section 4.4).

### 4.1 Prototype Hardware

*LookUp3D*'s performance is bounded by the camera's resolution and frame rate, the projector's synchronized refresh rate, linear stage stride, and signal-to-noise-ratio permitted by the scene light.



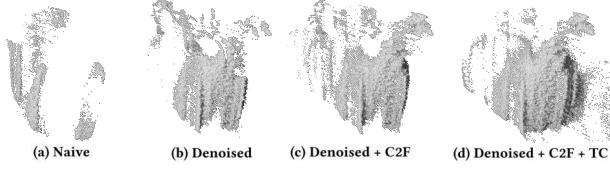


Fig. 6. **Adapting Reconstruction from Static to Dynamic Setting.** We illustrate key algorithmic improvements that enable reconstruction in the high-speed setting, demonstrated on a water balloon impact sequence. (a) The static reconstruction pipeline fails to recover many points with residual below 0.2. (b) LUT denoising via low-rank approximation improves coverage but remains incomplete. (c) Coarse-to-fine (C2F) search helps recover locally consistent depths, further improving the completeness. (d) Adding temporal consistency (TC) leverages frame-to-frame coherence and helps reconstructing most of the object under challenging deformations.

We built a prototype with as few constraints as possible to explore *LookUp3D*’s capabilities with dynamic scenes.

We chose the high-end Chronos HD camera, which achieves 1k fps at 2 Megapixel resolution and 3k fps at 0.6 Megapixel, paired with a Sigma 35mm f1.4 lens for our high-speed scanner. Unfortunately, a similarly fast and sufficiently accurate projector is not readily available on the market. Commercially available DLP projectors, for instance, achieve high frame rates by trading off bit depth: the TI DLP LightCrafter 6500 projects 24-bit data at 500Hz, 8-bit at 1400Hz, 1-bit at 9000Hz. To achieve high-speed capture, we require alternating between a RGB pattern (usually a 24-bit image) and a white flash. We propose a radically different hardware solution: an analog projector that can project the two alternating patterns at over 1k fps and at high-resolution.

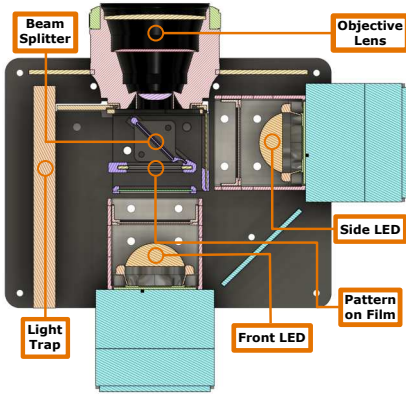


Fig. 7. **Analog Projector Diagram**

Our prototype analog projector (Fig. 7) comprises two 100 Watt LEDs perpendicular to each other. A beam splitter, placed at a 45-degree angle, ensures the flashes of light occur in the same frustum in front of the projector. The projector is also equipped with a Sigma 50mm f1.4 lens. A 35-mm color film slide with the pattern exposed onto it is placed in front of one of the LEDs, which removes limits on bit resolution for the RGB pattern.

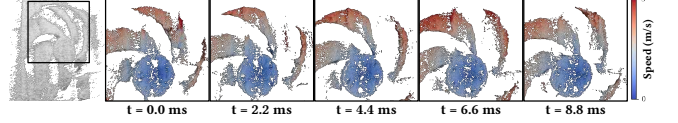


Fig. 8. **Fan Spinning Counterclockwise at 900 RPM.** We used our high-speed prototype with *LookUp3D* to reconstruct, at 450 fps, a fan. We focus on a zoomed-in part of the reconstructed fan and use a colormap to paint the speed (m/s) of each point in the cloud.

The projector house is made of laser-cut 3 mm-thick acrylic pieces. The holder for LEDs and the beam splitter were 3D printed with PLA. The lens holder was milled from a L-shaped piece of aluminum. Additionally, to ensure synchronicity between camera triggering and turning on and off LEDs at high frame rates, we use a Texas Instrument TIVA C Series TM4C123G ARM Cortex-M4F Launch Pad micro-controller; we can thus precisely switch between the color pattern and white flash at frequencies above 1 kHz. We are making the CAD design with all the individual pieces open source for the benefit of the community and to ensure reproducibility.

The only challenging part to reproduce in our setup is the film: we expose the film with a calibrated Full HD projector and a Canon EOS-1N analog camera with a Tamron 90mm f/2.8 lens. We document the procedure for bringing patterns onto film in the Supplemental Material (Section F), and have stored 35mm film slides for shipping to labs interested in reproducing our setup.

## 4.2 Quantitative Analysis

In this section, we test the accuracy and precision of *LookUp3D* by reconstructing and compare its 3D geometry against known geometries undergoing rigid transformations. We also test if *LookUp3D* is accurate enough to extract physical properties of dynamic scenes, since we think that a main potential application of *LookUp3D* is to reconstruct and retrieve physical properties of objects undergoing rigid and/or soft transformations. This is applicable to robotics applications and to validating physical simulation models.

**Bunny Drop.** We drop a bunny onto a hard surface and reconstruct its impact at 450 fps (Fig. 1). More than capturing the shock deformation and oscillations of the bunny’s ears, we can measure gravity from this sequence. We extract the centroid of 20 consecutive point clouds and fit a second degree polynomial to its displacement over time: we find a gravitational acceleration of  $9.60 \text{ m/s}^2$ , a 2% relative error from the nominal value of  $9.80 \text{ m/s}^2$ .

**Bunny on Linear Stage.** We then place the bunny on a platform and attach it to our linear stage. We set the linear stage to move 31.25 mm and we reconstruct the bunny’s motion at 450 fps (Fig. 28). With *LookUp3D*, we recovered a total distance traveled of 31.52 mm, 0.9% relative error from the ground truth, and a speed of 12.56 mm/s, whereas the linear stage manufacturer reports 12.50 mm/s. To further evaluate the accuracy of reconstructing such geometry moving, we compare the error of our reconstructed bunny from the translated ground truth mesh (from Section 3.4) and obtain a root mean squared error of 3.27 mm for 1, 200 frames.

Table 2. We drop a bouncy ball on multiple materials and reconstruct the scene with *LookUp3D*. We attempt to find the radius ( $R$ ) of the ball and the gravitational acceleration ( $g$ ) with our point clouds. We also calculate the coefficient of restitution  $e$  to evaluate how elastic the collision is.

	Ball Drop Experiment				
	Acrylic	Aluminium	Paper	Sand Block	Wood
$R$ (cm)	33.16	26.47	32.65	34.30	34.38
$g$ (m/s <sup>2</sup> )	10.99	9.57	11.03	11.90	10.28
$e$	0.48	0.58	0.23	0.50	0.52

**Fan Spinning.** We reconstruct a fan spinning at 900 RPM and find point-to-point correspondences between adjacent frames using Iterative Closest Point. From these correspondences, we calculate the instantaneous speed for each point and display the reconstruction with a colormap in Fig. 8. The furthest points on the blade are 0.05 meters away from the center – in circular motion at 900 RPM, the speed at that point would be 4.7 m/s. We extract speeds between 4.0 to 5.0 m/s for the furthest points from the center.

**Ball Bouncing.** We grab a bouncy ball and perform drops on multiple materials to test collision elasticity. We reconstruct the drop and subsequent bounces of the ball to acquire coefficients of restitution for the different materials (Table 2). Since we measured the radius of the ball with a caliper (31.00 mm), we also fit a sphere to the reconstructed model to verify the accuracy of our results. We track the sphere’s center over time to find the velocity and gravitational acceleration (for which the nominal value is 9.80 m/s<sup>2</sup>) of the scene. For the coefficient of restitution, we calculate  $e = \left| \frac{v_i}{v_f} \right|$ , where  $v_i$  and  $v_f$  are the the velocities *before* and *after* a collision, respectively.

This opens the opportunity to integrate *LookUp3D* into applications in robotics and physical simulation models, where one can use the reconstructed point clouds to estimate physical properties.

### 4.3 Qualitative Results

We have established, through our quantitative results, that *LookUp3D* is able to capture accurate motion at high frame rates. Here we highlight a mosaic of dynamic scenes (Fig. 10 at 100 fps and Fig. 13 at 450 fps) captured with our prototype and reconstructed with *LookUp3D*. Finally, we push our system to record a water balloon bouncing at 1,450 fps and 0.4 Megapixel (Fig. 14) – even in a controlled lighting environment, sensor noise is really noticeable with camera exposure at 0.3 ms and *LookUp3D*’s performance decreases. We also stress that, due to the high speed in capture and the deformations of the materials in the scenes, there is no ground truth available and we do not report any error metric for these reconstructions.

### 4.4 Commercial Comparisons

In addition to the quantitative evaluations, we compare our high-speed prototype against three commercial dynamic depth sensing devices: Photoneo MotionCAM 3D M+, Intel Realsense D455, and Microsoft Azure Kinect. These scanners cannot be used simultaneously as they all actively emit light; we thus repeat each scene for each scanner, striving to keep the initial conditions as similar as

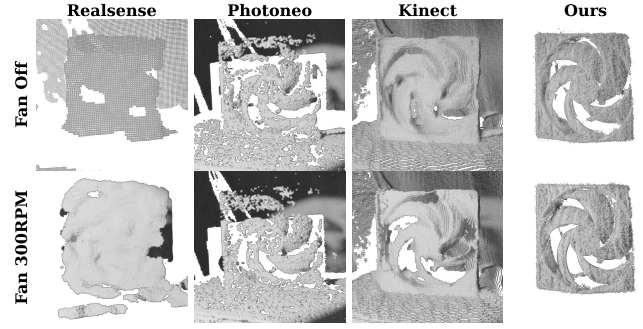


Fig. 9. **Comparison of Reconstructing a Fan.** Top: an inactive fan captured by three commercial devices (Realsense, Photoneo, Kinect) and our high-speed scanning prototype. Bottom: the fan spinning at 300 RPM. The video in the Supplemental Materials shows a video of each system.

possible. We note that there is a discrepancy in the frame rate and resolution of the scanners: 8 fps/0.9MP Photoneo; 90 fps/0.04MP RealSense; 30 fps/0.4MP Azure Kinect; and 450 fps/1MP for ours. For better comparison, we first use the devices to capture a fan turned off (top row of Fig. 9). Then, we turn on the fan to 300 RPM and visually evaluate each scanner’s performance (bottom row of Fig. 9). The *RealSense* device we tested makes use of stereo-matching – it is equipped with two infrared cameras and it projects an infrared pattern to assist in feature matching across the two different views. It has the highest frame rate (90 fps), but a low resolution point cloud. *Photoneo* makes use of parallel structured light with a proprietary CMOS sensor. It offers the highest resolution, but at a very low frame rate (8 fps). The *Kinect*, equipped with a Time-of-Flight camera, strikes a balance between the previous two scanners in terms of scanning speed (30 fps) and resolution. Our scanner, in contrast, compromises on neither scanning speed nor resolution.

## 5 CONCLUDING REMARKS

We introduced a new approach for structured light 3D scanning that does not require modeling or calibrating the projector, and demonstrated its applicability for both static and dynamic scenes.

**Limitations.** Since we rely on light diffusely scattering off of the calibration board, our system cannot resolve inter-reflections, translucent objects, highly specular or reflective surfaces, and shadow regions (Fig. 11), similarly to other structured light methods. Additionally, *LookUp3D* reconstructions are inherently restricted to the calibrated depth range, as there is no direct way to extrapolate the data to regions outside of the volume. We also attempted to use our high-speed prototype with *LookUp3D* to reconstruct a balloon popping (Fig. 31). Unfortunately, even at 1450 fps, the motion of the explosion is too abrupt: since we acquire two images (a pattern projection and a white flash), if these two images do not overlap much, *LookUp3D* will most likely fail to recover any relevant depth. Finally, our projectors have limited light power and our method will not work if the flux of ambient light is much higher than that of the projector (e.g., outdoors). For more details on ambient light in our dynamic set-up, refer to our Supplemental (Section B).



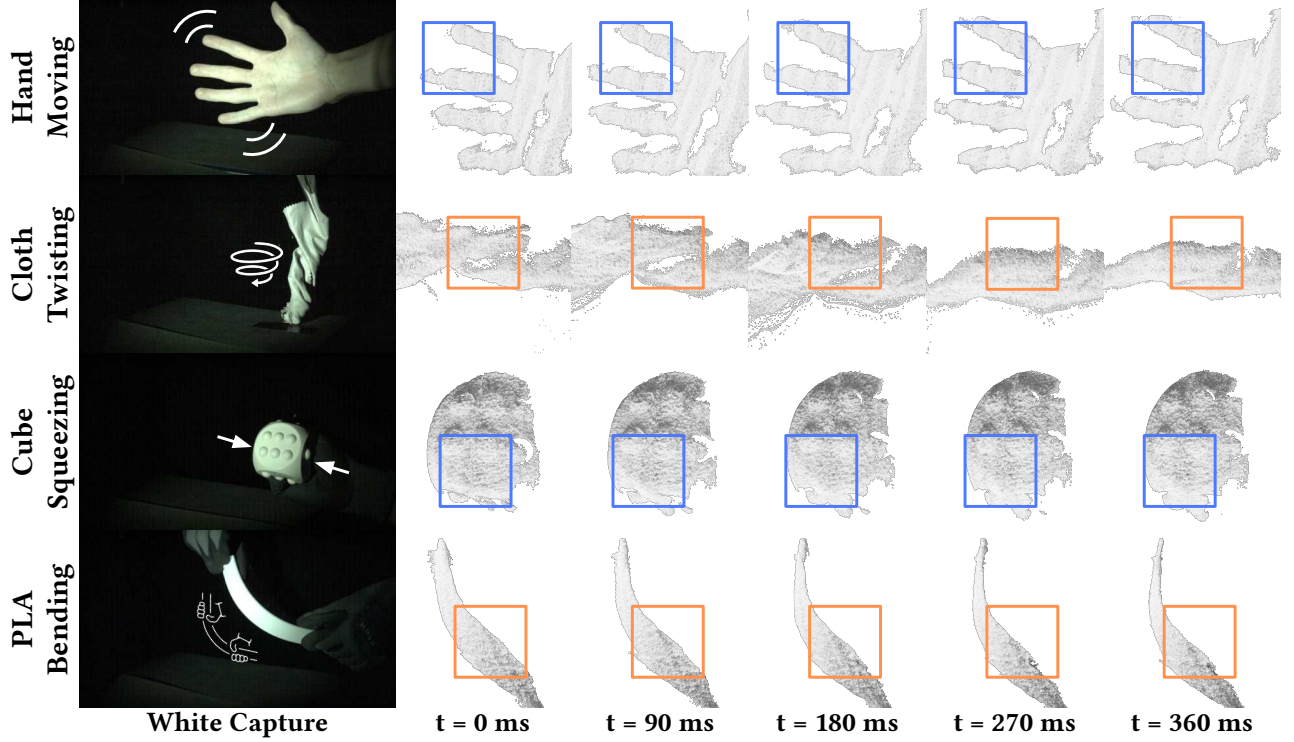


Fig. 10. **Dynamic sequences for different scenes at 100 fps.** The exposure for each captured frame was 4.5 milliseconds. For the white capture images, we synthetically increase gain and contrast, otherwise the captures are too dark to be properly displayed. We also add boxes on the reconstructed point clouds to aid visualization of subtle motion. The first row is a hand moving; the second row is a piece of cloth being twisted; the third row is a silicone cube being squeezed; the fourth and last row is a 3D-printed piece being bent from its edges. We have provided clips of the scenes in our accompanying video.

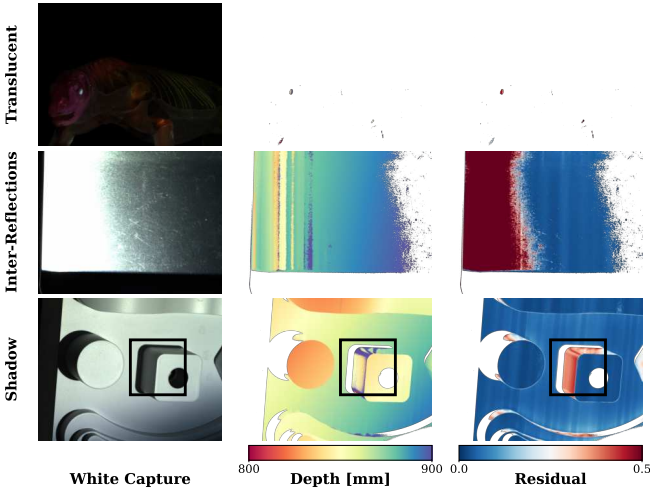


Fig. 11. **Limitations.** *LookUp3D* is designed to record light that diffusely scatters from a calibration board. When light refracts through a translucent medium, our method has no chance of working. Similarly, we cannot decode overexposed pixels or inter-reflections. In the bottom row, the square highlights a shadowed region where decoding fails.

We also notice that wrong depths align well with high residuals, showing their utility as an indicator of reconstruction confidence. The ease of calibration of the system might allow for some creative solutions for these challenging cases, such as adding multiplexed light sources and calibrating with a specular pattern at different orientations, which we believe are exciting avenues for future work.

Another limitation is that our reconstructions are currently done offline, whereas the commercial devices provide real-time data. Our current unoptimized implementation takes 2 seconds per frame on a CPU (1 core of an AMD EPYC 9654P Processor clocked at 3.7 GHz) and 0.1 second per frame on an NVIDIA RTX A6000 GPU.

*Future Work.* Increasing the LED power (and thus the amount of light available) from our analog projector would allow us to achieve even higher frame rates: we currently cannot further increase the power output as our cooling system is unable to dissipate it, leading to a melted film. We believe that the simplicity offered by our approach (a fixed image projector) will foster additional hardware research in this direction, overcoming this limitation, and is a much easier proposition than developing high frame rate projectors for standard SL pattern stacks. A straightforward, but challenging from an engineering standpoint, way to increase power would be using a larger film, but other options such as printing on or etching a glass slide might lead to a more compact projector. If this limitation is overcome, we expect to acquire depth data of similar quality as

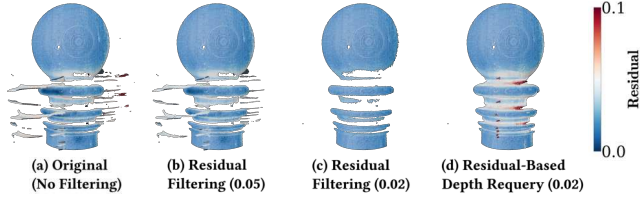


Fig. 12. **Residual Filtering and Depth Requery.** (a) The original pawn result has artifacts around its neck, caused by heavy shadows, specularities, and inter-reflections. (b) Filtering at a threshold of 0.05 removes many wrong depths. (c) A stricter threshold of 0.02 removes noticeable outliers but yields a less complete point cloud. (d) Using the inliers from (c) to restrict the search range leads to a more complete and accurate reconstruction.

our static scans but for dynamic scenes. We note that there is only one difference between our static and dynamic results, the exposure time of the projector and camera, which directly affects sensor noise and reconstruction errors.

The possibility of using arbitrary patterns with our *LookUp3D* reconstruction algorithm and the removal of the necessity to calibrate the projector opens many interesting avenues for future work: (1) it is possible to use multiple synchronized projectors and cameras to increase scanning view without worrying about interference (as long as the projectors do not overexpose), (2) the use of multiple analog projectors multiplexed in time to reduce noise at the cost of lower temporal resolution, (3) the use of invisible light patterns (infrared) potentially combined with visible light.

Another promising direction lies on the algorithmic side, where we can take advantage of two key properties of the existing system: the availability of per-pixel residuals as a natural byproduct of the reconstruction process, and the structured, differentiable nature of the underlying depth-to-color mapping. Residual maps not only serve as confidence indicators but also enable refinement on depths. In a preliminary experiment on the static pawn scene using *LookUp3D* with only three channels, shown in Fig. 12, we identify outliers via residual thresholding and requery their depths using a narrowed search range guided by depths from nearby inliers. Without additional input or parameter tuning, we significantly improve the reconstruction accuracy in the same challenging regions highlighted in our failure cases – such as specularities, inter-reflections, and heavy shadows, reducing the root mean squared error (RMSE) from 6.39mm to 1.52mm. We believe residual-based refinement holds strong potential for further improving robustness and accuracy, especially in dynamic or noisy settings. Building on this, a second direction involves embedding the lookup into an optimization framework. In particular, using residuals to guide (stochastic) gradient descent for depth refinement could lead to more accurate and stable reconstructions.

*Reproducibility.* To foster reproducibility and adoption of this new technique, which we believe offers a different and exciting perspective on structured light scanning, we will release an open source implementation<sup>1</sup> of our algorithm and an open hardware specification for our high speed projector.

<sup>1</sup><https://github.com/geometryprocessing/scanner>

## ACKNOWLEDGMENTS

This work was partially supported by the the NSF grants OAC-2411349 and OAC-2411221. Giancarlo Pereira was partially supported by the New York University Tandon School of Engineering Fellowship. We thank NYU IT High Performance Computing services, for help with resources, services, and expertise. We thank Christopher Musco for fruitful discussions regarding low-rank approximations.

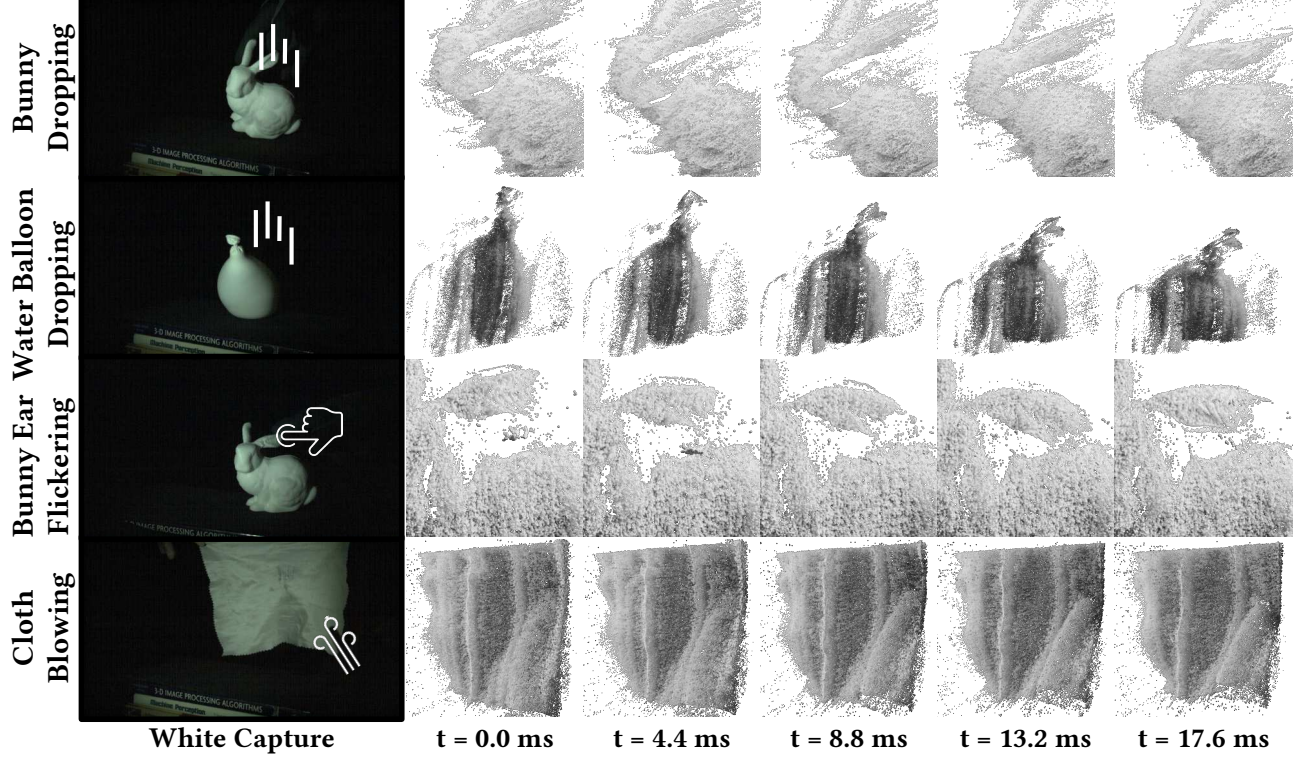


Fig. 13. **Dynamic sequences for different scenes at 450 fps.** The sensor exposure was set to 1 millisecond, which decreases signal-to-noise ratio and causes more sporadic noise in the reconstructed point clouds. For the white capture images displayed above, we synthetically increase gain and contrast, otherwise the captures are too dark to be properly displayed. The first row is a silicone bunny in free fall hitting a hard surface; the second row is a water balloon in free fall also hitting a hard surface; third row is the silicone bunny’s ear wiggling at high frequency; the fourth and last row is a cloth hanging while air blows on it. We have provided clips of the full scenes in our accompanying video.

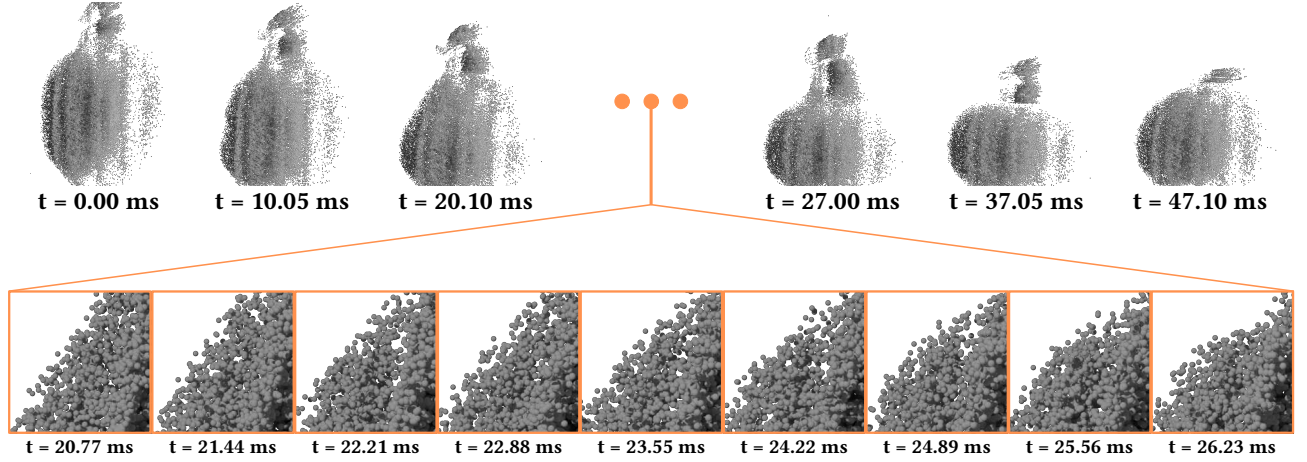


Fig. 14. **Sequence at 1450 fps.** We highlight the motion of a water balloon as it hits a hard surface in less than one twentieth of a second. To achieve 1450 fps, we decrease the camera resolution to 0.4Megapixel and expose each frame for 0.3 millisecond. Sensor noise is very pronounced at this exposure and decreases the points we can confidently reconstruct. The bottom row are zoom-ins of consecutive frames as shock wave propagates to the top of the water balloon. This exemplifies that *LookUp3D* captures minute details in very small time steps. We have provided a clip of the scene in our accompanying video.

## REFERENCES

- Seyed Ehsan Marjani Bajestani and Giovanni Beltrame. 2023. Event-Based RGB Sensing With Structured Light. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 5458–5467.
- K. L. Boyer and A. C. Kak. 1987. Color-Encoded Structured Light for Rapid Active Ranging. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-9, 1 (Jan. 1987), 14–28. <https://doi.org/10.1109/TPAMI.1987.4767869>
- Brian Carrhill and Robert Hummel. 1985. Experiments with the intensity ratio depth sensor. *Computer Vision, Graphics, and Image Processing* 32, 3 (Dec. 1985), 337–358. [https://doi.org/10.1016/0734-189X\(85\)90056-8](https://doi.org/10.1016/0734-189X(85)90056-8)
- D. Caspi, N. Kiryati, and J. Shamir. 1998a. Range imaging with adaptive color structured light. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 5 (1998), 470–480. <https://doi.org/10.1109/34.682177>
- D. Caspi, N. Kiryati, and J. Shamir. 1998b. Range imaging with adaptive color structured light. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 5 (May 1998), 470–480. <https://doi.org/10.1109/34.682177>
- Chao Chen, Ya Kong, Huaiwen Wang, and Zonghua Zhang. 2022. A High-Accuracy Calibration Method for a Telecentric Structured Light System. *Sensors* 22, 17 (Sept. 2022), 6370. <https://doi.org/10.3390/s22176370>
- O. Cossairt, N. Matsuda, and M. Gupta. 2015. Robust 3D Acquisition Using Motion Contrast 3D Scanning. In *Imaging and Applied Optics 2015 (COSI)*. OSA, CT2E.1. <https://doi.org/10.1364/cosi.2015.ct2e.1>
- Valentin Deschaintre, Yiming Lin, and Abhijeet Ghosh. 2021. Deep Polarization Imaging for 3D shape and SVBRDF Acquisition. <https://doi.org/10.48550/ARXIV.2105.02875>
- Marek Dobosz and Adam Woźniak. 2005. CMM touch trigger probes testing using a reference axis. *Precision Engineering* 29, 3 (July 2005), 281–289. <https://doi.org/10.1016/j.precisioneng.2004.11.008>
- Carl Eckart and Gale Young. 1936. The approximation of one matrix by another of lower rank. *Psychometrika* 1, 3 (1936), 211–218.
- Mohit Gupta and Nikhil Nakhate. 2018. A Geometric Perspective on Structured Light Coding. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part XVI* (Munich, Germany). Springer-Verlag, Berlin, Heidelberg, 90–107. [https://doi.org/10.1007/978-3-030-01270-0\\_6](https://doi.org/10.1007/978-3-030-01270-0_6)
- M. Gupta and S. K. Nayar. 2012. Micro Phase Shifting. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Providence, RI, 813–820. <https://doi.org/10.1109/CVPR.2012.6247753>
- O. Hall-Holt and S. Rusinkiewicz. 2001. Stripe boundary codes for real-time structured-light range scanning of moving objects. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, Vol. 2. IEEE Comput. Soc, Vancouver, BC, Canada, 359–366. <https://doi.org/10.1109/ICCV.2001.937648>
- Pengyu Hu, Shuming Yang, and Huiwen Zhang. 2021. High-speed and accurate 3D shape measurement using DIC-assisted phase matching and triple-scanning. *Optics and Lasers in Engineering* 147 (Dec. 2021), 106725. <https://doi.org/10.1016/j.optlaseng.2021.106725>
- Xueyan Huang, Yueyi Zhang, and Zhiwei Xiong. 2021. High-speed structured light based 3D scanning using an event camera. *Optics Express* 29, 22 (Oct. 2021), 35864. <https://doi.org/10.1364/oe.437944>
- S. Inokuchi, K. Sata, and F. Matsuda. 1984. Range imaging system for 3-D object recognition. *Proceedings of the International Conference on Pattern Recognition 1984* (1984), 806–808.
- Changsoo Je, Sang Wook Lee, and Rae-Hong Park. 2012. Colour-stripe permutation pattern for rapid structured-light range imaging. *Optics Communications* 285, 9 (May 2012), 2320–2331. <https://doi.org/10.1016/j.optcom.2012.01.025>
- Andrew Jones, Andrew Gardner, Mark Bolas, Ian McDowall, and Paul Debevec. 2006. Simulating Spatially Varying Lighting on a Live Performance. In *The 3rd European Conference on Visual Media Production (CVMP 2006) - Part of the 2nd Multimedia Conference 2006*. 127–133. <https://doi.org/10.1049/cp:20061934>
- Andrew Jones, Magnus Lang, Graham Fyffe, Xueming Yu, Jay Busch, Ian McDowall, Mark Bolas, and Paul Debevec. 2009. Achieving eye contact in a one-to-many 3D video teleconferencing system. *ACM Trans. Graph.* 28, 3, Article 64 (July 2009), 8 pages. <https://doi.org/10.1145/1531326.1531370>
- Microsoft Kinect. 2023. Kinect. <https://en.wikipedia.org/w/index.php?title=Kinect&oldid=1185848427> Page Version ID: 1185848427.
- S. J. Koppal, S. Yamazaki, and S. G. Narasimhan. 2012. Exploiting DLP Illumination Dithering for Reconstruction and Photography of High-speed Scenes. *International Journal of Computer Vision* 96, 1 (January 2012), 125 – 144.
- Douglas Lanman and Gabriel Taubin. 2009a. Build your own 3D scanner: 3D photography for beginners. In *ACM SIGGRAPH 2009 Courses*. ACM, New Orleans Louisiana, 1–94. <https://doi.org/10.1145/1667239.1667247>
- Douglas Lanman and Gabriel Taubin. 2009b. Build Your Own 3D Scanner: Optical Triangulation for Beginners. In *ACM SIGGRAPH ASIA 2009 Courses* (Yokohama, Japan) (SIGGRAPH ASIA '09). Association for Computing Machinery, New York, NY, USA, Article 2, 94 pages. <https://doi.org/10.1145/1665817.1665819>
- Huafen Luo, Jing Xu, Nguyen Hoa Binh, Shuntao Liu, Chi Zhang, and Ken Chen. 2014. A simple calibration procedure for structured light system. *Optics and Lasers in Engineering* 57 (2014), 6–12. <https://doi.org/10.1016/j.optlaseng.2014.01.010>
- Parsa Mirdehghan, Wenzheng Chen, and Kiriakos N. Kutulakos. 2018. Optimal Structured Light a la Carte. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Salt Lake City, UT, 6248–6257. <https://doi.org/10.1109/CVPR.2018.00654>
- Daniel Moreno and Gabriel Taubin. 2012. Simple, Accurate, and Robust Projector-Camera Calibration. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*. IEEE, Zurich, Switzerland, 464–471. <https://doi.org/10.1109/3DIMPVT.2012.77>
- Manasi Muglikar, Guillermo Gallego, and Davide Scaramuzza. 2021. ESL: Event-based Structured Light. In *2021 International Conference on 3D Vision (3DV)*. 1165–1174. <https://doi.org/10.1109/3DV53792.2021.00124>
- J.L. Posdamer and M.D. Altschuler. 1982. Surface measurement by space-encoded projected beam systems. *Computer Graphics and Image Processing* 18, 1 (Jan. 1982), 1–17. [https://doi.org/10.1016/0146-664X\(82\)90096-X](https://doi.org/10.1016/0146-664X(82)90096-X)
- PrimeSense. 2023. PrimeSense. <https://en.wikipedia.org/w/index.php?title=PrimeSense&oldid=1183798693> Page Version ID: 1183798693.
- Henri Rebecq, Guillermo Gallego, Elias Mueggler, and Davide Scaramuzza. 2018. EMVS: Event-Based Multi-View Stereo-3D Reconstruction with an Event Camera in Real-Time. *Int. J. Comput. Vision* 126, 12 (Dec. 2018), 1394–1414. <https://doi.org/10.1007/s11263-017-1050-6>
- Joaquim Salvi, Jordi Pagès, and Joan Batlle. 2004. Pattern codification strategies in structured light systems. *Pattern Recognition* 37, 4 (April 2004), 827–849. <https://doi.org/10.1016/j.patcog.2003.10.002>
- Satoshi TABATA, Shohei NOGUCHI, Yoshihiro WATANABE, and Masatoshi ISHIKAWA. 2016. High-speed 3D Sensing with Three-view Geometry Using a Segmented Pattern. *Transactions of the Society of Instrument and Control Engineers* 52, 3 (2016), 141–151. <https://doi.org/10.9746/sicet.52.141>
- Pingbo Tang, Daniel Huber, Burcu Akinci, Robert Lipman, and Alan Lytle. 2010. Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Automation in Construction* 19, 7 (Nov. 2010), 829–843. <https://doi.org/10.1016/j.autcon.2010.06.007>
- Sam Van der Jeught and Joris J.J. Dirckx. 2016. Real-time structured light profilometry: a review. *Optics and Lasers in Engineering* 87 (Dec. 2016), 18–31. <https://doi.org/10.1016/j.optlaseng.2016.01.011>
- Clarence Wust and David W. Capson. 1991. Surface profile measurement using color fringe projection. *Machine Vision and Applications* 4, 3 (June 1991), 193–203. <https://doi.org/10.1007/BF01230201>
- Lei Xiao, Felix Heide, Matthew O’Toole, Andreas Kolb, Matthias B. Hullin, Kyros Kutulakos, and Wolfgang Heidrich. 2015. Defocus deblurring and superresolution for time-of-flight depth cameras. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2376–2384. <https://doi.org/10.1109/CVPR.2015.7298851>
- Song Zhang. 2018. High-speed 3D shape measurement with structured light methods: A review. *Optics and Lasers in Engineering* 106 (July 2018), 119–131. <https://doi.org/10.1016/j.optlaseng.2018.02.017>
- Song Zhang, Daniel Van Der Weide, and James Oliver. 2010. Superfast phase-shifting method for 3-D shape measurement. *Optics Express* 18, 9 (April 2010), 9684. <https://doi.org/10.1364/OE.18.009684>
- Davor Zoraja, Tomislav Petković, Jean Forest, and Tomislav Pribanić. 2023. Calibration of a Structured Light Imaging System in Two-Layer Flat Refractive Geometry for Underwater Imaging. *Sensors* 23, 12 (June 2023), 5444. <https://doi.org/10.3390/s23125444>



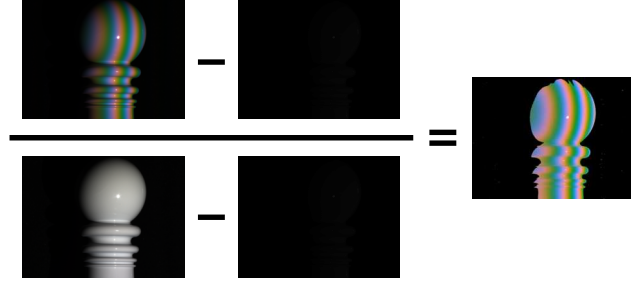


Fig. 15. Normalized colors (right) are obtained by dividing each pixel of an image acquired while projecting a pattern (top left) with the corresponding pixel of an image obtained by projecting white (bottom left), subtracting an image obtained while projecting black (middle) from both. The subtraction of ambient image becomes negligible in a darkroom, which we exploit in our high-speed setting to increase frame rate.

## A SUPPLEMENTAL MATERIAL

The sections below form the Supplement text to the paper "LookUp3D: Data-Driven 3D Scanning." We cover the effects of ambient light (Section B), the effects of over a large volume of reconstruction (Section C), additional details on residual analysis and thresholding (Section D), additional details on pattern design and evaluation (Section E), and the film development for patterns (Section F). We also explore more implementation details of *LookUp3D*, including denoising of the lookup tables (LUT), the use of a coarse-to-fine (C2F) approach in reconstruction to effectively add neighboring information, temporal consistency (TC) in dynamic scenes to restrict the depth range, and the exciting avenue of using a multilayer perceptron to learn the depth-to-color mapping (Section G). We also display a few additional dynamic reconstructions in Section H.

## B EFFECTS OF AMBIENT LIGHTING

Normalization is an important step in image processing to remove varying albedo, texture, and other nuisances in dealing with capturing objects. We use three different captures in *LookUp3D* to construct the normalized pixel intensity of a capture: a pattern illuminated  $I_P$ , white-flash illuminated  $I_W$ , and ambient non-illuminated  $I_B$ . The procedure, then, is element-wise described by the equation

$$I = \frac{I_P - I_B}{I_W - I_B} \quad (2)$$

and displayed in Fig. 15 with the example of a pawn.

In the high-speed setting, however, reducing the number of captures is important for achieving high frame rates, avoid motion blur, and accurately capturing the motion or deformation of objects.

In the main dynamic results presented, we forgo the ambient subtraction, which enables frame rates that are 1.5 higher in capturing dynamic scenes. We chose to capture all our high-speed data with the room blinds down and with the ceiling lights off, to get the best quality data out of our method and prototype. Furthermore, making the analog high-speed projector significantly brighter (using 100 W LEDs) is not only reducing the image noise at shorter exposures but also contributes to a  $I, I_W \gg I_B$  relation that further reduced the impact of ambient light in a darkroom.

This does bring in a limitation: the calibration and the data capture should be taken with similar lighting conditions, otherwise the quality of reconstruction degrades. For instance, if the calibration is done with the ceiling lights turned off, the reconstruction captures should not be done with ceiling lights on to avoid adding confusion into the depth decoding step.

While these main dynamic results were taken in darkroom for simplicity and reproducibility, we confirm that *LookUp3D* can obtain dynamic results with ambient light: we capture and subtract an ambient image at each timestep. This comes at the cost of a decreased frame rate, for instance from 450fps to 300fps. We highlight in this section two scenes: a hand wiggling its finger and a ball bouncing. Both scenes had the LUT calibrated in a darkroom, but the scenes were captured with the ceiling lights on.

Similar to the experiment in Section 4.2, we drop a ball on top of a ream of paper and let it bounce. For each reconstructed frame, we fit a sphere to the point cloud – this returns both the radius of the sphere and the residual (summed Euclidean distance of all points from perfect sphere). We use both values to calculate a weighted mean (using the inverse of the residual as the weight) of 31.20mm for the ball radius, whereas the real life measurement with a caliper is 31.00mm.

We additionally measured the maximum flux (in a darkroom) of our analog projector with a light meter: placed 0.5m away, it measured 1,510 Lux; at 1.0m, 650 Lux; and at 1.5m, 210 Lux. We then switched the ceiling lights on (700 Lux) and attempted to calibrate and reconstruct dynamic scenes: at 1.5 meter away, our projected pattern is no longer discernible in the scene when the ceiling lights are on.

## C EFFECTS OF DEFOCUSING

With *LookUp3D*, we store the normalized RGB values directly into a lookup table; therefore, any effect due to defocusing or vignetting from the projector gets baked into the LUT. This makes the approach particularly appealing since we skip modeling any complex optical behavior, including projector defocusing.

For this experiment, we use our dynamic set-up and calibrate a LUT with 1000 intervals, each with a linear stage stride of 0.5 mm – this calibration still takes under 10 minutes and requires no user interaction. We plot, on the left-hand side of Fig. 17, a ray that covers around 70 centimeters of depth, raging from 0.8 meters to 1.5 meters of distance from the camera sensor.

For the test scene, we grab the Bunny and move it through this large volume of reconstruction. The reconstructed point clouds in Fig. 17 show that we can still reconstruct the bunny as it sweeps the volume, with more noise in the first frame due to a decreased signal-to-noise-ratio as the bunny is around 1.25 m away from the camera-projector pair. The distance of the bunny between the first and last displayed frames is approximately 30 centimeters.

## D RESIDUALS ANALYSIS

The residual, computed as  $r_i = \|C_i(d_i^*) - I_i\|_2$  indicates consistency with the calibration data and serves as a proxy for reconstruction



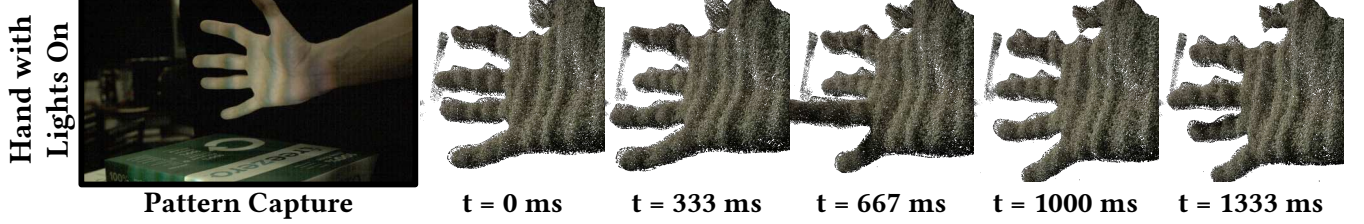


Fig. 16. **LookUp3D Reconstruction with Ceiling Lights On.** We reconstruct, at 300 fps and 1 ms exposure, a person’s hand wiggling their fingers. To handle ambient light, we now take three images per reconstructed frame: one with LED pattern on, one with white LED on, and one with both LEDs off.

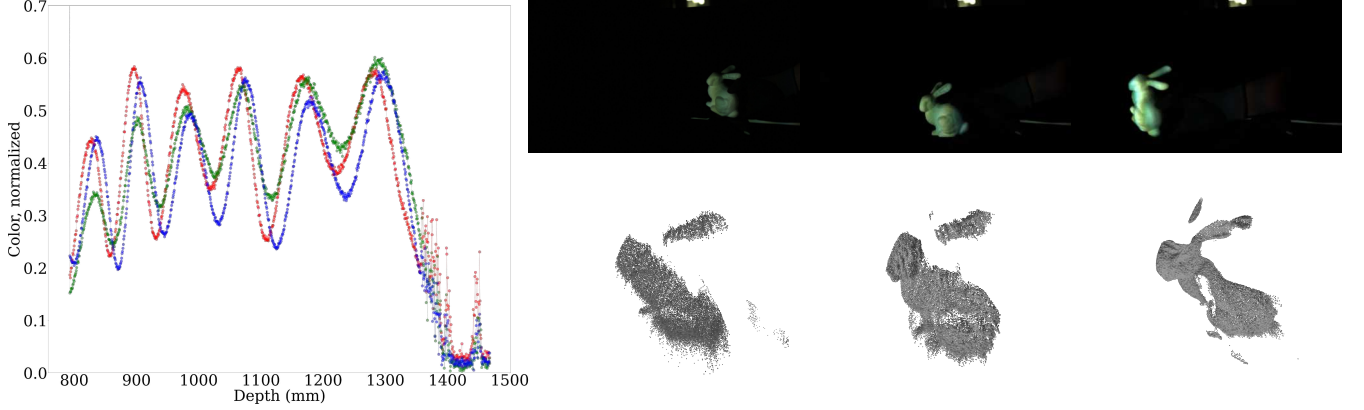


Fig. 17. **Effect of Defocus as Bunny Moves Through Large Volume.** We present a larger depth range with *LookUp3D* to understand the impact of defocusing. On the left, we plot what a calibrated ray from the LUT looks like, ranging from 0.8m to 1.5m of depth – any vignetting and defocusing effect from the projector gets baked into the LUT. On the right, we display three frames of bunny while the projector shines the *Spiral* pattern. Although we recorded this scene at 200 fps, we show three frames, each separated by 2.00 seconds, to highlight the bunny’s traveled distance. Immediately below each image, we render the reconstructed point cloud. The bunny moved, in this time span, approximately 30 centimeters forward and *LookUp3D* is still able to reconstruct the bunny’s geometry and position accurately.

confidence. It is an inherent byproduct of our lookup-based reconstruction process. No additional models or heuristics are required to assess confidence, making the system naturally self-explanatory.

To further understand and validate this signal, we investigate the relationship between the residual and observed brightness of each pixel (represented as the l2 norm of its RGB vector before normalization). As shown in Fig. 18, we observe a clear U-shaped trend: residuals are low for moderate brightness levels and rise at both extremes, which aligns with what we observe in the failure cases – dark pixels (e.g., in shadows) suffer from low signal-to-noise ratio, while overexposed regions (e.g., specularities, inter-reflections), saturate and flatten RGB variation, weakening discriminability. This observation highlights that residuals reflect physical image formation properties, and that our system performs reliably with moderate brightness.

To evaluate the effectiveness of using the residual to identify incorrectly reconstructed points (i.e., residual-based filtering), we conduct experiment on the same static scene (the pawn) captured with the spiral pattern. We test multiple threshold levels and report quantitative metrics against the ground truth CAD model (Table 3). While the main paper uses a coarse threshold of 0.1, progressively stricter thresholds yield sparser but more accurate reconstructions.

This observation enables a trade-off: downstream applications requiring precise surface geometry (e.g., measurement or physical property inference) may benefit from aggressively filtered point clouds, at the cost of completeness. To address this, we present in the future work preliminary results on residual-based refinement, a promising direction for further exploration.

Table 3. Quantitative evaluation of residual-based filtering. Tighter thresholds lead to consistent drops in both median absolute error (MedAE) and root mean squared error (RMSE). MedAE remains more stable, as it is less sensitive to outliers. Notably, filtering at a threshold of 0.02 yields an RMSE of 1.53 mm compared to 6.39 mm without filtering, by removing 23% of points, primarily noisy outliers.

Residual Threshold	# Points	MedAE (mm)	RMSE (mm)
No Filter	809,281	1.29	6.39
0.10	806,152	1.28	6.18
0.05	775,710	1.26	4.62
0.02	621,130	1.26	1.53
0.01	429,923	1.25	1.52

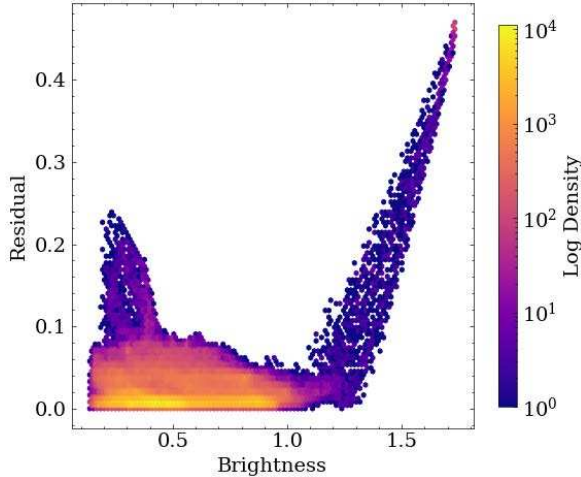


Fig. 18. Relationship between residual and observed brightness. This hexbin plot visualizes per-pixel density over residuals (y-axis) and pre-normalization brightness (x-axis). The residual exhibits a clear U-shaped trend: for brightness below 0.5, residuals increase moderately; above 1.0, residuals rise sharply in an almost linear fashion; and in between, residuals remain consistently low (under 0.1). Most points are concentrated in this stable region, suggesting that the majority of the scene is reconstructed with high confidence.

## E PATTERN EVALUATION

The mechanism of *LookUp3D* relies on establishing per-pixel mapping from observed color to depth. Ideally, this mapping is unique with each color corresponding to a unique depth. The extent to which this is true in practice plays a critical role in reconstruction accuracy. If two or more depth values have similar colors stored in the LUT during calibration, then, during reconstruction, a simple lookup search will often yield incorrect depths. In Fig. 20, we plot what the confusion matrix looks like for a *Spiral* pattern – the confusion is computed as the Euclidean distance between a vertical stripe and all other stripes and ranges from 0 (most dissimilar) to 1 (most similar). The perfect, although impossible, pattern would achieve a confusion matrix with 1s in the diagonal and zero elsewhere.

We evaluated reconstruction performance across a wide range of pattern designs in the static setting. This included established structured light patterns, such as Gray code and space-filling curve Hilbert patterns, as well as patterns encoded in 3, 6, and 9 color channels. For our set-up, we used a  $1920 \times 1080$  resolution projector; therefore, our patterns were designed as 1920 vertical stripes, each with a unique color sampled from smooth functions (top of Fig. 19).

The results, shown in Table 4, indicate that increasing the number of channels generally improves reconstruction accuracy. We also observed that different patterns tend to favor different object geometries, reflecting trade-offs between pattern smoothness, spatial coverage, and robustness to ambiguity. You can find what the Bunny looks like in Figs. 3, 13, 17, 28; the Irregular Structure in Figs. 11, 4, and 25; the Dodo in Fig. 25; and the House in Fig. 5.

We additionally visually display the results with the same patterns from Table 4 in Fig. 19 with the Pawn. We also see visually that

increasing the number of channels generally improves reconstruction accuracy. We would also like to highlight some of the artifacts of reconstruction: for instance, the wiggles in 6-channel *Hilbert* and the aliasing in 3-channel *Spiral*. These artifacts are due to strong confusion in *LookUp3D* and mostly arise from sensor noise during both calibration and scanning – if the pattern is already designed with similar color values, then even small amounts of sensor noise can cause different depth values to end up with extremely similar colors stored into the LUT. With 11 channels (and no denoising of the LUT), the confusion of *LookUp3D* decreases and the artifacts disappear in the pawn (right-most reconstruction of Fig. 19).

Given that our method is designed to generalize on both patterns and object geometries, and that the dynamic setting requires a single RGB pattern (i.e., 3 channels), we choose the spiral pattern for high-speed reconstruction. While Hilbert patterns yielded the lowest errors in static scans, they degrade sharply when blurred, which is an issue in high-speed capture with shallow depth of field and film-based projection. Spiral patterns, by contrast, maintain non-ambiguity more reliably under blur and warping (see Fig. 20), making them a better fit for dynamic scenes.

Future work may explore tailored pattern design and optimization for specific hardware and scene conditions, though our current results already demonstrate that simple, unoptimized patterns can yield accurate reconstructions across a range of setups.

## F FILM DEVELOPMENT FOR PATTERNS

Our analog projector can flicker high-power LEDs at extremely high rates. In order to get a three-channel pattern projected onto the scene, we need to place a transmissive sheet in front of the projector LED. For the prototype presented in this paper, we opted for a 35mm film.

We use a Canon EOS-1N analog camera with Tamron 90mm f/2.8 lens and use both FUJIFILM Reverse Film Fujichrome Velvia 50 135 Film and Kodak Ektar 100 Professional ISO 100. We design patterns as 1920 vertical stripes and use the same DLP projector and shine these various patterns onto a flat white screen. We place the camera atop the projector and expose each pattern on film at 1/6, 1/10, 1/12, 1/15, 1/20, 1/30 second. We use a commercial third-party to develop the film.

The patterns are designed such that every vertical stripe is its unique color and with enough color distinction between different depths to avoid noise. Developing the film introduces color mixing that is hard to model and provokes confusion in the lookup table (see Fig. 21 for the original pattern design on the left and the developed film on the right). The color mixing is also heavily dependent on the batch of film roll and the Bayer filter employed by the high-speed camera, making it even harder to predict beforehand any artifact that comes from exposing and developing the film.

In addition to color mixing, the film, after developing, obtains a grainy texture, which further causes issues in the lookup table. Most of the patterns we use are designed as smooth functions along the RGB cube, so the granularity introduced by the film goes against the smoothness. To overcome that, we purposefully keep the analog projector out of focus for the whole volume of reconstruction; by

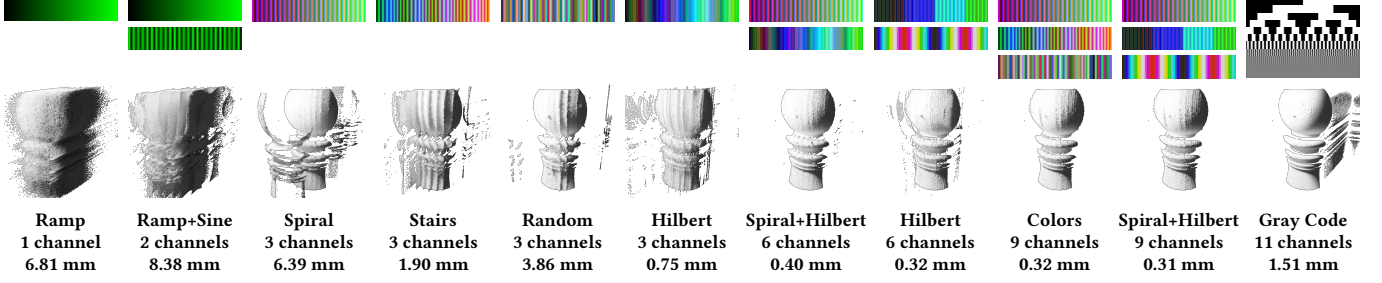


Fig. 19. **Reconstructions with Different Patterns.** We calibrate 11 different LUTs and reconstruct the same scene: a static pawn. In the top row, we display the patterns used for calibration and reconstruction. In the middle, we display the reconstructed point cloud – we would like to highlight that the quality of reconstruction with *LookUp3D* improves as more channels are included in the LUT. This is because the confusion of a naive lookup decreases as more channels are introduced. In the bottom line, we also report the root mean squared error (RMSE) in millimeters of the reconstructed cloud and the ground truth geometry of the pawn, further validating that, as more channels are introduced, the more precise the reconstruction becomes.

Table 4. Comparison of performance of our proposed *LookUp3D* method with several different patterns in static setup. We report the mean absolute error (MAE), median absolute error (MedAE), and root mean squared error (RMSE), all in millimeters, for each reconstructed point cloud with respect to its ground-truth geometry. For a visualization of what the patterns look like, refer to Fig. 19. For a visualization of the objects, check Fig. 3 for the Bunny, Fig. 25 for the 3D printed Dodo, Fig. 4, and Fig. 5 for the House. Hilbert and Spiral patterns stand out as better when constrained to 3 channels.

Pattern	# Channels	Bunny			Irregular Structure			Dodo			House		
		MAE	MedAE	RMSE	MAE	MedAE	RMSE	MAE	MedAE	RMSE	MAE	MedAE	RMSE
Ramp	1	4.54	3.62	5.83	4.03	3.12	5.51	7.99	6.11	12.10	5.48	4.09	7.96
Ramp+Sine	2	4.22	3.23	5.66	1.39	1.66	5.05	10.76	5.16	16.61	4.82	3.04	7.75
Spiral	3	5.21	0.46	10.42	2.96	1.26	6.61	17.40	0.95	31.07	41.73	4.48	70.4
Stairs	3	1.05	0.54	1.89	1.53	1.26	1.96	5.61	0.81	13.36	2.81	1.78	6.47
Random	3	3.90	0.49	11.82	1.97	1.17	5.87	3.27	0.65	12.16	8.72	1.82	25.24
Hilbert	3	0.86	0.49	1.46	1.39	1.21	13.50	2.63	0.80	7.82	2.21	1.71	2.77
Spiral+Hilbert	6	0.69	0.39	1.12	1.25	1.20	1.39	1.06	0.58	4.75	2.28	1.61	4.65
Hilbert	6	0.92	0.46	1.89	1.27	1.22	1.36	0.82	0.55	1.84	2.08	1.68	2.77
Colors	9	0.68	0.39	1.92	1.24	1.20	1.35	0.57	0.54	0.89	1.87	1.58	2.45
Spiral+Hilbert	9	0.60	0.40	1.03	1.25	1.20	1.34	0.55	0.52	0.66	1.86	1.57	2.57
Gray Code	11	0.50	0.32	1.26	1.28	1.24	1.36	0.50	0.48	0.74	1.86	1.57	2.46

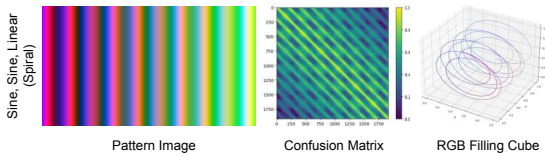


Fig. 20. **Spiral Pattern.** Representative *Spiral* pattern (left) with low probability of confusions between different image column pairs (middle). The representation of the pattern as a path in an RGB color cube is shown on the right. The linear component of the pattern in the green channel is particularly useful for identifying when a pixel is out of bounds of the reconstructible volume.

projecting a defocused pattern, we can achieve a smoother ray for the lookup table and avoid the granularity of the film.

Due to the difficulties of modeling color mixing and other random effects of developing film, our data capture currently relies on exposing several patterns on film and pick the best one based on

our experiments. There are likely better ways to produce the film and alternatives to film itself (such as stained glass), which we leave as future work.

## G LOOKUP3D IMPLEMENTATION DETAILS

Reconstruction with *LookUp3D*, in its unoptimized form, is a single lookup on each observed pixel  $I_i$  against a depth-to-color mapping function  $C_i(d)$ :

$$d_i^* = \arg \min_{d \in \mathcal{D}} \|C_i(d) - I_i\|_2, \quad (3)$$

where the search is done on a discrete set of points  $\mathcal{D}$ .

This naive implementation works well in our static setting, where noise is minimal since we use a high-end monochromatic camera and a high-end projector. However, bringing the naive implementation of the pipeline to our dynamic setting is quite challenging. In this section, we describe the steps we found necessary to achieve high-quality reconstructions in adversarial conditions.

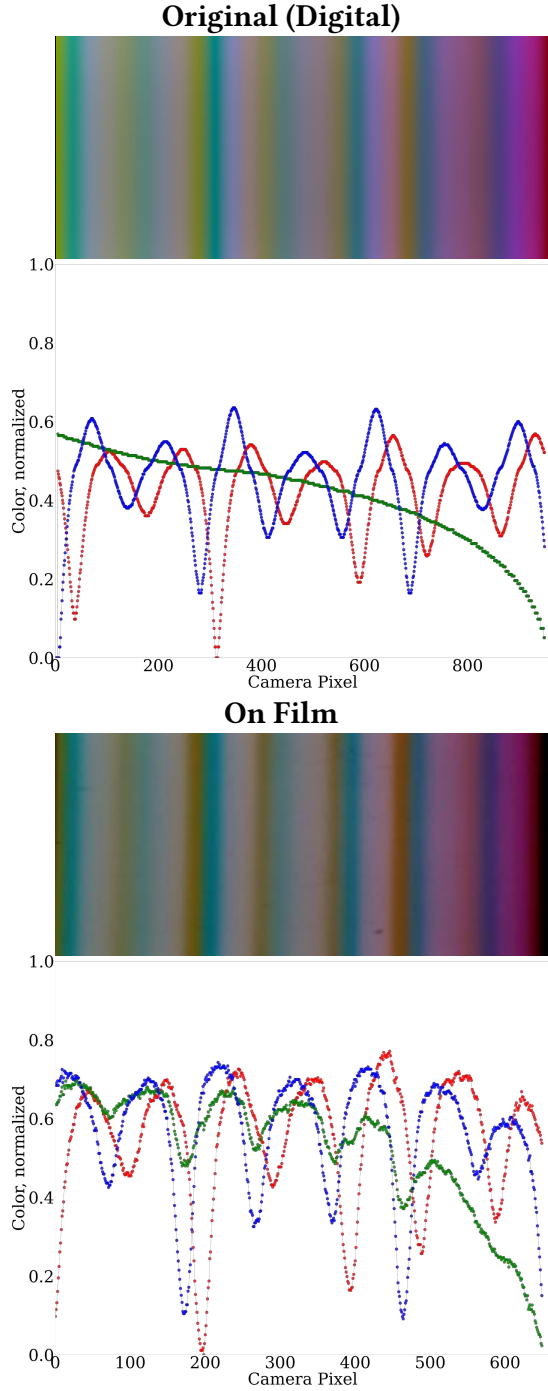


Fig. 21. **Film Development.** The *Spiral* pattern is designed as a monotonic slope in green and two shifted sinusoidal functions with parametrized amplitudes in the red and blue channels. When the film is developed with the *Spiral* pattern on it, the film granularity and the color mixing impact the quality of the pattern. This introduces confusion to the lookup tables in our dynamic setting, which we try to overcome with strategies such as coarse-to-fine reconstructions and enforcing temporal consistency.

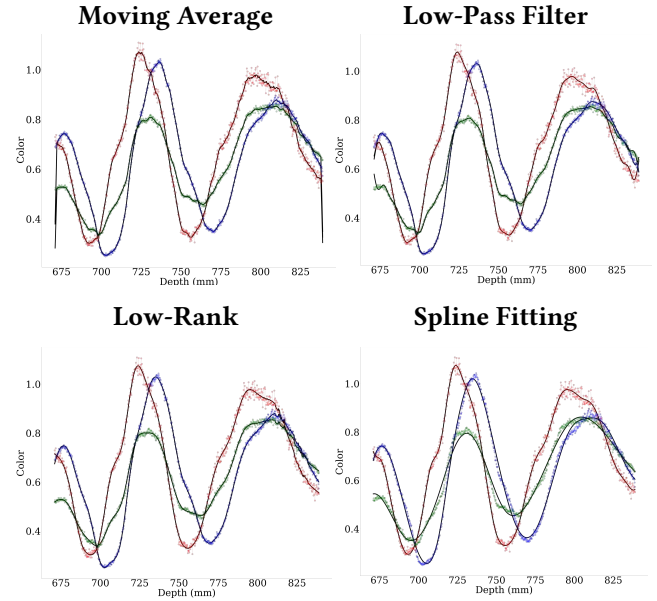


Fig. 22. **Example of Denoised Ray.** We illustrate, with color, what the red, green, and blue channels of a single ray looks like. We overlay in black the denoised rays from: a moving average of size 11 (top left), a low-pass filter with a cutoff at frequency 25 (top right), a low-rank approximation using the top 25 singular values (bottom left), a cubic B-Spline with smoothing parameter of 0.1 (bottom right). The black curves follow the trend of the red, green, and blue curves well, while neglecting some noisy, high-frequency components.

### G.1 Denoising.

Our lookup tables are a collection of densely sampled color values that are a  $H \times W \times n \times 3$  tensor, where  $H$  and  $W$  are the height and width of the image,  $n$  is the number of depth steps. The patterns we use are often smooth in comparison to binary patterns, i.e. they do not change rapidly. This is particularly important in compressing and denoising our data. There is a plethora of strategies to denoise data; here, we explore a few of such strategies and discuss their respective advantages and disadvantages.

*Moving Average.* The moving average is the simplest (and fastest) form of reducing variance in a function. In a single ray, we use a fixed window size to average the color intensity at a certain depth point with its neighboring depths (top left of Fig. 22). This does nothing to reduce storage and it maintains the original depth discretization from calibration.

*Low-Pass Filter.* We denoise each channel of the LUT independently. We apply a Fourier Transform to each ray of every channel, bringing them into the frequency domain. We then apply a low-pass filter and perform the inverse Fourier Transform to bring the ray back into the spatial domain, but now with the high-frequency details removed. With Fast Fourier Transform algorithms implemented in *numpy* and *scipy*, this is a fast way to get rid of noise in the data,



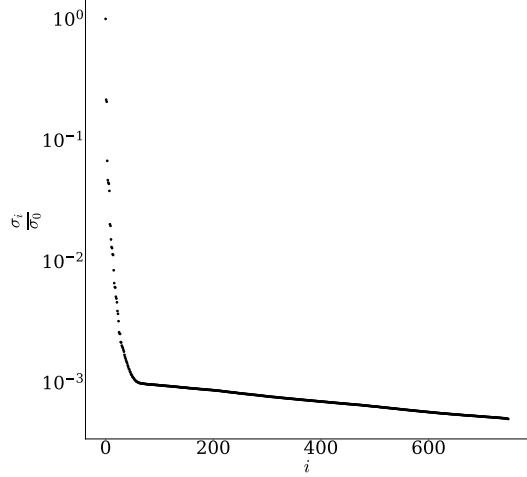


Fig. 23. **Singular Values of Red Channel of lookup table.** We plot all 750 singular values normalized by the top singular value  $\sigma_0$ . The mass of the red channel is concentrated in the top 50 singular values, meaning we can safely truncate the vast majority of bottom singular values while maintaining the overall structure of the matrix.

although it maintains the depth discretization from calibration and does nothing to reduce storage requirements.

**Low-Rank Approximation.** We perform the low-rank approximation independently on each channel of the LUT, e.g., the  $H \times W \times n$  tensor representing the red channel. We gather all pixels’ values into a matrix  $A \in \mathbb{R}^{k \times n}$  where  $k = HW$  gathers all the pixels. We then construct a low-rank approximation by taking the SVD  $A = U\Sigma V^T$  where  $U, V$  are orthonormal matrices and  $\Sigma$  has the singular values  $\sigma_0, \dots, \sigma_{n-1}$  along its diagonal.

By truncating the singular values to the top  $r$  and reconstructing the matrix  $A_r = U_r \Sigma_r V_r^T$ , one obtains the closest (in the Frobenius norm sense) rank- $r$  matrix [Eckart and Young 1936]. The accrued error is given by the Frobenius norm of the difference

$$\|A - A_r\|_F^2 = \sum_{i=r}^{n-1} \sigma_i^2,$$

and so by choosing a value of  $r$  for which most of the discarded singular values are much smaller than the top  $r$  singular values is desirable to maintain accuracy while denoising the data and compressing the matrix for storage.

In Fig. 23, we plot all 750 singular values of the red channel of the lookup table we use for our dynamic scenes. The plot shows that the mass of the matrix is concentrated in the top 50 singular values, which means that we can safely truncate the vast majority of these singular values while maintaining accuracy. In the bottom left corner of Fig. 22, we plot the example of a ray from the same lookup table, after keeping only the top 25 singular values for all color channels – the low-rank approximation accurately captures the trend in the color intensity through depth, while discarding most of the high frequency (and thus likely noise) components of the ray.

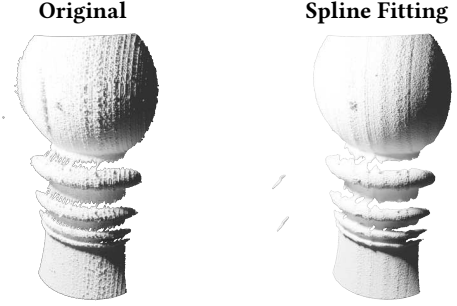


Fig. 24. **Spline Fitting with 9-channel pattern.** We denoise the LUT by fitting cubic splines per channel per ray, setting a smoothing parameter of 0.1. The splines, in addition to denoising the data, allow for a smooth depth-to-color mapping, further reducing the error in reconstruction.

Low-rank approximation is a powerful and fast tool to reduce storage requirements of the lookup tables: in the static set-up, the 3-channel *Spiral* pattern with 700 depth steps and 3.7 Megapixel resolution takes 38.8 GB to store; after low-rank approximation with the top 35 singular values for all channels, that storage is reduced to 1.8 GB, 21 times smaller than the original one. The depth discretization, however, remains the same.

**Spline Fitting.** We use, per-ray, cubic B-splines, creating a smooth depth-to-color mapping; moving average, low-pass filter, and low-rank approximation all maintain the original discretization from *LookUp3D* calibration. With splines, instead, we have a direct way to interpolated data and further reduce error of reconstructed clouds. In our static setting with the 9-channel *Spiral+Random+Stairs* pattern, RMSE of the pawn decreases from 1.44mm to 0.70mm (Fig. 24).

Spline fitting also reduces the storage requirements for the lookup tables, but not as drastically as the low-rank approximation. In the static set-up, the 3-channel *Spiral* pattern with 700 depth steps and 3.7 Megapixel resolution takes 38.8 GB to store; with cubic B-spline fitting that storage is reduced to 5.4 GB, 7 times smaller than the original one but 3 times larger than low-rank.

## G.2 Coarse to Fine (C2F).

We first downsample our captured image to very few pixels. We search, for this subsample of pixels, the whole range of depth values and pick the minimum. Then, when double the resolution and choose limit the subsample to be half of the whole depth range. However, we still ensure that the residual in this shortened range is below a threshold  $\epsilon$  – otherwise, we double the range back up until we search all depth points in the LUT or we find a depth position below threshold  $\epsilon$ .

C2F effectively reconstructs, first, a coarse geometry of the object and uses it as a prior for higher-resolution reconstructions. C2F forces the reconstructed points to lie on the main structure of the object. In Fig. 25, we reconstruct the pawn, a 3D printed dodo, and the precisely-machined irregular structured with the 3-channel *Spiral* pattern. On the left, the two objects are reconstructed with the naive method; on the right, we use C2F in the reconstruction pipeline.



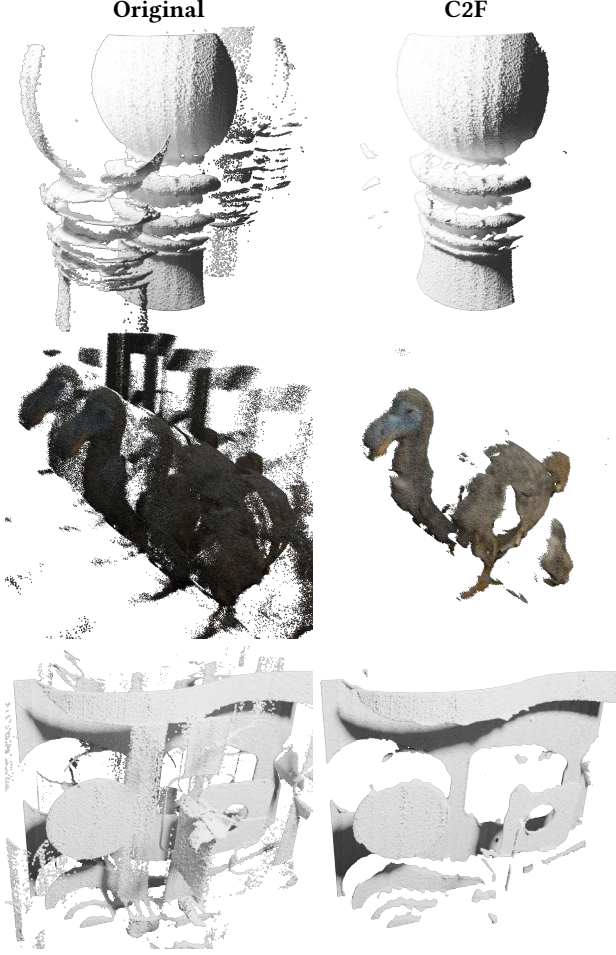


Fig. 25. **Coarse-to-Fine (C2F) with 3-channel pattern.** We first use a lower-resolution normalized image to reconstruct a coarse geometry of the object. Then, we increase the resolution of the image but limit the range of depths to search in the lookup table, effectively using the coarser geometry as a prior for the full-resolution reconstruction. In the original reconstruction, the pawn has many points wrongly reconstructed; with C2F, several of these points get pushed back into the correct main structure.

For the pawn, the RMSE drops from 6.66 mm (993,296 points) to 0.76 mm (947,678 points); for the dodo, the median absolute error was 0.95 mm and the RMSE was 31.07 mm (1,169,540 points), but with C2F they drop to 0.61 mm and 6.26 mm (869,780 points) respectively; for the irregular structure, the RMSE drops from 6.61 mm (2,451,315 points) to 2.36 mm (2,153,456 points).

### G.3 Temporal Consistency.

Given the reconstruction of the previous frame, we avoid searching through the whole depth range again and instead search around a small parameter  $\delta$  millimeters around the previous depth and find the minimum. If the residual in this shortened range is above a threshold  $\epsilon$ , we double the search to  $2\delta$ . We repeat until we search all depth points in the LUT or we find a depth position below threshold

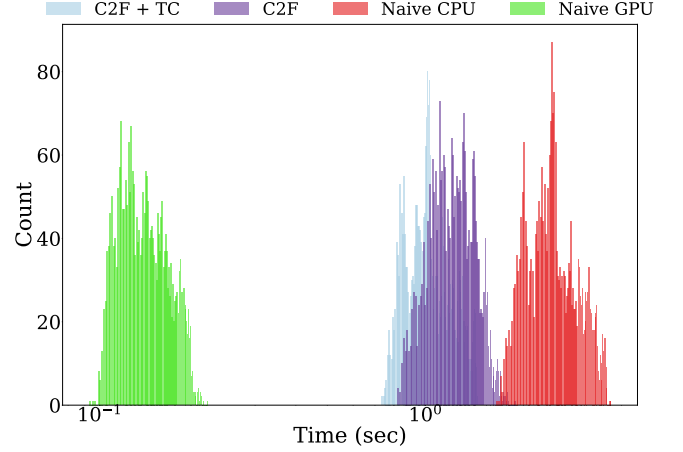


Fig. 26. **Times for Depth Decoding using Different Strategies.** We collect the per-frame time to decode depth for ten different dynamic scenes. We use the naive (with low-rank approximation) approach that is implemented for the static reconstructions; we also use the coarse-to-fine (C2F) approach, and, ultimately, add the temporal consistency (TC), which gives the shortest median time for depth reconstruction. Naive CPU, C2F, and TC reconstructions are done using one core of an AMD EPYC 9654P processor clocked at 3.7 GHz. The naive GPU reconstruction is done in an NVIDIA RTX A6000 GPU.

$\epsilon$ . The parameters  $\delta$  and  $\epsilon$  can be modified; a smaller  $\delta$  can lead to artifacts, where; a smaller  $\epsilon$  can lead to longer reconstructions as most pixels will end up searching the whole ray to find the minimum. We set  $\delta = 5$  mm in all our reconstructions and  $\epsilon$  to 0.1 or 0.2, depending on how noisy the results were.

Finally, we show that both C2F and TC strategies, in addition to improving the reconstruction quality in the dynamic setting, also reduce (by at least half) the average time of decoding depth for a full frame (Fig. 26). All reconstructions are done using 1 core of an AMD EPYC 9654P processor clocked at 3.7 GHz. We also implement, with PyTorch, an argmin search in the GPU, using an NVIDIA RTX A6000 GPU.

### G.4 Multilayer Perceptron

All the results presented in this paper are reconstructed by a per-pixel lookup against a depth-to-color mapping  $C_i(d)$ :

$$d_i^* = \arg \min_{d \in \mathcal{D}} \|C_i(d) - I_i\|_2, \quad (4)$$

where the search is performed over a discrete set  $\mathcal{D}$ .

As an unoptimized proof of concept prototype, we also learn the lookup search with a multilayer perceptron (MLP) for depth decoding. We use five input features per ray ( $i/H$ ,  $j/W$ ,  $r$ ,  $g$ ,  $b$ ), where  $H$ ,  $W$  are the height and width of the camera resolution,  $i$ ,  $j$  are the row and column of a certain pixel, and  $r$ ,  $g$ ,  $b$  are the normalized color values associated with the pixel. We apply a positional encoding with  $L=10$  frequency levels, yielding a 105-dimensional input. The network is a compact MLP with two hidden layers of widths (256, 128) and a final 1-unit output; the activation is a *Swish* function (where  $\text{Swish}(x) = \frac{x}{1+e^{-x}}$ ). We train with AdamW

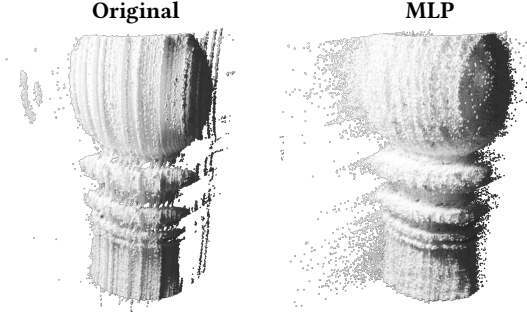


Fig. 27. **MLP depth prediction with a 3-channel pattern.** We train an MLP on the 3-channel *Hilbert* pattern and predict the pawn geometry.

(learning rate  $10^{-3}$ ) for up to 100 epochs (validation RMSE typically plateaus by around 20 epochs; we report the best epoch) using mean squared error between predicted and stored depth. Depth is normalized to  $[0, 1]$  during training and mapped back to mm for reporting.

Training data are built from a  $300 \times 300$  crop of the LUT (masking depths in  $[500, 2000]$  mm), giving a total of 67,409,986 rays. Each epoch processes  $\approx 4,115$  mini-batches of up to 16,384 rays (shuffle; last batch is smaller). On a single NVIDIA H100 (80 GB HBM3), each epoch takes  $\sim 2$ – $3$  minutes; over 100 epochs this corresponds to roughly 3–5 hours wall-clock in our environment. The model has 60,161 parameters ( $\approx 0.23$  MB in single floating-point precision). Inference-time reconstruction runs in about 1–2 seconds per frame on the same GPU.

Fig. 27 shows a side-by-side comparison using the 3-channel *Hilbert* pattern: direct lookup (994,319 points, RMSE 1.86 mm) vs. the trained MLP (991,706 points, RMSE 1.03 mm).

## H ADDITIONAL DYNAMIC RECONSTRUCTIONS

In this section, we show three dynamic sequences reconstructed with our high-speed prototype and *LookUp3D*.

We placed the Stanford bunny on a platform and attached it to our linear stage. We begin capturing and set the linear stage to move 31.25 millimeters at 12.50 mm/s. We reconstruct its linear motion at 450 fps (Fig. 28). We use these reconstructions to validate both the accuracy of the reconstructed point cloud, achieving a root mean square error of 3.27 mm. This result is mentioned in our quantitative analysis of the high-speed scanning regime (Section 4.2).

We grab a small, white, flat plane (approx  $20\text{cm} \times 15\text{cm}$ ) and move (rotate and translate) it in the volume of reconstruction. We reconstruct around six seconds of motion at 450 fps, resulting in over 3,000 frames and more than one billion reconstructed points (Fig. 29). We assume that the plane is and remains perfectly flat throughout the motion; this is an additional way to quantitatively evaluate the performance of *LookUp3D* and our prototype at high frame rates for each frame. We fit a perfect plane to our reconstruct point clouds and measure the total deviation from this assumption: we obtain a root mean square error to be 3.83 mm, validating precision of our system at high frame rates.

We grab a sheet of paper and tear it in half. We reconstruct around three seconds of it at 100 fps (Fig. 30).

We also attempted to use our high-speed prototype with *LookUp3D* to reconstruct a balloon popping (Fig. 31). Unfortunately, even at 1450 fps, the motion of the balloon as it explodes is too fast for our scanner. Since we take two images (a pattern projected and a white flash), if these two images do not overlap much, *LookUp3D* will most likely fail to recover any relevant depth.

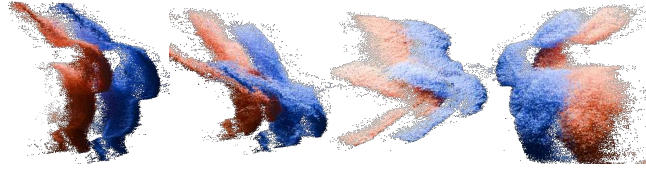


Fig. 28. **Bunny in Linear Motion.** We reconstructed, at 450 fps, the bunny being pushed by a linear stage. This image displays two point clouds together in the same coordinate system – the one to the left (orange) is the bunny at the beginning of the motion, before the linear stage had started moving; the one to the right (blue) is the bunny at the end of the motion, 1249 frames later, just after the linear stage had stopped moving. We set the speed to 12.50 mm/s and the distance to 31.25 mm – we recovered a speed of 12.56 mm/s and a distance of 31.52 mm with our reconstructions.

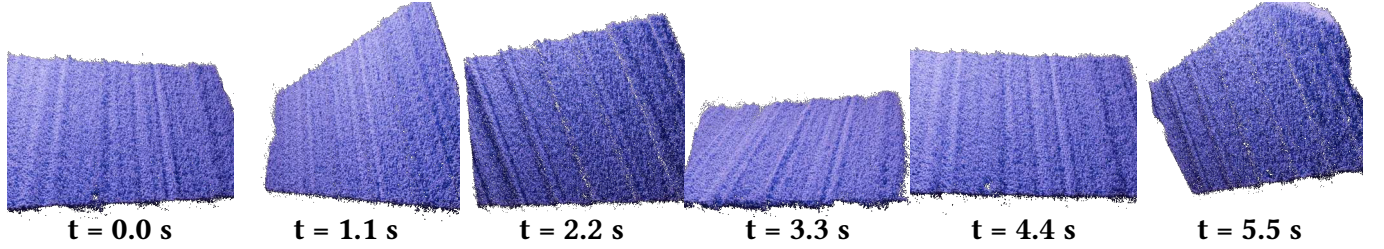


Fig. 29. **Flat Plane in Motion.** We reconstructed, at 450 fps, a flat plane moving around the reconstruction volume. The average root mean square error was less than 4 mm, validating precision of *LookUp3D* and our prototype at high frame rates.

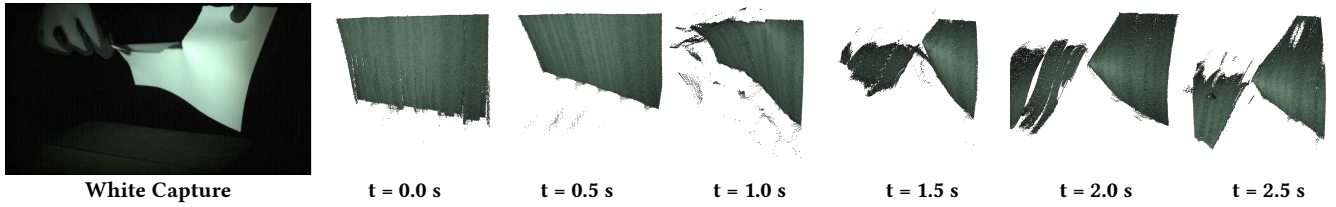


Fig. 30. **Ripping Paper.** We reconstructed, at 100 fps, a sheet of paper being split in half.

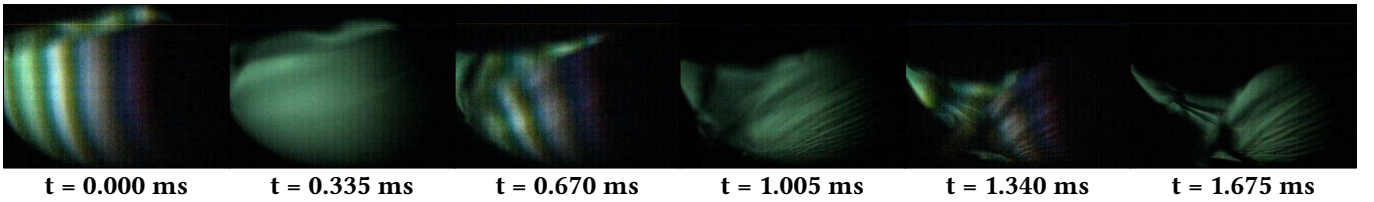


Fig. 31. **Failure of Balloon Popping.** We set our camera exposure to 0.3 millisecond and recording frame rate at 2,900 fps. The motion of the balloon popping, however, is too fast – consecutive frames do not overlap much. Since *LookUp3D* normalizes a pattern image against a white flash, the normalized image cannot be computed with very different pattern and white flash.