

Dynascape: Immersive Authoring of Real-World Dynamic Scenes with Spatially Tracked RGB-D Videos

Zhongyuan Yu
zhongyuan.yu@tu-dresden.de
Immersive Experience Lab
Technische Universität Dresden
Dresden, Germany

Victor Victor
victor.victor@mailbox.tu-dresden.de
Immersive Experience Lab
Technische Universität Dresden
Dresden, Germany

Daniel Zeidler
daniel.zeidler@mailbox.tu-dresden.de
Immersive Experience Lab
Technische Universität Dresden
Dresden, Germany

Matthew McGinity
matthew.mcginity@tu-dresden.de
Immersive Experience Lab
Technische Universität Dresden
Dresden, Germany

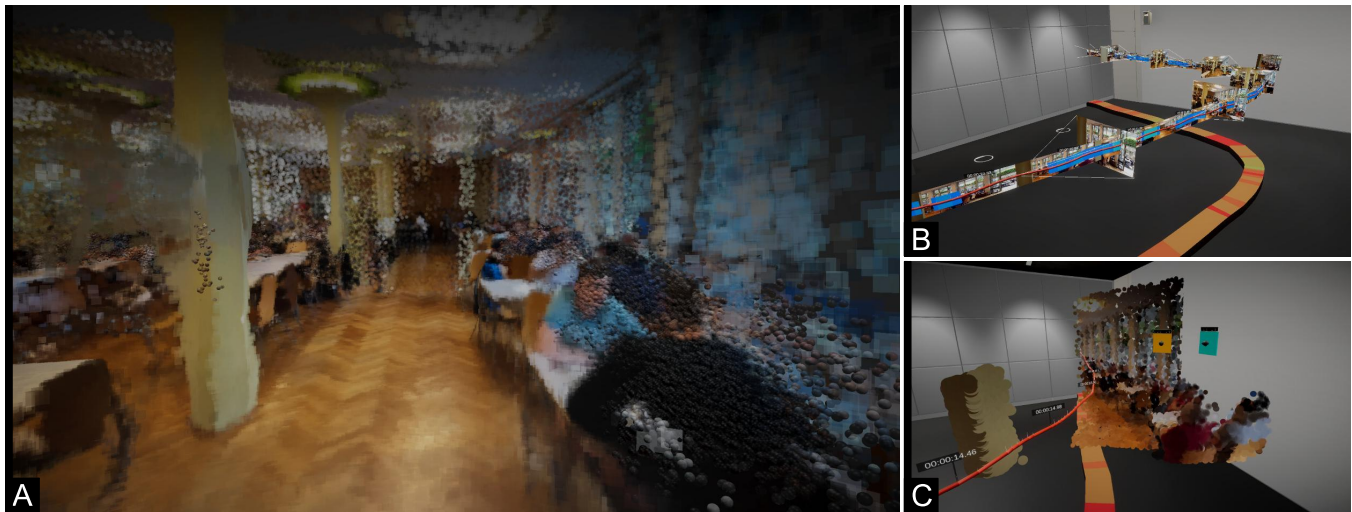


Figure 1: Dynascape. (A) Stylistic rendering with humans, foreground, and background rendered with different visual styles. (B) Previewing a spatially tracked RGB-D video with camera trajectory based visualizations. (C) Plane indicators help to align a video recording with the physical wall.

ABSTRACT

In this paper, we present Dynascape, an immersive approach to the composition and playback of dynamic real-world scenes in mixed and virtual reality. We use spatially tracked RGB-D cameras to capture point cloud representations of arbitrary dynamic real-world scenes. Dynascape provides a suite of tools for spatial and temporal editing and composition of such scenes, as well as fine control over

their visual appearance. We also explore strategies for spatiotemporal navigation and different tools for the in situ authoring and viewing of mixed and virtual reality scenes. Dynascape is intended as a research platform for exploring the creative potential of dynamic point clouds captured with mobile, tracked RGB-D cameras. We believe our work represents a first attempt to author and playback spatially tracked RGB-D video in an immersive environment, and opens up new possibilities for involving dynamic 3D scenes in virtual space.

CCS CONCEPTS

• **Human-centered computing** → **User interface toolkits; Visualization toolkits; Mixed / augmented reality.**

KEYWORDS

Immersive Authoring, Human Computer Interaction, Data Visualization

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

VRST 2023, October 09–11, 2023, Christchurch, New Zealand

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0328-7/23/10...\$15.00
<https://doi.org/10.1145/3611659.3615718>

ACM Reference Format:

Zhongyuan Yu, Daniel Zeidler, Victor Victor, and Matthew McGinity. 2023. Dynascope: Immersive Authoring of Real-World Dynamic Scenes with Spatially Tracked RGB-D Videos. In *29th ACM Symposium on Virtual Reality Software and Technology (VRST 2023), October 09–11, 2023, Christchurch, New Zealand*. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3611659.3615718>

1 INTRODUCTION

The capture of dynamic real-world scenes for display in VR, such as a busy street or crowded marketplace, remains an open challenge. While panoramic 360° cameras can capture arbitrarily dynamic scenes, the lack of 3D structure and singular viewpoint greatly restricts the viewers' freedom of motion during playback. Conversely, 360° LiDAR scanners and photogrammetry can capture 3D structures, thereby allowing for unrestricted 6 degrees of freedom of motion during viewing (6-DOF display), but are restricted to capturing static scenes only [10, 30]. Multi-camera video capture systems succeed in capturing both 3D structure and motion (e.g. [15]), but are currently only suitable for studio settings and cannot easily be used to capture arbitrary real-world scenarios. With the aforementioned techniques, the capture and immersive display of lively real-world scenarios remain elusive.

Recently a new class of mobile sensors has emerged that combines a color camera with a depth sensor and visual-inertial odometry, allowing the sensor to keep track of its position and orientation over time. We refer to such devices as tracked RGB-D cameras. Such functionality can be found in stereo cameras such as the ZED-X [28] or, even more conveniently, in mobile phones and tablets equipped with a LiDAR depth sensor, such as recent editions of the Apple iPhone or iPad [1]. The usefulness of these devices arises from combining the depth data with the camera pose information to recover the 3D world position of each RGB pixel. This permits tracked RGB-D videos to be easily displayed in virtual reality as animated 3D point clouds, with each video pixel displayed in its 3D location. As these 3D point clouds permit free 6-DOF viewing in virtual reality, they represent an interesting non-photorealistic approach to the capture and display of dynamic scenes. They are easy to use, highly mobile, affordable, inconspicuous, and can be used in almost any environment. Further, for certain applications, such as VR exposure therapy, immersive journalism, education, or art, non-photorealistic point cloud representations may be sufficient or even preferable for evoking a desired experience in the viewer [32, 34]. We are motivated by the belief that, in such use cases, animated point clouds derived from tracked RGB-D videos represent an accessible and expressive method for the capture and display of real-world scenes.

However, some challenges must be overcome if such an approach is to be effective, for tracked RGB-D recordings can seldom be used without some form of preparation or adjustment. With a singular point of view and limited field of view, a single RGB-D camera can only provide a partial view of a scene, and only by combining multiple recordings can a more comprehensive representation of a scene be attained. This necessitates tools for the spatial and temporal editing and composition of multiple RGB-D videos. It also creates a need for approaches to asset management and previewing.

Further, as the depth information is restricted in range and accuracy, methods for filtering noisy points are required. Finally, unlike video pixels, a 3D point must be given a specific visual size, shape, and appearance, and these visual styles greatly influence the virtual experience. Therefore tools for controlling the visual appearance of points are needed.

This work presents Dynascope, an approach to constructing VR scenes from tracked RGB-D videos in virtual reality. With tools for asset management, interactive previewing, temporal and spatial composition, alignment and layering, visual styling, and noise removal, the system is designed to explore the potential of tracked RGB-D videos for dynamic scene capture.

Dynascope supports authoring and viewing in both mixed reality and virtual reality. We use the term *in situ authoring* to denote the authoring of site-specific compositions in the precise location they are intended to be *viewed*. In this paper, we primarily focus on mixed reality *in situ* authoring, in which the author works in mixed reality to place recordings directly in the location they are intended to be viewed. However, we note that our tool can be used equally in VR and *ex situ*. The benefits of immersive *in situ* authoring are elaborated in the design analysis (Section 3) below.

Our main contributions are the following:

- (1) A mixed reality platform for composing scenes with animated point clouds captured with tracked RGB-D cameras.
- (2) Interactive previewing of tracked RGB-D videos at 1:1 and miniaturized scales, with tools and methods for analysis and simplification.
- (3) An immersive interface for *in situ* spatial and temporal navigation, authoring, and alignment.
- (4) An approach to tailoring visual appearance based on the concept of styles and class.
- (5) A prototype implementation that demonstrates the functionality and potential of immersive *in situ* authoring.

2 BACKGROUND AND RELATED WORKS

2.1 Dynamic 3D Scene Capturing and Rendering

Panoramic (360° or 180°) cameras can capture arbitrarily dynamic scenes by either holding a single camera and moving through a range of perspectives or by setting up multiple cameras in fixed positions. However, the optimal viewing experience is typically attained by looking through the perspective of the original capture angle, which significantly restricts the freedom of motion of the viewer. Moreover, it lacks the ability to provide adequate depth perception and spatial awareness due to the loss of geometric information during the capturing process. Depth cameras with a stationary setup, such as [5, 24, 37], can be utilized to capture geometric information. However, they have a limited field of view and capture range, resulting in a highly restricted capture area. Furthermore, those cameras typically require additional devices, such as a PC, and thus lack portability. This leads to a dearth of scene diversity as a consequence.

In recent years, the increasing prevalence of iPads and iPhones equipped with LiDAR scanners has revolutionized depth capturing capabilities. This advancement has paved the way for the creation of large-scale and diverse RGB-D recordings that encompass a wide range of indoor and outdoor scenes. Notable examples of such

datasets include [38] and [3]. With the significant advancements in world tracking capabilities, it has now become feasible to track camera pose accurately on modern iOS (ARKit) and Android (ARCore) smartphones through the utilization of visual-inertial odometry techniques, which combine data from the device’s motion sensing hardware with computer vision analysis [2]. The camera pose is essential for computing a global position of the geometries captured by the depth cameras. The camera pose matrix assists in converting the depth data from camera coordinates to world coordinates. The term *spatially tracked RGB-D video* refers to a video format that combines the RGB-D and 3D camera pose information in individual frames, e.g., the bibcam video format proposed by [29]. With this new form of medium, diverse real-world dynamic scenes can be captured and potentially replayed in a comprehensive manner.

Certain approaches aim to render dynamic 3D scenes by extracting the recorded media into geometric meshes and textures. The extraction can be accomplished by processing the videos recorded by multiple cameras, such as [16], or by inferring the scene from a video captured by a single camera, as demonstrated in research [22]. These methods often rely on sophisticated algorithms and techniques to process and analyze the captured data, while visual artifacts and imperfections could still exist in the final result when observed in an immersive environment.

Neural Radiance Fields (NeRF) [20] is a recently developed 3D reconstruction and view synthesis method. This rendering algorithm is capable of rendering a photorealistic 3D volumetric scene from sparse input views. While NeRF-based approaches are effective for high-fidelity view synthesis and can even be integrated into VR systems [17, 21], these methods are less explainable and expensive to train when compared to point cloud based rendering approaches. At the same time, point clouds possess the capability to represent complex geometry and can be easily deformed and animated [5, 24, 37].

In recent years, we have witnessed the rise of point clouds as a captivating visual aesthetic, accompanied by a surge in creative applications and artistic exploration, as reviewed by [13] and [4]. The music video for Radiohead’s “House of Cards” by Aaron Koblin was created entirely with point clouds captured by a laser scanner. While a live 3D cinema, “Upending”, made by OpenEndedGroup, captured everyday objects, environments, and human bodies, then converted them into point clouds to enable exploration and analysis from unfamiliar perspectives, reflecting the viewers’ perceptual processes.

2.2 Immersive Timeline Editing and in situ Authoring

In addition to capturing and replaying, the editing of such dynamic 3D data also plays a crucial role.

Griffin et al. [9] propose the editing of panoramic as well as RGB-D videos directly in virtual reality. They propose the idea of rendering the timeline in a cylindrical or planar shape with single-track or multi-track editing functionalities. However, the timeline layout is mainly designed for panoramic videos or stationary RGB-D videos. It is not customized for spatially tracked RGB-D videos. Fouché et al. [8] propose a design space for the visualization and interaction with S4D datasets, such as time-varying 3D point clouds.

The design space includes the layout design of the timeline as a linear, convex arc of a circle, convex parabola, etc. However, this design is not intended for data with inherent meaningful spatial time points.

In recent years, we have seen an emergence of AR approaches with in situ authoring capabilities. These approaches notably reduce the requirement for frequent context switching, allowing for a more seamless and intuitive design process. This has been demonstrated in the context of interactivity design [7, 14, 39], spatial placement design [6, 23], or visualization configuration [25] on various datasets like immersive video such as 180°, 360°, RGB-D video [9, 11], volumetric video [14], abstract data [18, 25], or 3D models [39].

To enhance interactivity, [14] proposed the idea of making interactive experiences with volumetric videos by giving the user control over time. [7] propose a system for authoring AR narratives, allowing children to animate character movements. For spatial placement design, [6] allows users to place and adjust the pose of virtual cameras for filming. [23] allows users to author semantic adaptation in virtual reality. While [18] allows users to adjust the dimension of the region of interest. In the context of visualization authoring, [25] proposed the idea of configuring visualizations using HMD along with spatially-aware mobile devices.

Several approaches regarding point cloud authoring in AR/VR have emerged in recent years. Wang et al. [33] proposed a tablet-based AR system for environmental design with point clouds, allowing a rapid design iteration. Ipsita et al. [12] designed a system to ease the virtual content creation process for domain users. The system offers in situ content creation experience by allowing users to select regions of interest in scanned point clouds and attach behavioral properties to them. Both approaches are designed to work with indoor scenes and take static point clouds as input.

Spatially tracked RGB-D video holds the potential for creating narratives in space with varying visual aesthetics. However, being a relatively new form of media, there is a lack of research and tools that fully exploit the potential to author its interactivity, spatial placement, and visual appearance in immersive space in situ.

3 ANALYSIS

In order to identify desirable features of our platform and illustrate their importance, we briefly describe four potential use cases.

3.1 AR Museum Tour

A museum wishes to produce a guided tour of its collections, to be viewed in situ in the museum itself with mixed-reality headsets or tablets. In separate sessions, they record various experts as they perform guided tours, following each with a tracked RGB-D camera. Often multiple takes are captured, producing hundreds of recordings. A **content previewing and asset management** system is used to select the best clips. **Multiple takes** are cut and edited together, and **alignment methods** are used to anchor clips to real-world locations. As the clips will be viewed in situ in mixed reality, the static background is removed, leaving only foreground humans. **Spatial playback control** allows clips to be

assembled into branching spatial narratives, allowing museum visitors to choose their own paths through the museum or simply follow their virtual guides.

3.2 VR Museum Tour

Using the same material, the museum now releases a version for viewing at home. As remote users are not able to see the museum environment with augmented reality, static non-human points are extracted from the recordings and rendered with different visual styles to provide a sense of the structure of the museum. Alternatively, the recordings can be embedded in a virtual polygonal replica of the museum. However, as the museum is physically larger than the users' living room, **spatial editing** is needed to re-arrange the recordings into a more compact space, creating a form of spatial collage. This could be done by the "authors" at the museum, or the viewers themselves could be given the tools to compose their own bespoke spatial arrangements.

3.3 VR Exposure Therapy - Crowded Street

A therapist is treating a patient for agoraphobia with virtual reality exposure therapy. She would like to recreate the sensation of being in a specific crowded narrow street found in the neighborhood of the patient. After recording a stroll along said street, she uses **spatiotemporal layering** to layer different moments in time on top of one another in order to create versions of the street with varying levels of crowdedness. This is made easier by using a preview that reveals the paths of people in space. She also uses visual styles to create different levels of abstraction and realness in order to modulate the intensity of the experience for her patient.

3.4 Immersive Journalism

A journalist captures a violent demonstration, following the unfolding action over a number of hours. She then assembles a short immersive news story, selecting key events and composing them into a spatiotemporal montage. Critical to her work is the ability to rapidly preview many clips, seeking in particular moments in time with many people in the frame. For this, she uses **analytic previews** that display the number and density of people across space and time, allowing her to find moments of interest rapidly.

In the above examples, we see an overlap between the roles of "author" and "viewer", such that it is sometimes neither necessary nor possible to draw a sharp distinction between the two roles.

4 DESIGN

Motivated by the above examples, we now outline the design of our system.

4.1 Interface and Previews

During playback, tracked videos are displayed as animated 3D point clouds. During authoring in virtual reality, however, it can be challenging to interact with such point clouds, as they are constantly moving and often lack any persistent features. In addition, acquiring an overall impression of the content and spatial structure of a video is a crucial first step in crafting immersive experiences. As such, atemporal interfaces and representations of tracked videos

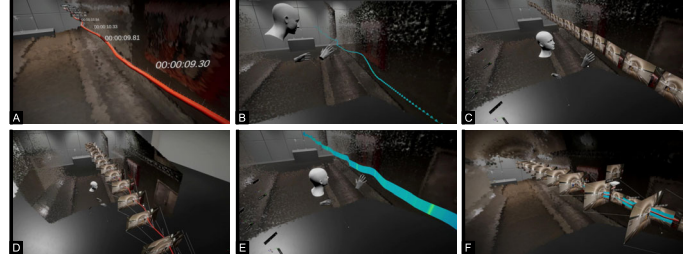


Figure 2: Camera Trajectory based visualizations showing (A) Time markings, (B) Camera directions, (C) Video content as a strip, (D) Both camera directions and video content, (E) Camera Speed. (F) Shows a combination of the visualizations from (A) to (E).

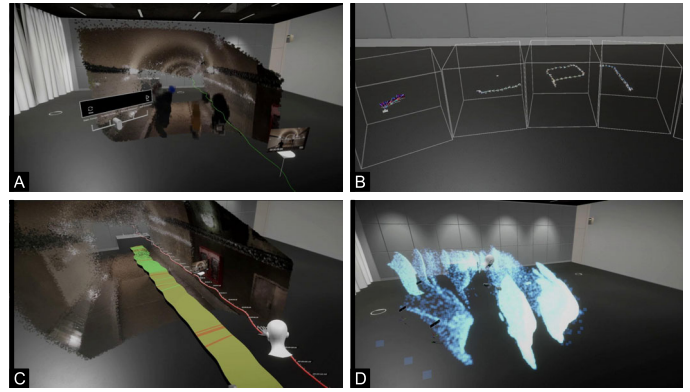


Figure 3: (A) and (B): The proposed point cloud preview and the small multiples view. (C) A color-encoded path projected from the camera trajectory shows the crowd state of individual frames. (D) 3D heatmap illustrates the location and path of humans over time.

are required. Here, we describe several methods for achieving both goals in virtual reality.

4.1.1 Camera Trajectory. As the primary interface for manipulating tracked RGB-D video, we adopt a 3D representation of the camera trajectory. Rendered as a tubular curve, beginning with the initial and ending with the final camera pose of the source video, this base interface can be optionally augmented with a number of features: a) Time markings, the demarcating passage of time in seconds. b) Camera direction, represented as mini-frusta indicating the direction of the camera over time. c) Video stills presented as a video strip along the camera trajectory. d) Video stills presented on frusta at set intervals, encoding both camera direction and video content. e) Encodings of camera speed, as color. (exemplar renderings are shown in Figure 2 (A) - (E))

4.1.2 Point Cloud Previews. In addition to the camera trajectory, for each video, static previews of the entire point cloud are extracted from the entire timeline of the video. These static previews provide an instantaneous overview of the structure and content of a video.

Methods for interactively controlling the spatial and temporal sampling are provided, with both sequential and stochastic sampling across space and time supported.

4.1.3 Point Classification and Filtering. During sampling, points are classified into different categories, allowing for class-based control of sampling rate, filtering and visual appearance, and simple visual analytics. Some point labels will have already been assigned during the capture stage and embedded in the video file itself, as in the case of the human segmentation mask provided by the ARKit. Further segmentation and labeling may be performed on the video in a pre-processing stage prior to feeding the material into Dynascope (discussed further below). Further labeling can be performed during the projection of the point into 3D space. In our current implementation, we use the following classification scheme: a point is classified as *Foreground* if it has known depth or *Background* if the depth is beyond the functional range of the depth sensor. A point is labeled as *Human* if it is known to belong to a human. In addition, user-defined 3D spatial primitives (such as a sphere or box) can be used to define 3D region labels. This is very useful for culling or giving a different visual appearance to different regions of the point cloud, such as the ground plane, walls, or when combining sub-regions of multiple recordings. When viewing a point cloud preview, the user can hide and show different classes of points in order to obtain a clearer view. Further, they can adjust the size and opacity of points belonging to different classes (or any other visual parameter as described below) and adjust the spatial and temporal sampling density of each class. (see Figure 3 (A)).

4.1.4 Small Multiples View. The camera trajectory and the point cloud preview can be displayed at a 1:1 scale or miniaturized to provide a birds-eye view. This allows for the arrangement of multiple small previews of different videos or, alternatively, multiple small previews of the same video, but with different rendering styles (see Figure 3 (B)).

4.1.5 Visual Analysis. Camera trajectories can be further augmented with encodings of time-varying information, such as audio volume, the number of people in view, or any other information that can be extracted from the videos. This allows for various visual encodings of space and time-varying data along the camera trajectory or projected onto the ground. (see Figure 3 (C)) Furthermore, point-based “heatmaps” can be used to show the density of points belonging to a particular class of points. This is particularly useful for the Human class, where a 3D or 2D heatmap (by projecting 3D points onto the ground plane) can be used to illustrate the location and path of humans over time. (see Figure 3 (D))

4.2 Appearance Authoring

During playback, a 3D point cloud is constructed using camera pose and pixel depth information to recover the 3D position of each pixel. Points can be rendered as camera-facing squares or circles, otherwise as 3D cubes or spheres. In addition, they can have variable size, opacity, color, emissivity, softness, and motion, allowing for a great deal of visual styling.

To support point cloud creation from the spatially tracked RGB-D video and allow for great flexibility during the styling process, we designed the rendering workflow shown in Figure 4. Video

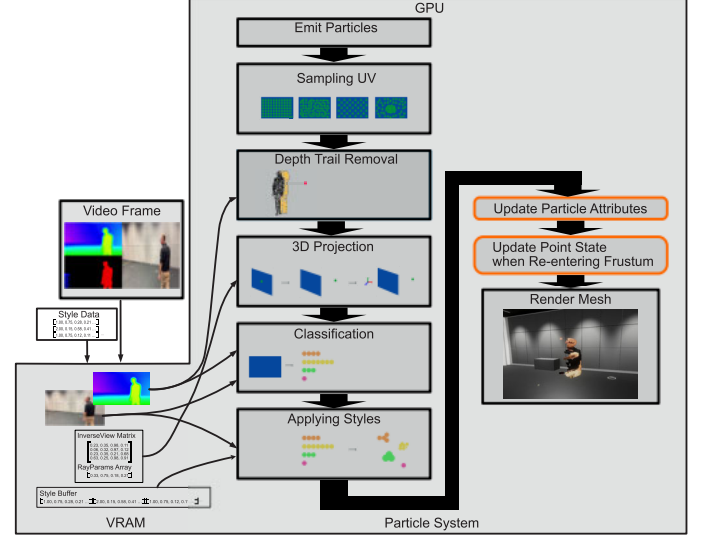


Figure 4: Point cloud rendering. Video frames are decoded and passed to the GPU as separate RGB, depth and human-mask textures. Newly created particles go through a sequence of sampling, depth trail removal, 3D projection, and classification. Existing particles are removed if their lifetime has expired or they have re-entered the frustum and been “over-painted”. All particles are then animated, aged and finally rendered with the style linked to their class.

frames are decoded into separate RGB, depth, and mask textures, allowing different pixel formats for each. Instead of hard coding the visualization styles on GPU, we treat the styles as style data. This data is kept on external storage and updated to GPU when it changes. We designed to let the newly created particles go through the rendering pipeline in order of sampling, depth trail removal, 3D projection, classification, and stylization. All particles are animated, aged, and rendered with their style applied. In the following, we elaborate on our considerations and the design details for each stage in the workflow, innovative projecting and lifetime control strategies, as well as the details regarding the in situ appearance authoring process.

4.2.1 Particle Creation and Sampling. In our current implementation, an RGB-D video has 960x1080 color pixels and 960x540 depth pixels encoded at a frame rate of 60Hz. As such, if a 3D point were created every frame for each RGB pixel, the point cloud would grow at a rate of 60 million new points per second, instantly exceeding the rendering capabilities of the system. To avoid an ever-growing number of points, we adopt policies for controlling the creation and removal of points over time. To begin, we control the particle creation rate by defining a maximum limit per video frame. During particle creation, we first select a UV coordinate in the RGB-D video using a regular grid or random spatial distribution. The selected pixel is then classified (in our case, into foreground, background, human classes), and then either accepted or rejected according to a class-specific creation probability. This allows for the creation of relatively more or less points from each category.

4.2.2 Depth-Trail Removal. Due to the limited accuracy of the mobile depth sensor and over-smoothing of the depth frame, the depth becomes smoothed across the boundary of objects, where it should be sharp and discontinuous. When rendering these points in 3D, this leads to the appearance of “depth trails” at the boundaries where the depth changes dramatically. (see Figure 5 (F)) As these trails often appear at the boundary of humans or objects, we adopt here a simple depth-gradient filtering method for trail removal. During point sampling, we discard points with gaps in depth between neighboring pixels above a certain threshold. The result is shown in Figure 5 (E).

4.2.3 3D Projection. A 3D particle is emitted for each sampled pixel that survives the class and depth-trail filtering described above. The world position of the particles is calculated by projecting the selected pixel forward according to the recorded depth, then transforming with the recorded camera pose and the user-based transform of the video trajectory in virtual space.

4.2.4 Background Projection. With the majority of RGB-D sensors, the sensing range of the RGB camera far exceeds the sensing range of the depth sensor. As a result, in outdoor scenarios, a significant proportion of the image may have unknown depth - all that is known is that the point lies somewhere beyond the sensing range of the device. We classify these as Background points. How should these points be handled? One solution is to simply discard them, which may be the ideal strategy in some contexts, such as in situ mixed reality. For virtual reality, however, removing the background points can lead to large empty regions in the virtual scene. However, if they are to be rendered, where should they be positioned? We adopt here a method of projecting background points onto a background dome. Combined with long lifetimes, over-painting, and the correct visual style, this produces a form of painted background. (see Figure 5 (C) and (D))

4.2.5 Applying Visual Styles. The combination of the parameters for size, shape, opacity, color and motion, as well as sampling rate, lifetime and overpainting rate, background projection and trail removal, all define a *visual style*. In our system, a user can define any number of visual styles, allowing interactive or programmed adjustment of any of these parameters. Most importantly, each point class in each video can be assigned a different visual style, allowing for a great deal of flexibility in adjusting the visual appearance of the composition.

4.2.6 Spatial Masks. In specific scenarios, users may need to remove or modify the appearance of specific portions of the point cloud. For example, removing the floor or highlighting a region. We provide 3D spatial primitives (such as a sphere or box) in which points can be given different visual appearances.

4.2.7 Further Visual Effects. We provide further visual effects, such as re-coloring, emissivity, and motion in the form of turbulence or gravity. These effects can be further parameterised by the movements or proximity of the user, bringing the scene to life. For example, points may float away as the user approaches or glow upon proximity.

4.2.8 Over-painting. On creation, points are assigned a lifetime, ranging from infinite to a single render frame. When a point reaches

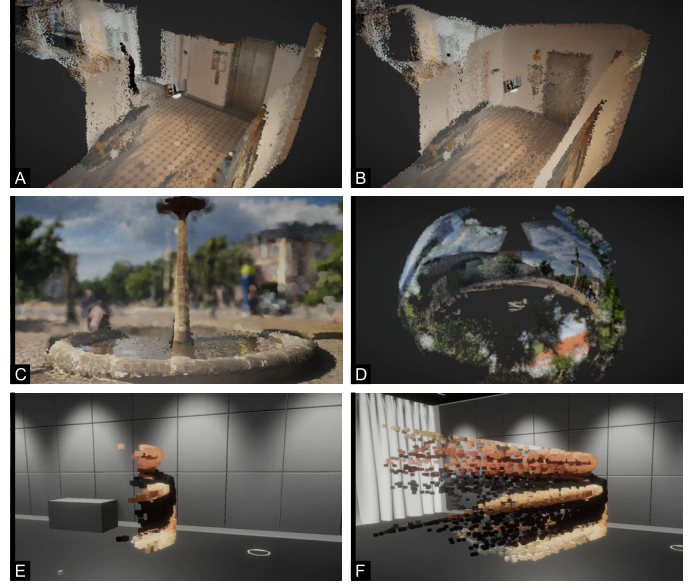


Figure 5: “Over-painting” point removal when points re-enter frustum (A) and without (B). (C) - (D) Background dome projection: points with unknown depth are projected onto a dome and rendered with a soft style. (E) - (F) Human rendering with and without trail removal.

the end of its lifetime, it is removed. As the camera moves around, it very often re-captures previously seen parts of the world. Rather than further adding points, we use a frustum-based point removal policy to replace existing points with new ones. This is achieved by shortening the lifetime of a point when it re-enters the camera frustum. The result is a form of *over-painting*, in which new points replace old ones. The lifetime reduction can be a function of points’ position in the camera frustum, allowing for the removal of points in the center at a faster rate than the edges. (see Figure 5 (A) and (B)).

4.2.9 In situ Appearance Authoring. Often visual styling can mean the difference between a chaotic mess of points and a recognizable or captivating scene. Finding the most suitable rendering style is a creative task and often requires some experimentation. Therefore, all of the above style parameters can be adjusted from within Dynascape, inside VR, using a matrix of sliders to enable fine-grain adjustments. This UI provides enhanced flexibility for authoring visual appearances while maintaining a simple interface. Any changes made to the variables are immediately applied to the visual style by updating the parameters to the rendering pipeline. By conducting the authoring process in the same location as the final display, creators can better understand the environmental conditions and context in which their content will be experienced. The specific lighting conditions, spatial layout, and overall ambiance of the space will be considered during the crafting process. In situ appearance authoring enables creators to make real-time adjustments and optimizations based on the immediate visual feedback they receive. They can assess how the content interacts with the surrounding elements, such as physical objects and architectural features.

4.3 Spatial Authoring

When composing a scene with multiple tracked RGB-D videos, methods for positioning and aligning point clouds with the real world (in MR), with the virtual world (in VR), or with other point clouds are required. Dynascope provides a number of tools to aid this.

4.3.1 Direct Manipulation. As described above, the 3D camera trajectory is the primary interface for interacting with a video. To move a video in space, the user can simply grab the trajectory with controllers or hands and position it arbitrarily. Scaling is also supported, or it can be locked to 1:1 if desired.

4.3.2 Alignment Tools. In addition to direct manipulation, we provide helpers for rapid and accurate alignment. A 3D arrow icon can be used to define the starting pose of a video, as shown in Figure 6 (A). Alignment with the real world or between multiple videos is achieved using the concept of source-target frames. For example, consider an in situ recording, in which the user now wishes to align the video with the real world. The source frame is first aligned with some well-defined region in the point cloud, such as the corner of a door or desk, and the target frame is then given the same position as the real-world door or desk. The source is then snapped to the target. Source-target alignment is particularly useful when aligning multiple recordings, as the same target can be shared with multiple source frames. (see Figure 6 (B) and (C)). Multi-video alignment can also be achieved using the QR code “clapperboards” during filming. A QR code is placed somewhere in the filming location, and care is taken to record the QR code at the beginning of each take. Whenever the QR code is in view in the video, the user need only trigger a source-target alignment as described above, and the QR code will act as a source frame.

4.3.3 Miniature Manipulation. When not displayed at a 1:1 scale, the size and position of videos can be manipulated with the help of an adjustable bounding box manipulator, as depicted in Figure 6 (E).

4.3.4 Time-varying Alignment. Although localization techniques have made significant advancements, it remains a common occurrence for the tracking of AR cameras to experience drift or tracking loss. When tracking drift occurs, aligning an in situ video with the real-world with a single static transform is not possible for all moments. To address this challenge, we provide a time-varying transformation authoring method that enables creators to make multiple adjustments to the alignment over time, as shown in Figure 6 (F). This technique involves gradually transitioning the pose of the spatial video from one position to another during playback. This can be used to correct tracking drift or equally so, creatively, to deliberately manipulate and distort the spatial structure of a scene over time.

4.4 Temporal Authoring

When involved in the crafting and playback of spatially tracked RGB-D videos, similar to 2D screen-based video editing, cutting and re-arrangement of the timeline is a fundamental requirement. Considering the benefits offered by immersive visualization and the ability to interact with body gestures in the immersive environment,

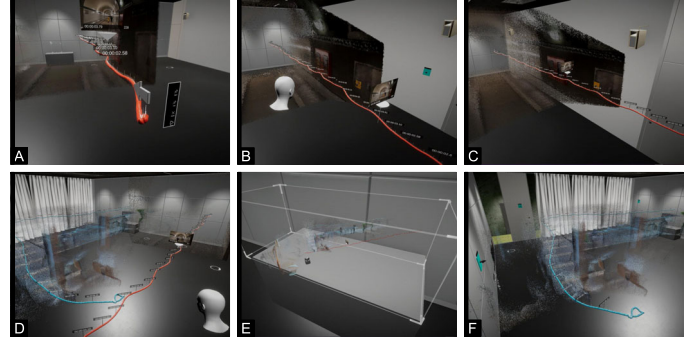


Figure 6: The proposed spatial and temporal authoring. (A) Spatial authoring with direction indicators. (B) - (C) Before and after spatial authoring with plane indicators. (D) Multiple storylines in space. (E) The scaled storylines. (F) Space alignment over time.

as well as the 3D nature of the spatially tracked RGB-D video data, it is natural to employ a spatial timeline that enables users to easily select and edit the media in space. This section introduces the concept of our spatial timeline that helps edit spatially tracked RGB-D videos in space.

4.4.1 On-Trajectory Timeline Editing. We directly render the video timeline on the camera trajectory (see Figure 2 (A)). A space-time cursor, constrained to the camera trajectory, allows users to seek in time and space by simply dragging the cursor. Users can view time stamps and seek to exact frames directly. Cutting is achieved by pressing a button after setting in and out points on the timeline.

4.4.2 Body-Anchored Time Control. In addition to directly manipulating the time cursor or seeking a time point by touching the trajectory, Dynascope offers automatic time control based on body position. Playback speed is modulated by the location of the user relative to the camera trajectory, allowing the user to control time by simply walking. For videos with audio, however, dynamic time scaling may be undesirable. In this case, videos are constrained to play in real-time or to pause entirely, depending on the proximity of the viewer. Interfaces are provided to adjust the parameters and help with switching the automatic time control on or off.

4.4.3 Dual Space/Time Representation. A challenge arises with the trajectory timeline view described above when recordings contain no or very little camera motion. In this case, the camera trajectory ceases to be a useful method of navigating time. For this reason, we augment the trajectory view with a 2D purely temporal timeline, similar to that found in a traditional video editing tool. The user can manipulate the videos in both representations. (see Figure 7)

4.4.4 Multi-Clip Editing. Similar to traditional multi-clip video editing, multiple clips can be listed together on a 2D UI and interactively adjusted. However, there is no concept of layers, as all videos can be played simultaneously in an immersive environment without requiring layer composition. The idea of the main timeline is still valid since it is possible to adjust the starting/ending/span time for individual video clips. In Figure 7, clips are represented by colored bars with starting and ending holders. The color matches

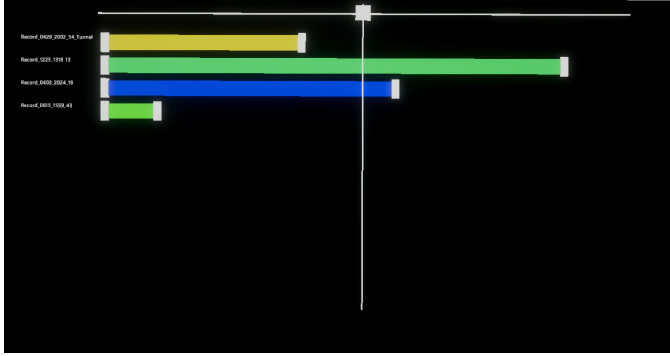


Figure 7: Multiple clips aligned on the 2D timeline.

the color of the corresponding 3D camera trajectory. Similar to traditional multi-clip video editing, the bar itself can be grabbed back and forth along the timeline, while the holders can be used to adjust the time span of the clip. Cutting and pasting are implemented by using multiple in-out markers on the timeline. Periods of time falling outside of a pair of in-out markers are considered "cut" and skipped during playback. After cutting, the camera trajectory will become discontinuous, but the overall dimension from the video start to the end will be maintained. To allow different spatial positioning of clips cut from the same video, the video is duplicated. When combining a cut clip with another clip, the starting point of the source clip and the ending point of the target clip will be concatenated. Crossfading is not supported at the moment, but could be implemented with a simple increase or decrease in point size or opacity at the beginning at the end of each clip.

4.4.5 Narrative Branching. Designing storylines that visitors can engage with by walking along designated paths introduces a captivating application of the spatially tracked RGB-D videos. Multiple videos can be concatenated in space, forming spatial narratives. Here we extended the concept to allow narrative branching. While two clips aligned in time would normally play simultaneously, branching allows dynamic activation and deactivation of sequences of clips, depending on the actions of the viewer. A prototypical application of narrative branching would be the user selecting a door to enter or a path to follow at a junction. We offer two possibilities to create branches based on multiple clips in our prototype. After the user figures out the source and the target clip, one way to create the branching storyline is to drag the source clip directly toward a certain point on the target clip. However, this can be less accurate and becomes tedious when the user has to hold the source clip over a long distance. The other way is to first create a branching point with the above-mentioned space-time cursor and then snap the starting point of the source clip to it.

5 IMPLEMENTATION

5.1 Overview

Dynascope was implemented using the Unity3D Engine and MRTK framework for Meta Quest 2 devices. Rendering is streamed from the PC to the head-mounted display through the Airlink [19]. Both virtual reality and video-passthrough mixed reality are supported.

Oculus "spatial anchors" are used for virtual to real world alignment. The source code will be available on our webpage.

5.2 Capture and Processing

We employed the Bibcam format [29] to record spatially tracked RGB-D videos. This format allows us to encode color frames, depth frames, and camera pose information into individual frames. The camera pose information is obtained from ARKit [2] in real-time and encoded to each frame as pixel barcodes.

To best illustrate our concept, we implemented a process to extract human entity information, specifically the number of people present, from our recorded videos on a frame-by-frame basis. We obtained a human stencil map [31] from ARKit [2] and kept it along with the color and depth frame in the Bibcam format [29], which served as the basis for extracting human-related information from the video frames. We then extracted the number of people from the human stencil map with the contour counting algorithm.

A GPU-accelerated method for extracting and saving to disk key-frame RGB images, point cloud previews, and camera trajectories was implemented. This pre-processing step is not necessary, but when performed, it enables instant loading and rendering of the entire timeline of a video, eliminating the need to playback from beginning to end every time a video asset is introduced.

5.3 Visualizations

5.3.1 Camera Trajectories. Camera trajectories are rendered as polygonal tubes. Downsampling and smoothing of the camera trajectory are performed with accumulated distance, time, and curvature thresholds. In order to visualize semantic information, we extract and map it to colors, creating an array in VRAM for rendering. Additionally, to enable texture previewing, we store the extracted frames in VRAM as a texture array.

5.3.2 Point Rendering and Visual Styles. The Unity "Visual Effects Graph" GPU-accelerated particle system is adopted for point cloud rendering. Each point stores position, age, color, and class id, but no information on visual appearance. Instead, during rendering, the class id is used to index style parameters stored in a separate buffer. This indirection not only greatly reduces VRAM usage but allows for efficient real-time manipulation of visual styles.

5.4 Performance

Our system is designed to efficiently generate point clouds from multiple video sources simultaneously. We are able to render up to 2 million points in total with around 70 FPS with the Oculus Quest 2 headset.

6 DEMONSTRATION

6.1 Virtual Tour Guide

To best highlight the value of Dynascope, we used an iPad to capture a guided tour of a museum. Binaural in-ear microphones are worn by the camera operator to capture an immersive soundscape. Considering the significant difference in size between the museum and the lab, as well as the extensive duration of the recording, we creatively manipulated **both space and time** to adapt the recording to fit within the physical surroundings of the lab while preserving

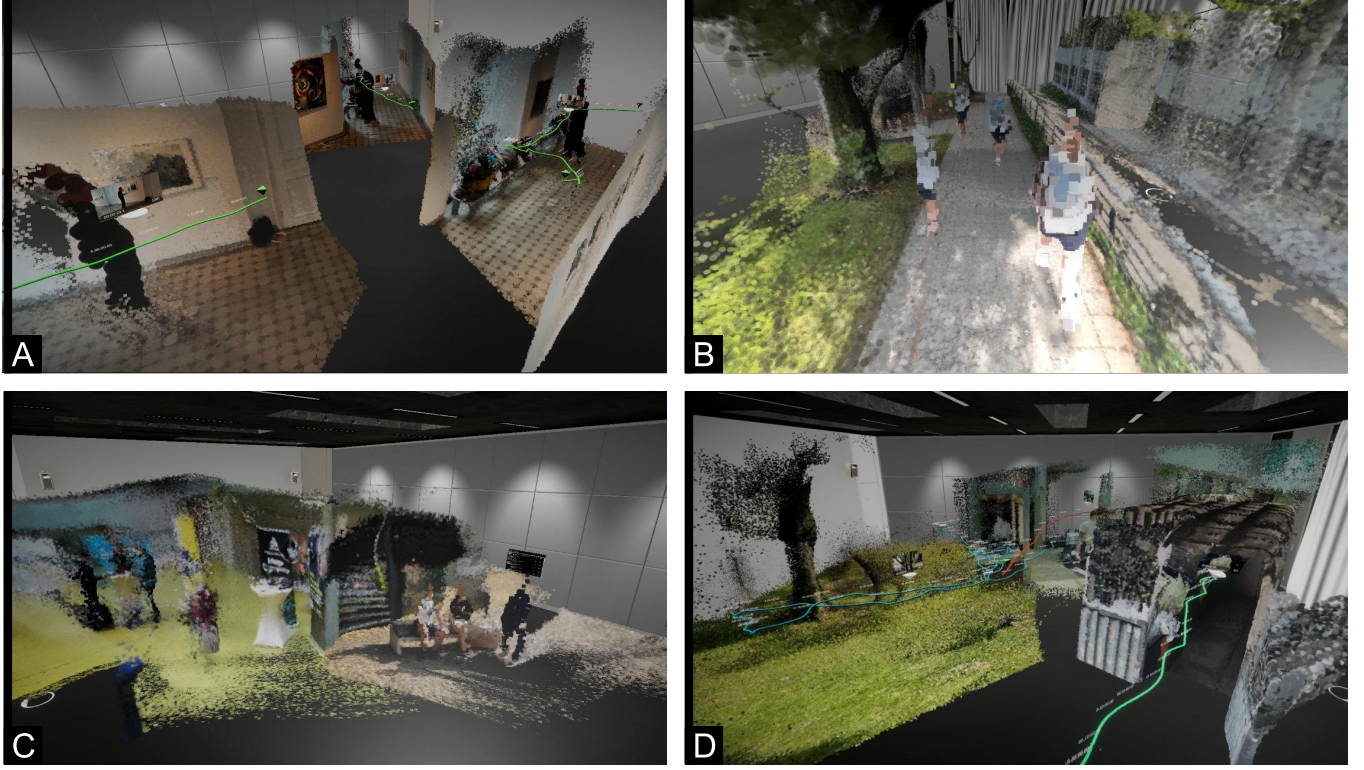


Figure 8: The exemplary applications of Dynascope. (A) The immersive replay of a virtual tour with manipulated space and time. (B) A street view with the repeated (4 times) rendering of a pedestrian by manipulating time. (C) Virtual collage that composites a dynamic indoor and outdoor scene. (D) Three animated point clouds composited together to create storylines.

the key attractions and noteworthy aspects. With the help of the implemented prototype, we processed the recorded video and got semantic information extracted.

To get an insight into the interesting segments in the recorded video, and in light of the fact that the museum is much larger than the lab in space, we used the **minitiarised overview** to place the entire clip onto the top of a table, and then used sliders to highlight and remove humans, background and foreground points. After identifying interesting places, we cut the video into small segments with the **On-Trajectory Timeline Editing** tool. Afterward, we restored the scale of the clips to their original state, placed the segments in the virtual space, and used the spatial authoring tools to arrange the clips in space. Finally, we used visual styles to apply various rendering strategies to the foreground and human points to enhance the overall visual appearance as well as the feeling of presence. During playback, **Body-Anchored Time Control** allows viewers to simply follow the tour guide. The resulting rendering, showcasing the static environment and dynamic human points, can be seen in Figure 8.

6.2 Street View

To best showcase the potential of Dynascope, we filmed a few people walking along the street with three iOS devices with LiDAR capability and synthesized a busy street by manipulating the time.

After preprocessing, we started by enabling the **Camera Trajectory Based Preview** to render a path on the ground, colorized based on the human entity information present in the video. This visualization allowed us to see the trajectory of the camera movement and better understand the crowd dynamics. We then exploited the **alignment tools** to align all the videos and position them in the lab space. By setting an In and Out point, we isolated the desired segments for playback. Subsequently, we duplicated the people present in those clips and synthesized a busy street full of people by adjusting visual styles for individual segments. Human points appeared dynamic, while environment points had a longer lifetime and appeared static. Figure 8 (B) shows an exemplary rendering of a street view.

6.3 Virtual collage

To highlight the creative potential of spatiotemporal collage, we demonstrate the merging of indoor and outdoor recordings.

We started by previewing a list of recordings with the proposed **Camera Trajectory Based Preview** and the **Point Cloud Based Preview** in a scaled form with the **Small-multiples View** (as shown in Figure 3 (B)). After choosing desired clips, a bustling outdoor setting, and a vibrant indoor scene, we placed the indoor scene with the direction indicator and snapped the outdoor scene to the indoor scene with the plane indicator. When replaying the outdoor scene, we also exploited the **Time-varying Alignment**

feature to adjust the alignment multiple times. Its transformation updated gradually over time. Upon finalizing the placement of the spatial video, we used visual styles to adjust the visual appearance of the final rendering.

6.4 Crafting Storylines

To illustrate the potential of Dynascope in crafting storylines in space, we decided to choose clips from our recordings, place and concatenate them in the lab space.

With the **previewing** feature, we chose a canteen clip as the main branch and plan to involve clips captured in a green area as well as a waste area as side branches of the storyline. The rendering changes when viewers walk in different directions, they would see the green area when choosing environmentally friendly products and see the waste area when walking to environmentally damaging products in the canteen. The storyline looks similar to Figure 6 (F) in space. After **Spatial Authoring** and **Appearance Authoring** process, we chose the In and Out points with the **On-Trajectory Timeline Editing** tool and used the **Narrative Branching** feature to create branching points in the main branch and snapped side branches onto it. This offers the opportunity of concatenating branches precisely in space.

7 DISCUSSION AND FUTURE WORK

In this section, we reflect on our immersive approach to authoring and playback of spatially tracked RGB-D videos. We also discuss potential improvements in various areas, such as data capturing, processing, and aligning.

Potentials of In Situ Authoring. While the number of features for authoring is limited in our prototype, we could see the potential of Dynascope to support in situ authoring of spatially tracked RGB-D videos in an immersive environment. Dynascope helps users to obtain a comprehensive understanding of the end outcomes by allowing them to edit media content directly in the immersive environment. When working directly in the same space where the video will be viewed, the surrounding environment aids in serving as a vital reference for spatial-temporal and appearance authoring.

Video Editing. While exploring the spatial and temporal editing of the spatially tracked RGB-D video, our primary emphasis lies only on the fundamental aspects of video editing. Therefore, this paper does not cover additional video editing features, such as shifting and crossfading, within its scope.

Data Source. Furthermore, apart from the depth cameras integrated into iOS devices, Dynascope has the potential to accommodate various other depth cameras with world tracking capability. Notably, a recent version of the ZED camera [28] introduces novel functionalities that hold great potential in capturing semantic information while acquiring spatially tracked RGB-D videos. With the advancement of visual localization algorithms, spatial information from regular RGB-D videos could be precisely extracted and thus will be ready to be used in our system. With the rapid progress in depth estimators and AI techniques, achieving exact depth information estimation is becoming increasingly feasible. This advancement opens up the potential to transform a regular

video into a spatially tracked video, allowing users to visualize and author it with the capabilities of Dynascope.

RGB-D Video Processing. Dynascope proposes the idea of encoding semantic information to the camera trajectory as well as the camera path on the ground. Currently, we extract the number of people in individual frames, implemented with a contour counting algorithm using OpenCV. The extraction of semantic information has been intensively researched in recent years in the field of computer vision and scene understanding. We can expect comprehensive possible information will be easily accessible in the future. That information can then be encoded into the proposed visualization channels in this paper.

Alignment Between Multiple Cameras. Another potential issue is that the virtual objects might not be perfectly aligned amid the camera drift while creating storylines with overlapped virtual objects in space. ICP-based alignment algorithm, e.g. [36], can be involved in the future to help with the alignment from one point cloud to the other. Our timeline-based alignment strategy can also benefit clip-to-clip alignment once we get the aligned data from those algorithms.

Rendering Fidelity. The capturing and rendering of point clouds in high fidelity remains a challenging task. Nonetheless, even higher resolutions and greater accuracy in depth capturing can be expected with the development of sensor techniques in the future. Additionally, techniques that have shown success in improving visuals for point clouds, such as [26, 27], can be integrated. Furthermore, NeRF-based rendering techniques like [35, 40] could serve as an effective final rendering step or be integrated into the appearance authoring process. Color information from the source RGB-D video can also be utilized for view synthesis.

8 CONCLUSION

In this work, we proposed an immersive approach to the authoring of dynamic scenes captured with tracked RGB-D cameras. The tool supports both mixed and virtual reality, and in situ and ex situ authoring and viewing. We proposed concepts and methods for previewing clips, performing spatial and temporal editing, and controlling visual appearance. We developed rendering strategies to render foreground, background, and human as point clouds from RGB-D video in 3D to help with the authoring process. We also implemented an interactive prototype and conducted a walk-through session to show the value of our work. We believe that our research makes a valuable contribution to the in situ authoring and playback of dynamic 3D scenes. We hope that our findings inspire further exploration and advancements in this area.

REFERENCES

- [1] Apple. 2020. *Apple unveils new iPad Pro with breakthrough LiDAR Scanner and brings trackpad support to iPadOS*. Retrieved June 13, 2023 from <https://www.apple.com/newsroom/2020/03/apple-unveils-new-ipad-pro-with-lidar-scanner-and-trackpad-support-in-ipados/>
- [2] Apple. 2021. *Understanding World Tracking - Discover features and best practices for building rear-camera AR experiences*. Retrieved June 13, 2023 from https://developer.apple.com/documentation/arkit/configuration_objects/understanding_world_tracking
- [3] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Yuri Feigin, Peter Fu, Thomas Gebauer, Daniel Kurz, Tal Dimry, Brandon Joffe, Arik Schwartz, and Elad Shulman. 2021. ARKitScenes: A Diverse Real-World Dataset For 3D Indoor Scene

- Understanding Using Mobile RGB-D Data. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*. https://openreview.net/forum?id=tjZjv_qh_CE
- [4] Paul Chapman (Ed.). 2017. *Art of the Point Cloud*. Wild Harbour Books.
- [5] Anargyros Chatzitofis, Leonidas Saroglou, Prodromos Boutis, Petros Drakoulis, Nikolaos Zioulis, Shishir Subramanyam, Bart Kevelham, Caecilia Charbonnier, Pablo Cesar, Dimitrios Zarpalas, et al. 2020. HUMAN4D: A Human-Centric Multimodal Dataset for Motions and Immersive Media. *IEEE Access* 8 (2020), 176241–176262.
- [6] Subramanian Chidambaram, Sai Swarup Reddy, Matthew Rumble, Ananya Ipsita, Ana Villanueva, Thomas Redick, Wolfgang Stuerzlinger, and Karthik Ramani. 2022. EditAR: A Digital Twin Authoring Environment for Creation of AR/VR and Video Instructions from a Single Demonstration. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 326–335. <https://doi.org/10.1109/ISMAR55827.2022.00048>
- [7] Stephanie Claudino Daffara, Anna Brewer, Balasaravanan Thoravi Kumaravel, and Björn Hartmann. 2020. Living Paper: Authoring AR Narratives Across Digital and Tangible Media. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (CHI EA '20). Association for Computing Machinery, New York, NY, USA, 1–10. <https://doi.org/10.1145/3334480.3383091>
- [8] Gwendal Fouché, Ferran Argelaguet Sanz, Emmanuel Faure, and Charles Kervrann. 2022. Timeline Design Space for Immersive Exploration of Time-Varying Spatial 3D Data. In *Proceedings of the 28th ACM Symposium on Virtual Reality Software and Technology* (Tsukuba, Japan) (VRST '22). Association for Computing Machinery, New York, NY, USA, Article 21, 11 pages. <https://doi.org/10.1145/3562939.3565612>
- [9] Ruairi Griffin, Tobias Langlotz, and Stefanie Zollmann. 2021. 6DIVE: 6 Degrees-of-Freedom Immersive Video Editor. In *Frontiers in Virtual Reality*.
- [10] Carsten Griwodz, Simone Gasparini, Lilian Calvet, Pierre Gurdjos, Fabien Castan, Benoit Maujean, Gregoire De Lillo, and Yann Lanthony. 2021. Alice-Vision Meshroom: An Open-Source 3D Reconstruction Pipeline. In *Proceedings of the 12th ACM Multimedia Systems Conference* (Istanbul, Turkey) (MM-Sys '21). Association for Computing Machinery, New York, NY, USA, 241–247. <https://doi.org/10.1145/3458305.3478443>
- [11] Robin Horst, Savina Diez, and Ralf Dörner. 2019. Highlighting Techniques for 360° Video Virtual Reality and Their Immersive Authoring. In *Advances in Visual Computing*, George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Daniela Ushizima, Sek Chai, Shinjiro Sueda, Xin Lin, Aidong Lu, Daniel Thalmann, Chaoli Wang, and Panpan Xu (Eds.). Springer International Publishing, Cham, 515–526.
- [12] Ananya Ipsita, Runlin Duan, Hao Li, Subramanian C, Yuanzhi Cao, Min Liu, Alexander J Quinn, and Karthik Ramani. 2023. The Design of a Virtual Prototyping System for Authoring Interactive VR Environments from Real World Scans. *Journal of Computing and Information Science in Engineering* (07 2023), 1–18. <https://doi.org/10.1115/1.4062970> arXiv:<https://asmedigitalcollection.asme.org/computingengineering/article-pdf/doi/10.1115/1.4062970/7025216/jcise-23-1217.pdf>
- [13] Lucija Ivšic, Nina Rajcic, Jon McCormack, and Vince Dziekan. 2022. The Art of Point Clouds: 3D LiDAR Scanning and Photogrammetry in Science & Art. In *10th International Conference on Digital and Interactive Arts (ARTECH 2021)*. Association for Computing Machinery, New York, NY, USA, 1–8. <https://doi.org/10.1145/3483529.3483702>
- [14] Qiao Jin, Yu Liu, Puqi Zhou, Bo Han, Svetlana Yarosh, and Feng Qian. 2023. Volumlive: An Authoring System for Adding Interactivity to Volumetric Video. In *2023 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. 569–570. <https://doi.org/10.1109/VRW58643.2023.00127>
- [15] Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Godisart, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. 2019. Panoptic Studio: A Massively Multiview System for Social Interaction Capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 1 (Jan. 2019), 190–204. <https://doi.org/10.1109/TPAMI.2017.2782743> Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence
- [16] Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Scott Godisart, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. 2017. Panoptic Studio: A Massively Multiview System for Social Interaction Capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).
- [17] Ke Li, Tim Rolf, Susanne Schmidt, Reinhard Bacher, Simone Frintrop, Wim Leemans, and Frank Steinicke. 2023. Bringing Instant Neural Graphics Primitives to Immersive Virtual Reality. In *2023 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. 739–740. <https://doi.org/10.1109/VRW58643.2023.00212>
- [18] Weizhou Luo, Zhongyuan Yu, Rufat Rzaev, Marc Satkowski, Stefan Gumhold, Matthew McGinity, and Raimund Dachselt. 2023. PEARL: Physical Environment based Augmented Reality Lenses for In-Situ Human Movement Analysis. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) (CHI '23, 381). Association for Computing Machinery, New York, NY, USA, 15 pages. <https://doi.org/10.1145/3544548.3580715>
- [19] Meta. 2021. *Introducing Oculus Air Link, a Wireless Way to Play PC VR Games on Oculus Quest 2, Plus Infinite Office Updates, Support for 120 Hz on Quest 2, and More*. Retrieved June 13, 2023 from <https://www.meta.com/en-gb/blog/quest/introducing-oculus-air-link-a-wireless-way-to-play-pc-vr-games-on-oculus-quest-2-plus-infinite-office-updates-support-for-120-hz-on-quest-2-and-more/>
- [20] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. <https://doi.org/10.48550/arXiv.2003.08934> [cs]
- [21] NVIDIA. 2023. Turn 2D Images into Immersive 3D Scenes with NVIDIA Instant NeRF in VR. <https://developer.nvidia.com/blog/turn-2d-images-into-immersive-3d-scenes-with-nvidia-instant-nerf-in-vr/>
- [22] Rohit Pandey, Anastasia Tkach, Shuoran Yang, Pavel Pidlipskyi, Jonathan Taylor, Ricardo Martin-Brualla, Andrea Tagliasacchi, George Papandreou, Philip L. Davidson, Cem Keskin, Shahram Izadi, and Sean Ryan Fanello. 2019. Volumetric Capture of Humans With a Single RGBD Camera via Semi-Parametric Learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16–20, 2019*. Computer Vision Foundation / IEEE, 9709–9718. <https://doi.org/10.1109/CVPR.2019.00994>
- [23] Xun Qian, Fengming He, Xiyun Hu, Tianyi Wang, Ananya Ipsita, and Karthik Ramani. 2022. ScalAR: Authoring Semantically Adaptive Augmented Reality Experiences in Virtual Reality. In *CHI Conference on Human Factors in Computing Systems*. ACM, New Orleans LA USA, 1–18. <https://doi.org/10.1145/3491102.3517665>
- [24] Holger Regenbrecht, Katrin Meng, Arne Reepen, Stephan Beck, and Tobias Langlotz. 2017. Mixed Voxel Reality: Presence and Embodiment in Low Fidelity, Visually Coherent, Mixed Reality Environments. In *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 90–99. <https://doi.org/10.1109/ISMAR.2017.26>
- [25] Marc Satkowski, Weizhou Luo, and Raimund Dachselt. 2021. Towards In-situ Authoring of AR Visualizations with Mobile Devices. In *2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. 324–325. <https://doi.org/10.1109/ISMAR-Adjunct54149.2021.00073>
- [26] Patric Schmitz, Timothy Blut, Christian Mattes, and Leif Kobbelt. 2020. High-Fidelity Point-Based Rendering of Large-Scale 3-D Scan Datasets. *IEEE Computer Graphics and Applications* 40, 3 (2020), 19–31. <https://doi.org/10.1109/MCG.2020.2974064>
- [27] Markus Schütz, Bernhard Kerbl, and Michael Wimmer. 2022. Software Rasterization of 2 Billion Points in Real Time. *Proc. ACM Comput. Graph. Interact. Tech.* 5, 3, Article 24 (jul 2022), 17 pages. <https://doi.org/10.1145/3543863>
- [28] stereolabs. 2023. *ZED X Smartest stereo camera for the toughest tasks*. Retrieved June 13, 2023 from <https://www.stereolabs.com/zed-x/>
- [29] Keijiro Takahashi. 2021. *Bibcam - Burnt-in barcode metadata camera*. Retrieved June 13, 2023 from <https://github.com/keijiro/Bibcam>
- [30] Paweł Tysiac, Anna Sieńska, Marta Tarnowska, Piotr Kedzior, and Marcin Jagoda. 2023. Combination of terrestrial laser scanning and UAV photogrammetry for 3D modelling and degradation assessment of heritage building based on a lighting analysis: case study—St. Adalbert Church in Gdansk, Poland. 11, 1 (2023), 53. <https://doi.org/10.1186/s40494-023-00897-5>
- [31] Unity. 2021. *ARKit Occlusion*. Retrieved June 13, 2023 from <https://docs.unity3d.com/Packages/com.unity.xr.arkit@5.1/manual/arkit-occlusion.html>
- [32] Ronja Wagner, Ole Wegen, Daniel Limberger, Jürgen Döllner, and Matthias Trapp. 2022. A Non-Photorealistic Rendering Technique for Art-directed Hatching of 3D Point Clouds. In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022) - GRAPP*. INSTICC, SciTePress, 220–227. <https://doi.org/10.5220/0010849500003124>
- [33] Zeyu Wang, Cuong Nguyen, Paul Asente, and Julie Dorsey. 2023. PointShopAR: Supporting Environmental Design Prototyping Using Point Cloud in Augmented Reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) (CHI '23). Association for Computing Machinery, New York, NY, USA, Article 34, 15 pages. <https://doi.org/10.1145/3544548.3580776>
- [34] O. Wegen, J. Döllner, R. Wagner, D. Limberger, R. Richter, and M. Trapp. [n. d.]. Non-Photorealistic Rendering of 3D Point Clouds for Cartographic Visualization. 5 ([n. d.]), 161. <https://doi.org/10.5194/ica-abs-5-161-2022>
- [35] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. 2023. Point-NeRF: Point-based Neural Radiance Fields. <https://doi.org/10.48550/arXiv.2201.08845> arXiv:2201.08845 [cs]
- [36] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. 2016. Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 11 (2016), 2241–2254. <https://doi.org/10.1109/TPAMI.2015.2513405>
- [37] Meng Zhang, Wenxuan Guo, Bohao Fan, Yifan Chen, Jianjiang Feng, and Jie Zhou. 2023. A Flexible Multi-view Multi-modal Imaging System for Outdoor Scenes. *CoRR* abs/2302.10465 (2023). <https://doi.org/10.48550/arXiv.2302.10465> arXiv:2302.10465

- [38] Haojie Zhao, Junsong Chen, Lijun Wang, and Huchuan Lu. 2023. ARKitTrack: A New Diverse Dataset for Tracking Using Mobile RGB-D Data. In *CVPR*.
- [39] Zhengzhe Zhu, Ziyi Liu, Tianyi Wang, Youyou Zhang, Xun Qian, Pashin Farsak Raja, Ana Villanueva, and Karthik Ramani. 2022. MechARspace: An Authoring System Enabling Bidirectional Binding of Augmented Reality with Toys in Real-time. In *The 35th Annual ACM Symposium on User Interface Software and Technology*. ACM, Bend OR USA, 1–16. <https://doi.org/10.1145/3526113.3545668>
- [40] D. Zimny, T. Trzciński, and P. Spurek. 2022. Points2NeRF: Generating Neural Radiance Fields from 3D Point Cloud. <https://doi.org/10.48550/arXiv.2206.01290> arXiv:2206.01290 [cs]