ReCapture: AR-Guided Time-lapse Photography

Ruyu Yan
ry233@cornell.edu
Cornell University
Ithaca, New York, USA

Jiatian Sun
jiatians@cs.cornell.edu
Cornell University
Ithaca, New York, USA

Longxiulin Deng
ld469@cornell.edu
Cornell University
Ithaca, New York, USA

Abe Davis
abedavis@cornell.edu
Cornell University
Ithaca, New York, USA

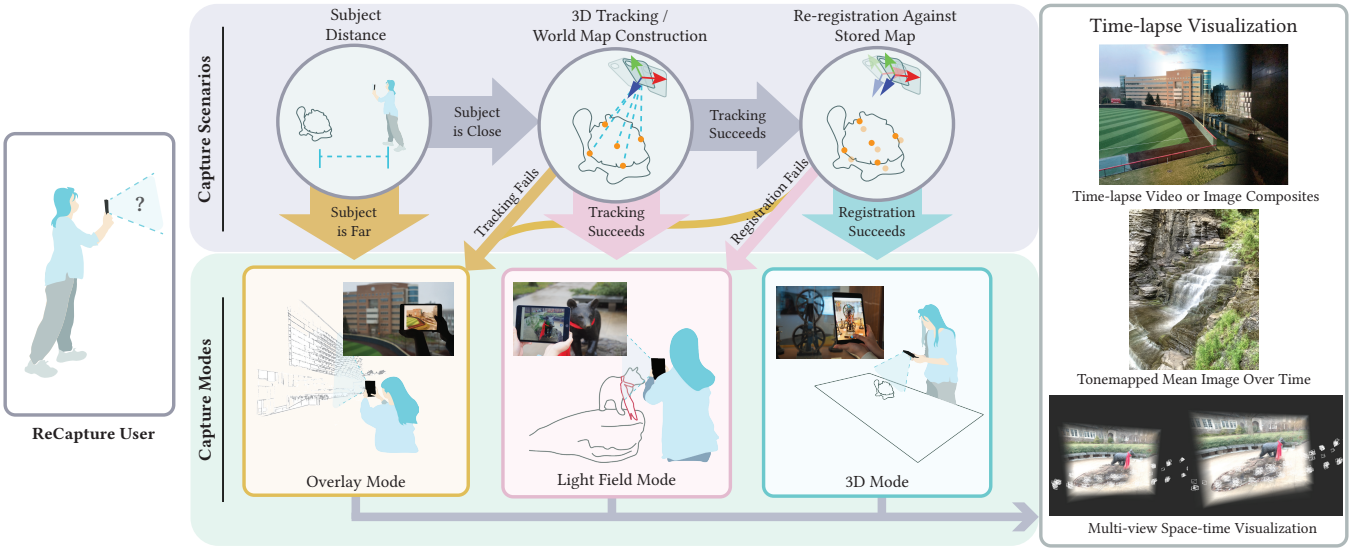

Figure 1: ReCapture provides three different capture modes to facilitate robust hand-held time-lapse capture across a wide variety of scenarios. Each mode relies on different tracking assumptions. Overlay mode (bottom left) works well for distant subjects and requires no tracking information. 3D Mode (bottom right) provides the most precise guidance, but requires both tracking and re-registration to succeed at capture time. Light Field Mode (bottom middle) requires tracking but not re-registration and is designed to help when the precision of recaptured views is important but unknown at capture time. We also explore static and interactive visualizations of captured time-lapse data (right).

ABSTRACT

We present *ReCapture*, a system that leverages AR-based guidance to help users capture time-lapse data with hand-held mobile devices. ReCapture works by repeatedly guiding users back to the precise location of previously captured images so they can record time-lapse videos one frame at a time without leaving their camera in the scene. Building on previous work in computational re-photography, we combine three different guidance modes to enable parallel hand-held time-lapse capture in general settings. We demonstrate the versatility of our system on a wide variety of subjects and scenes captured over a year of development and regular use, and explore different visualizations of unstructured hand-held time-lapse data.

CCS CONCEPTS

• **Human-centered computing** → **User interface programming**.

KEYWORDS

Augmented Reality, rephotography, time-lapse vidoes, view synthesis, light field reconstruction

ACM Reference Format:
Ruyu Yan, Jiatian Sun, Longxiulin Deng, and Abe Davis. 2022. ReCapture: AR-Guided Time-lapse Photography. In *The 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22), October 29-November 2,*

UIST '22, October 29-November 2, 2022, Bend, OR, USA

ACM ISBN 978-1-4503-9320-1/22/10…$15.00
https://doi.org/10.1145/3526113.3545641

2022, Bend, OR, USA. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3526113.3545641

1 INTRODUCTION

Time-lapse offers a powerful way to visualize changes that happen slowly over time, but recording time-lapse video is challenging; it traditionally requires leaving a camera pointed at the subject for an extended period, during which even slight camera movement can have a significant impact on the resulting video. Most time-lapse is recorded by fixing the camera to a tripod left in the scene throughout the capture. This strategy can be effective, but with several limitations. First, it requires a dedicated camera and tripod for capture. Second, ensuring the setup will not be disturbed may be impossible in uncontrolled or public settings. And third, the camera and tripod remain occupied throughout the process, which renders them unavailable for other uses. Furthermore, recording multiple perspectives of a scene (e.g., for 3D scene reconstruction) requires duplicating this setup, which compounds the cost and inconvenience of capture. These challenges contribute to a relative scarcity of time-lapse data and make multi-view time-lapse exceedingly rare.

Our work explores an alternative approach to time-lapse capture that leverages Augmented Reality (AR)-based guidance to help users repeatedly photograph subjects from consistent viewpoints over time. Instead of leaving their camera in a scene, users are free to take it with them whenever frames are not actively being recorded. When they return to the site of an ongoing time-lapse later on, our system helps guide them back to the precise locations of any previously recorded images to recapture additional time-lapse frames. In this way, users can build time-lapse video frame-by-frame with common hand-held devices. Our approach offers several advantages: it uses ubiquitous hardware; lets users capture a near-arbitrary number of time-lapse videos in parallel on a single device; and since nothing needs to be left in the scene, users can more easily record time-lapse in public and other uncontrolled environments. Our work is, to our knowledge, the first to explore parallel hand-held time-lapse capture, which we believe will enable an exciting range of new applications spanning personal photography, historical documentation, scientific fieldwork, and much more.

1.1 Key Challenges & Contributions

While our target application is new, the approach we take to hand-held time-lapse capture builds on previous exploration of closely related problems. In particular, our core contributions are perhaps best understood in the context of other work on computational re-photography. Previous work in this space has focused on recapturing individual images for a limited range of subjects (see Section 2). Our work extends this problem to frequent, repeated recapture under much more general settings, which presents significant new challenges:

Robustness & Versatility: Subjects often undergo substantial changes in both appearance and geometry over the course of a time-lapse. Notably, many common phenomena—e.g., plant growth, snowfall, or the transition from day to night—can often cause image-based registration to fail even for image pairs taken from the exact same location. This leads to a range of different tracking and re-registration scenarios that place different limits on the ability to localize a user's camera. At the same time, some subjects are more sensitive to the consistency of recaptured views than others. Together, these factors create a landscape of different accuracy demands and tracking conditions that can arise during capture.

Convenience: Unlike single-image re-photography, time-lapse re-photography requires frequent action from the user over extended periods of time. This makes the speed and convenience of capture a critical consideration. For example, design decisions that reduce time to capture (the time it takes a user to open the application and recapture one image of a given target) can have a significant impact on how often users choose to record new images, and in many cases, capturing more images can outweigh the importance of recapturing individual images more accurately. This forces us to balance the goal of facilitating very precise re-photography against the often competing goal of encouraging frequent capture.

To better understand these challenges, we spent over a year iterating on the design of ***ReCapture***, our iOS mobile app for time-lapse re-photography. At the time of writing, we have used ReCapture to re-photograph thousands of images spanning over a hundred different subjects. Key to making this possible was the observation that, for reasons related to the challenges mentioned above, different guidance strategies work better in different capture scenarios. In this paper, we outline the various factors that impact capture, relate those factors the selection of user guidance strategies, and describe how these observations motivate the design of our mobile application. Our results and analysis provide compelling evidence of the potential for systems like ReCapture across a wide variety of applications.

1.2 Overview

Section 3 describes how different time-lapse subjects and capture conditions give rise to scenarios that call for different user guidance strategies. In Section 4 we describe how we designed ReCapture to address a broad range of such scenarios, and in Sections 5-6 we present a user study to validate key aspects of that design. Section 7 describes different ways to visualize captured data and presents several results. Section 8 describes our own observations using ReCapture and discusses other high-level takeaways from our work. Our results are best appreciated by viewing the video content on our project website, where readers can also find more information about our work and the ReCapture app.

2 RELATED WORK

Guidance for 2D Photography: Several works have explored AR-based guidance for applications in photography. Adams et al. [1] provide live feedback for panoramic image acquisition. Rawat and Kankanhalli [15] take a data-driven approach to assisting shot composition, combining contextual information with composition rules learned from social media to drive visual feedback provided to the user. Tan et al. [23] focus on using AR to help capture consistent product reference images, targeting applications in e-commerce. E

et al. [5] and E et al. [6] focus on providing user feedback to help amateur photographers follow established rules of photographic composition. Slightly closer to our work is that of Kim and Lee [7] on PicMe, which provides a tool to help users specify and capture desired image compositions. None of the above works use 3D tracking for guidance, which can be a sensible design choice for applications with looser requirements on viewpoint precision.

Guidance for 3D Photography: Some past work has focused on capturing specific spatial distributions of images in a scene (e.g., viewpoints distributed along a sphere surrounding an object of interest). The target use for such work is typically image-based rendering or 3D reconstruction. Newcombe et al. [14] and Xiang et al. [25] focus on real-time visual feedback for surface geometry acquisition. Davis et al. [4] and Mildenhall et al. [12] each present AR-based guidance systems for capturing light field data, which they pair with rendering algorithms designed for unstructured input. In particular, the light field capture mode in ReCapture builds on the interface described in Davis et al. [4]. Our supplemental results also use Mildenhall et al. [12] to visualize some of the data captured with this mode.

Computational Re-photography: Our work is most closely related to prior work on *computational re-photography*, which deals with re-capturing previously recorded images of a scene. Shih et al. [19] present a cart-mounted system that uses a motor-controlled platform to re-photograph laser speckle images with millimeter-scale precision. Even more related to our problem is the work of Bae et al. [2], which introduced computational re-photography in the context of recapturing historical photos. Like us, they focus on using interactive visual feedback to guide users toward previously captured images. However, their system, which uses a laptop and tripod, is not hand-held, only addresses single-image capture, and is demonstrated on a very limited range of subjects (primarily rigid architecture).

Collaborative Photography: Our work also shares some similarity with work on collaborative and Internet photography, where photos are collected from publicly-available online sources, as in Snavely et al. [21] and Snavely et al. [20], or through a collaborative game, as in [24]. Inspired by these works, ReCapture uses GPS to offer directions to nearby ongoing time-lapse capture targets to facilitate collaborative capture.

Time-lapse Analysis and Visualization: The analysis and visualization of time-lapse data have also been explored in several works. Sunkavalli et al. [22] and Bennett and McMillan [3] take a computational approach to visualizing traditionally-captured time-lapse data, while Rubinstein et al. [16] explore motion denoising with a particular focus on time-lapse. Closer in spirit to our work is that of Martin-Brualla et al. [9, 10, 11] and Li et al. [8], which composes time-lapse video using crowd sampled images harvested from the Internet.

3 CAPTURE CONDITIONS

To address a wide range of subjects and settings, we need to understand the factors that place different demands and limitations on capture. We can reason about these factors in terms of what information is required at capture time and what information is available. We discuss requirements in terms of how much recaptured views are allowed to deviate from their target, and available information in terms of tracking and re-registration accuracy. It is impossible to anticipate every subject and use case, but we can reason about the mechanisms behind common trends and failure modes to better inform our system's design.

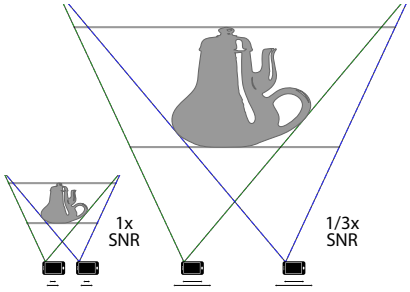

Figure 2: Subject Distance & Pose SNR: We see two scenes that differ only in scale. The pair of images captured on the left will be identical to the pair captured on the right, but the corresponding viewpoints on the right are separated by 3x the distance. This same scale factor applies to the noise of pose estimates, reducing positional SNR to a third of what it is on the left. Note also that simply moving the left subject to the same depth as the right one without scaling would result in an even smaller disparity range, causing an even more negative impact on SNR.

3.1 Tracking & Re-registration

The most effective guidance strategies for re-photography incorporate real-time 3D information about where the camera is and where it needs to go. However, access to this information depends on solving two underlying vision problems: tracking, and re-registration. Tracking provides information about the camera's current location, while re-registration tells us the relative pose of previous images. Either one of these can fail, but tracking will usually succeed whenever re-registration does.[1] This leads to three common conditions: tracking and re-registration can both succeed, tracking can succeed when re-registration fails, or tracking and re-registration can both fail. We encountered each of these conditions frequently in our own use of ReCapture.

3.2 Subject Distance

Visual pose estimation relies on the disparity of image features in a scene, which scales inversely with distance from the camera. This means that as distance to a subject increases, images separated by the same baseline begin to look more similar, which impacts re-photography in multiple ways. On one hand, it reduces precision requirements, since the same amount of positional error will have a smaller effect on recaptured images. On the other hand, it also decreases the signal-to-noise ratio (SNR) of pose estimation (see Figure 2), which makes tracking less stable. To understand the role

[1]This is because tracking involves corresponding images taken under the same conditions, while re-registration involves corresponding images taken at different times.

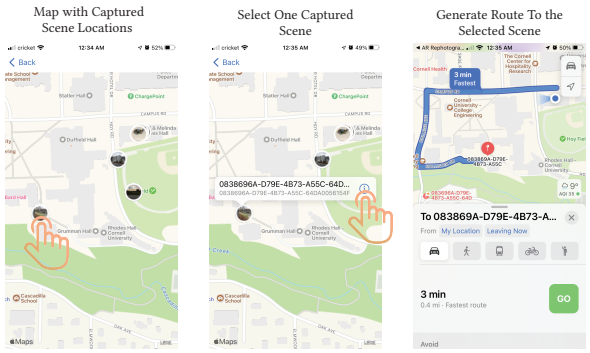

Figure 3: Map View of Nearby Scenes Being Captured: Users can see thumbnails marking the GPS locations of *ReCapture* targets on a map. When they click on a target, they can instantly get directions to the scene of the target. This is designed to facilitate collaborative capture among multiple users.

these factors play in choosing a guidance strategy, consider how each affects the value of the information provided by 3D tracking; relaxing the requirements of capture makes 3D tracking less necessary, and decreasing pose SNR makes 3D tracking less reliable. In practice, this means that for more distant subjects, it can often be advantageous to rely only on 2D guidance.

4 APPLICATION DESIGN

ReCapture groups time-lapse data into *scenes* containing one or more *target views* of a common subject. Users can browse current scenes in a gallery mode or on a map showing GPS pins with the location of each scene. In map view, tapping on a scene will display map directions to that scene from the user's current location to better facilitate collaborative capture (see Figure 3). In our experience, features designed to reduce time to capture resulted in more frequent use, particularly if they made it easier to incorporate capture into daily routines (e.g., a morning commute). With this in mind, upon opening the app, we display nearby scenes in gallery view sorted by their distance from the user's current GPS location. From here, tapping a scene will immediately open it for recapture.

ReCapture offers three different capture modes. Each mode was initially designed to address one of the tracking and re-registration conditions described in Section 3.1. However, as we discuss in Section 8, modes that rely on less 3D information are sometimes preferable even when more 3D information is available. As such, we let the user select which mode to use in different scenarios.

4.1 3D Mode

Our *3D Mode* capture interface uses both tracking and re-registration to provide precise guidance for very accurate recapture. The user begins capture by moving their camera around to initialize tracking and, if recapturing a previous scene, register the current view against a previously recorded world map. In ReCapture, we use Apple ARKit's world tracking API for this process. Once tracking is initialized, we render 3D image planes for the target viewpoint of each ongoing time-lapse in the scene (see Figure 4, bottom left). New target views can be added by tapping the camera icon on the screen. To recapture an existing target view, the user taps on its corresponding image plane, at which point rotation and translation guides will appear on the screen (see Figure 5). Together, these guides take the more complicated 6DOF navigation task of matching a given target pose and factor it into lower-dimensional sub-tasks that are easier to perform.

The rotation guide consists of a red target fixed in the camera's field of view and a gray-blue target rendered pointing in the direction of the target viewpoint's local z-axis. Aligning the red target with the gray one will cause it to turn green, indicating that the current camera orientation matches that of the target pose.

The translation guide, used to convey the translation of the target pose relative to the current camera, takes the form of a circle rendered near the top left of the screen. Translation along the current view's local x-axis is visualized as a horizontal arrow emanating from the center of the guide. The arrow's direction indicates whether the target is to the right or left of the current pose, and its length scales with distance along the current camera's x-axis. Translation along the current y-axis is visualized with a corresponding vertical arrow that works analogously. Translation along the current z-axis is visualized in the size of a second concentric filled circle. If this second circle is smaller than the first, the target is in front of the current camera. If it is larger, then the target is behind. The size and fill color of this z-axis circle scale with the target's distance along the z-axis, shifting in hue from red to green as the camera gets closer to the target.

When the user's camera is within specified distance and orientation thresholds of the target pose, the application automatically records an image. The user can then review the image, compare it with the target, view an alignment of the two based on a best-fit homography, and decide whether to save or retake the image.

When tracking and re-registration are reliable, 3D Mode is extremely effective at facilitating highly accurate re-photography. In fact, our original design for ReCapture contained only one mode, which was an earlier version of this interface. However, the advantages of 3D Mode depends on reliable tracking and re-registration, which are often unavailable in real-world settings. Over time, we added our other two capture modes to address common failure modes we observed through frequent use.

4.2 Overlay Mode

Where 3D Mode relies heavily on the success of tracking algorithms, our second interface foregoes their use entirely. *Overlay Mode* provides feedback in the form of a static semi-transparent image of the target view overlaid on top of the camera's current viewfinder feed. To match the target viewpoint, the user adjusts their camera until features in the live image line up with those in the overlay. Overlay mode works especially well with distant subjects for the reasons discussed in Section 3.2. It is also extremely robust, as it does not rely on any form of tracking or registration, which makes it a reliable fallback for capturing any scene. The main weakness of Overlay Mode is that precisely recapturing subjects close to the camera can be incredibly difficult. In our own use of ReCapture we

Figure 4: Three Recapture Modes in *ReCapture*: On the left is 3D Mode, designed for recapturing close-up scenes in situations where tracking and re-registration both work. In the middle is Overlay Mode, which offers guidance in the form of a simple target image overlay for capturing landscapes or scenes where tracking and re-registration fail. On the right is Light Field Mode, which helps users capture a dense range of views and is effective in situations that call for high precision recapture but re-registration fails, or for capturing dense image data to use in image-based reconstructions of the scene.

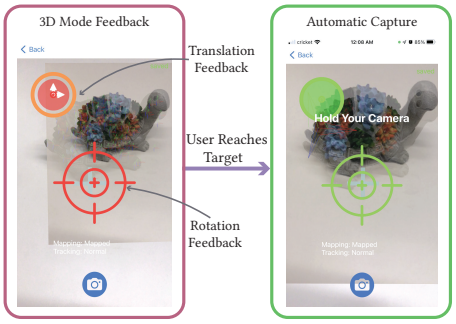

Figure 5: 3D Mode Capture Interface During Recapture: The screenshot on the left shows the interactive feedback provided by the 3D mode. At the upper left corner, the arrows represent translational error in the x and y axes, and the radius of the red circle indicates the translational error in the z axis. A gray-blue target is rendered in the scene, showing the orientation of the reference camera pose. The screenshot on the right demonstrates the status when all six degrees of freedom are matched and automatic capturing is triggered.

found that, as a very approximate rule of thumb, objects larger than a car viewed from several meters away were often easier to capture in overlay mode than 3D mode. We explore this observation in more detail in our user study, described in Section 5.

We also experimented with adding 2D homography-based guidance to overlay mode, similar to interfaces presented in previous work on related problems (e.g., [1, 2, 6, 7]). We believe this type of guidance could be helpful in future systems, but found that 2D tracking frequently failed in the same scenarios where 3D tracking was unsuccessful, and it added latency to overlay mode that outweighed its benefits in our own use of the app.

4.3 Light Field Mode

Our third capture mode is *Light Field Mode*, which is inspired by similar interfaces used in work on light field capture, and can also be used to capture dense data for image-based rendering. We originally added this mode upon finding that many scenes captured in 3D Mode were difficult to re-register under very different lighting conditions (e.g., recapturing a day-lit object at night). However, tracking often still succeeds in these scenarios, which we can use to facilitate more accurate recapture than Overlay Mode offers. We do this by taking a different, more indirect approach to re-photography. Instead of helping the user capture a specific viewpoint, we help them quickly and densely sample a range of views that is likely to cover the intended target. Then, later on, we can use off-line processing to either find the closest captured view to our subject, or directly render the target viewpoint using image-based rendering techniques [4, 12, 13].

Light Field Mode uses a simplified version of the coverage map visualization presented in Davis et al. [4]. After tracking is initialized, the user taps on the subject they wish to capture. We project

the corresponding ray into the scene and intersect it with scene surface estimates provided by ARKit to find a point of focus, which we display as a small sphere in the scene. The user can then hold the camera icon to trigger automatic capture whenever the camera sees the point of focus from a sufficiently new angle. The set of captured images is visualized as a coverage map displayed on a sphere of a larger radius centered at the point of focus. New photos are recorded based on a threshold for the minimum angular difference between the current camera location and a previously captured view, measured relative to the point of focus. Previous work has shown that users can efficiently capture data for image-based rendering this way by "painting" the coverage map using their camera [4]. An image visualizing the distribution of views captured for one of our light field scenes (the bear) can be seen in Figure 15.

Light Field Mode provides a way to perform high-precision re-photography without knowing the precise location of a target view at capture time, which is particularly useful when re-registration fails on nearby subjects. The main drawback of Light Field Mode is that, at least compared to our other modes, the process of densely sampling a range of images is often slow. However, in addition to facilitating accurate re-photography when re-registration fails, Light Field Mode also produces richer spatial data that can be used for more immersive types of rendering. On our project website, you will find examples of time-lapse light fields, which show movement through a scene across both space and time.

5 USER STUDY

Most of ReCapture's design was driven by our own experience using the app during development. However, we also felt it was important to validate some of our own observations with external users. Designing a user study to focus directly on time-lapse capture is difficult due to the long-term nature of the task. However, it is much easier to conduct a controlled study on the sub-problem of re-photography, which was not explored with hand-held solutions in previous work. With this in mind, we conducted a user study to examine capture in a number of different scenarios. We recruited 20 participants (9 males and 11 females, ages 20-50) via personal contacts and message boards, and conducted the experiments over the course of a week. Some participants worked in fields that require strong spatial skills, but none had extensive experience with AR. We designed the study to examine a key aspect of ReCapture's multi-interface design: the use of different guidance strategies for different capture scenarios. In particular, we focused on exploring the relative benefits of 3D Mode and Overlay Mode in different settings. We did not compare with Light Field Mode in these experiments because it addresses recapture in a fundamentally different way. We also surveyed participants after the study to get qualitative feedback about their experience.

5.1 Study Design

Users were asked to complete two recapture tasks with an iPad mini provided by us. Each task consisted of re-photographing a different scene from several target viewpoints selected ahead of time by us. We trained each user on both interfaces prior to their first task by having them capture a separate but similar scene. For each user, we randomly assigned half of the views in each scene to 3D Mode and the other half to Overlay Mode. We also randomized the order in which each mode was used to help counterbalance any proficiency gained over time. We set a time limit of one-minute per image for capturing the target images assigned to a given capture mode, measured cumulatively for all of the images assigned to that mode, to ensure that the process was completed. The two capture modes were tested back-to-back in randomized order. Our experiments included three different scenes, shown in Figure 6, which ranged in scale and subject distance from a close-up tabletop scene to a more distant outdoor landscape.

Users were instructed to match target viewpoints as closely as possible, and allowed to preview the target images for reference immediately before the experiment began. Users could also review images immediately upon capture by flipping between the captured image and an aligned blend with the target view. Users could then choose to retake an image, but were informed that this would not reset their task time. Any target not recaptured by the time limit was considered incomplete.

5.2 Evaluation

Each user filled out a questionnaire after completing the study where they were asked to evaluate various aspects of each capture mode for each task using a 5-point Likert scale. They were also asked to rate any preference for one mode or the other on different scenes and optionally respond to prompts for open-ended explanations or additional comments. The full questionnaire and responses can be found in our supplemental material.

In addition to the questionnaire, we used high-resolution images to scan and reconstruct a model of the close-up tabletop scene and calibrated this model against direct distance measurements to build a metrically accurate digital double. We then registered the target views and images recaptured by each user against this model using COLMAP [17, 18] to obtain accurate pose estimates for each image. This lets us evaluate the metric accuracy of each user's captured pose compared to the target. Reconstructions of the other two scenes were not suitable for the same high-precision analysis, as both were larger, semi-public spaces where we could not guarantee controlled ground truth.

6 USER STUDY RESULTS

6.1 Quantitative Analysis

We evaluated survey responses and accuracy measures according to a Student's t-distribution model.

6.1.1 Survey Results. The surveys results largely confirm the observations and analysis from our own use of the app. Users generally preferred 3D mode for the close-up scene and overlay mode for the scenes with content at greater distances (see Figure 7). Interestingly, 3 users (15%) did report a preference for overlay mode when capturing the close-up scene, but all 3 of these users were substantially more accurate using 3D mode; in fact, each of these users was more accurate on every image they took in 3D mode than they were for any image they took using overlay mode, both in terms of camera position and orientation. This may suggest that some users actively prefer interfaces that let them be less accurate, perhaps finding more opinionated feedback frustrating.

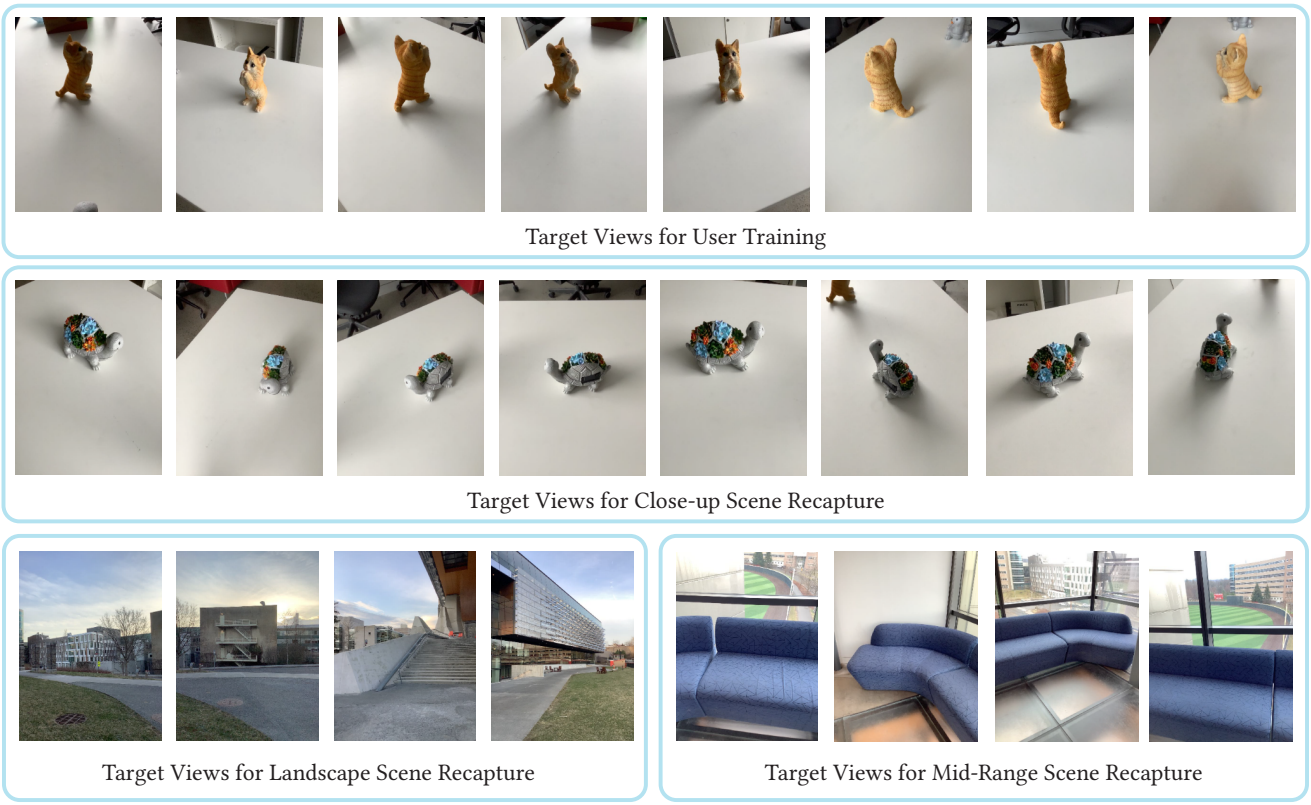

Figure 6: User Study Target Scenes: Close-up Scene (top) two garden figurines placed onto a white meeting table near the window. We created two tasks, each with 8 recapture targets by rotating the camera around one figurine. We kept the relative position between the figurines and the table stable, while we did not control the lighting condition in the room. Mid-range Scene (bottom right) an indoor public area with floor-to-ceiling windows. We created one task with 4 recapture targets by including part of outdoor scene and part of indoor scene in the camera view. Landscape Scene (bottom left) an outdoor location surrounded by buildings and urban vegetation. We created two tasks, each with 2 recapture targets viewing at objects 10-50 meters away from the camera. The 3D Tracking relocalization highly depends on the features on a stone bench that is within 5 meters away from the camera pose.

6.1.2 Re-capture Pose Accuracy. Our analysis of pose accuracy for the close-up scenes is quite conclusive, with every single user performing more accurately overall in 3D Mode ($p < 0.0005$). This trend also holds for every viewpoint individually if we average accuracy across all users. Figure 8 shows results for camera translation, which is generally the most critical type of error for re-photography as it introduces spatially-varying shifts in image content that are difficult to correct for using a homography.

6.1.3 Completion Rate. We also evaluated completion rate for each task, representing the fraction of views captured within the given time limit. Among all participants, there were 160 recapture targets for the close-up scene, 36 for the mid-range scene, and 44 for the outdoor scene, with targets in each scene distributed evenly across the two capture modes. As shown in Figure 9, only a single target was not completed on time for the close-up scene (in overlay mode). 33% of the targets were not completed for the mid range scene in 3D mode, compared with just 11% in overlay mode. In the outdoor landscape scene this discrepancy increased, with over 80% of targets left uncompleted in 3D mode compared with just 9% in overlay mode.

In addition to these numbers we observed that, as predicted, tracking noise for 3D Mode was generally higher in our more distant scenes. It is also notable that, within low completion rate for 3D Mode on our landscape scene, all users were able to initialize tracking, but the success of re-registration varied with lighting and weather conditions. This matches the observations from our own that motivated adding Light Field Mode to ReCapture.

6.2 Qualitative Analysis

We also draw insights from the written responses users provided to survey questions.

6.2.1 Learning Curves. Some participants noted that 3D Mode had a slightly higher learning curve than Overlay Mode. In the words of one user: "*The image overlay interface was easy to use, while it's*

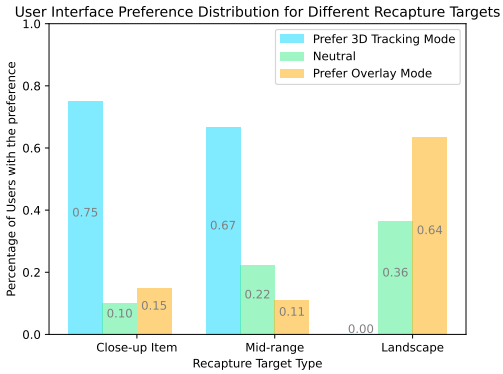

Figure 7: User Interface Preference on Different Recapture Targets: Users reported preferences for different interfaces on different subjects using a 5-point Likert scale. Overall, users preferred 3D Mode for our smaller, more nearby scenes, and Overlay Mode for our most distant scene. Note that the users who preferred Overlay Mode for capturing the close-up scene were more accurate with 3D Mode, despite their reported preference. These results support our multi-interface design, as well as our observations about when each of the tested interfaces works best.

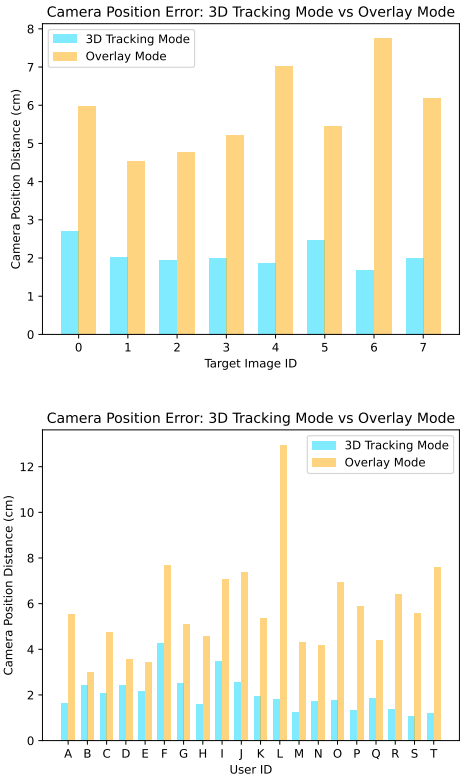

Figure 8: Average Re-photography Position Error in the Close-Up Scene Measured for Each Target Image (top) and User (bottom): Across all users and target images in the close-up scene, we see higher accuracy using 3D Mode.

hard to get a super precise image exactly like the provided one. The 3d tracking interface has a learning curve, but once I managed to use it, I was able to get some precisely captured images." There was one exception to this, participant *L*, who struggled with Overlay Mode more than other users but performed about average using 3D Mode. That participant's feedback offers the possible explanation that they are "*not very good with spatial tasks*" and found "*it was nice that the 3D tracking interface gave clear direction.*".

6.2.2 Overlay Mode Observations. The hardest part of using Overlay Mode is often distinguishing between translation error and rotation error. The challenge is that, while a single scene point can always be brought into alignment through pure translation or rotation, only the correct combination of these will correctly align points at different depths. As one participant put it: "The overlay interface was simple to understand but it was sometimes tricky to figure out which way to move the camera to improve the alignment." Another joked that Overlay Mode "*could be a cure for OCD because something is always off.*"

Some participants attributed the challenges of Overlay Mode to a perceived flaw in how the system displayed images for review. As one user speculated, "*alignment might have taken too much consideration of the backgrounds[sic] and did not prioritize the center object*", referring to situations where a central object initially appeared well-aligned, but the best-fit homography displayed during review "messed it up" by transforming background features into alignment, which caused the previously aligned foreground to ghost. Frustrating as this may be for some users, it highlights errors that can have substantial impact on time-lapse. Conveying this more clearly to the user before they take an image could improve the Overlay Mode experience.

6.2.3 3D Mode Observations. Tracking noise and difficulty with re-registration were the most significant issues with 3D Mode in our larger more distant scenes, so most feedback on the 3D Mode interface was focused on the close-up scene. Many users struggled at first with how to move the camera one degree of freedom at a time (e.g., translating while keeping orientation fixed), but quickly improved at this with practice and saw it as an asset upon final review. As one user put it: "*following the interface that gave directions on the screen (even though at first was less intuitive) seemed to give a better outcome so ultimately felt like the more rewarding interface.*"

One user noted that graphical feedback did not feel intuitive to them, speculating that "*processing audio/text information is faster than reading the different kinds of cues on the screen*" and suggesting the use of natural language instructions like "*move closer to the object*," or "*move your phone to the right on the current surface.*" This could be an interesting direction to explore for less visual users and could lead to more accessible designs for the visually impaired.

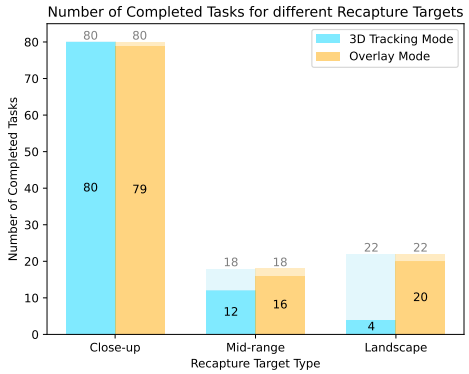

Figure 9: Number of Completed Recaptures (dark) v.s. Recapture Targets (light) for Different Tasks: The participants could almost always complete the close-up task in time, while the mid-range and landscape tasks were less controllable especially with the 3D mode. Most participants failed to relocalize their camera in the outdoor environment.

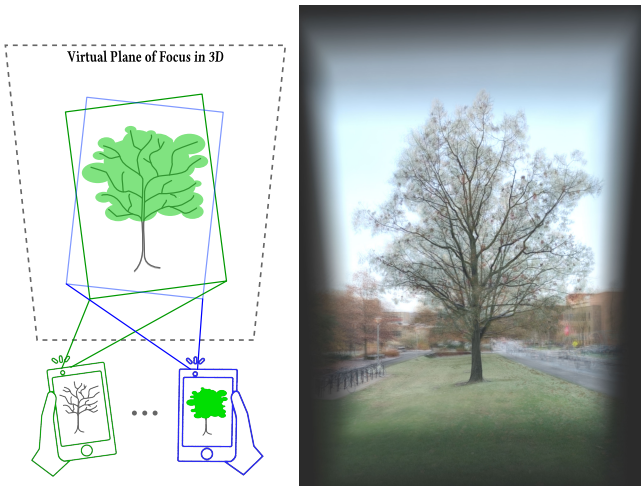

Figure 10: Focus Plane-Based Alignment: On the right we see a blend of 16 images featuring the same tree taken at different times of year. The tree itself looks quite different across these images, so most automatic alignment methods would focus on background features. To fix this, we estimate the relative poses of each image, and project all 16 onto a common 3D plane of focus placed at the virtual depth of our tree. We then blend projected images to get the result on the right.

7 TIME-LAPSE VISUALIZATION

The data produced by ReCapture is not typical time-lapse video. It is irregularly sampled in time, and while significantly reduced by our application's guidance, data still tends to contain small inconsistencies in captured viewpoints. In most cases, these issues are minor enough for us to address by aligning images with a simple homography and linearly blending between adjacent images in time will produce compelling video. We offer this visualization as a default for viewing captured data in-app with ReCapture. Users can sort this visualization by the date of capture, or by the time of day when images were captured. Sorting by date works well for visualizing things like plant growth of changing seasons, while sorting by time of day is a good way to visualize the movement of shadows with the sun. In addition to this default visualization, we explore various other visualizations and ways to deal with small inaccuracies in capture as well. More of the results described here can be found on our project webpage.

7.1 Interactive Space-time Viewer

One exciting aspect of our approach to time-lapse capture is the ability to more easily capture multi-view time-lapse data, which currently quite rare. To facilitate exploring this data in both space and time, we implemented a simple interactive 3D space-time viewing application in WebGL. For data captured entirely in 3D Mode or in a single session of Light Field Mode we obtain pose estimates for each image during capture. We can also use offline structure from motion (in our experiments performed with COLMAP [17, 18]) to combine data from different modes or refine capture-time pose estimates. Once we have pose estimates, we load these into a virtual scene with a user-controllable camera. To render the scene, we first calculate the nearest 16 views in space, time, or some user-specified combination of the two. We then project a weighted combination of these views onto a plane of focus at some user-controllable depth in front of the scene's virtual camera. This effectively performs a weighted k-nearest neighbor reconstruction over space and time given the unstructured sampling of a scene.

Our interactive space-time viewer is particularly useful when a scene's geometry changes frequently over time, even when rendering time-lapse video from a single viewpoint. To understand why, note that projecting multiple images onto a common plane of focus has the same effect as warping those images by homographies based on content near the plane of focus in the scene. In scenes where the subject changes geometry frequently, a best-fit homography calculated in 2D will generally align on background features. Projecting onto a user-controllable plane of focus in 3D lets us instead align on the intended subject, even if we cannot reconstruct its changing geometry (see Figure 10).

7.2 Image Composites

As video is a central focus of our work, our results are best appreciated in video form. However, we can also create static visualizations of our data by compositing different frames of a collected time lapse into single images.

7.2.1 Time-Lapse Images. In one type of composite, we map different regions of an output image to different frames from the captured time-lapse (see Figure 11). We can assign these mappings manually, but for large data sets we quickly explore alternatives by sorting input data according to different criteria (e.g., time of year, time of day, or the average correlated color temperature of pixels) and assigning output regions to an ordered sub-sampling of the sorted data.

Figure 11: *ReCapture* Composite Images Gallery: We can visualize time-lapse data in a static image by mapping different regions of an output image to different input frames from a captured time-lapse. Images labeled with season in the first row are generated by images captured in the fall, sorted by calendar time and filtered by average pixel value. The second row is generated by a smoother linear blending operation, where both season and environment light vary across the images. The third row shows various examples of lighting changes in the same scenes.

7.2.2 Mean Images. Another interesting way to visualize captured data is to simply align and average all images. For subjects captured under a wide range of lighting conditions, this produces images with very uniform lighting. It also provides a useful static way to visualize how much different parts of the scene move over time (see Figure 12), or how consistent re-photography was over the course of capture (see Figure 13). Mean images are also high-precision, which facilitates detail enhancement and tone-mapping algorithms associated with HDR imaging (e.g., Figure 1 right middle).

7.3 Time-Lapse Light Fields & IBR

In addition to facilitating precise re-photography in situations where re-registration is difficult, Light Field Mode offers a convenient way to capture data for image-based rendering (IBR) of static scenes under varying lighting conditions. On our website you will also find examples of videos interpolating between image-based reconstructions of scenes captured in Light Field Mode over both space and time.

8 DISCUSSION

8.1 Observations from Internal Use

Figure 14 offers a conservative summary of data captured by 3 members of our team over the span of one year at the time of writing. Excluding data captured in light field mode and data captured purely for testing or debugging purposes, we captured over 100 targets for a total of roughly 4000 images. Here we describe some of our own observations using the application.

Figure 12: Mean Images Comparison: By averaging aligned images from a time-lapse we can visualize what parts of the scene change over time, and what parts remain static. In the top example we see that plants and cars either are either removed or blurred, as they move and deform over time. In the middle row we can see roughly how much the recycling bins move around each time they are returned after being emptied. On the bottom row, we see that a pole is blurred compared to static content like buildings. The shape of the blur shows a slight bend in the pole, and suggests that it occasionally rotates over time.

8.1.1 Routine Use. One of our most salient observations was that ReCapture works best when capture is incorporated into a regular part of daily routine. In general, we tried to recapture a target whenever we happened to be nearby and had the time to do so. As a result, targets that we encountered regularly (e.g., on the walk to work, or a regular lunch spot) yielded the most images. This accounts for the heavy tail in Figure 14.

8.1.2 Mode Selection. We eventually converged on a fairly consistent decision tree for deciding what mode to use in different scenarios. We summarize this tree, which closely adheres to the motivations for including each capture mode, in Figure 1. However, there were situations where mode selection tended to deviate from this tree. Overlay mode launches very quickly, making it very easy to incorporate into regular routines, so we generally preferred it for very frequent captures, even if tracking was possible. Also, re-registration (at least, in ARKit at the time of writing) is fairly sensitive to changes in the appearance of subjects, so failures were common at different times of day, and in many cases a subject would change over time (e.g., leaves growing or shedding on a tree) in a way that would cause re-registration to fail even under similar lighting conditions. In some of these cases, we switched capture mode mid-time-lapse. We added Light Field Mode specifically to address casses where this was necessary and Overlay Mode proved difficult due to a small subject distance. It is notable that we almost never switched a target from a mode with fewer tracking and registration requirements to one with more. In aggregate, these factors contributed to us using Overlay Mode more and with more regularity over time.

8.1.3 Other Observations. Certain behaviors make ReCapture much easier to use in practice. In 3D Mode, choosing a consistent textured patch of the scene to use when re-initializing the scene map makes capture substantially easier. Similarly, in Overlay Mode, remembering where the camera is positioned relative to fixed points of reference in the scene makes recapture much easier over time. We also found that resting the device against a surface when setting a

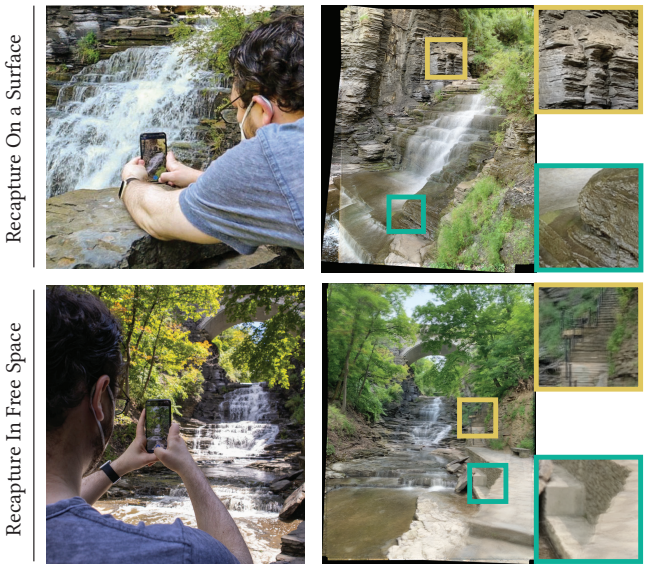

Figure 13: Target View Selection in Overlay Mode: The left column shows a user recapturing two different target views in a similar outdoor setting. For the top target, the user rested their phone on a surface for added stability (top left). For the bottom target, they held the phone in the air. By averaging the views captured of each target, we can visualize viewpoint inconsistencies as blur in a mean image. In the bottom viewpoint, we see slight blurring of fixed objects at some depths. For the top viewpoint, we see sharp detail across different depths, which reflects the added stability during capture.

target view tends to lead to more precise recapture. Figure 13 illustrates this by comparing the alignment of images for two nearby outdoor targets, one captured by resting the device on a surface and the other captured by holding the device in free space.

Our first implementation of Overlay Mode used the most recent recapture of a target view as reference. We initially implemented it this way anticipating that it would better accommodate recapture of scenes that changed regularly. However, we observed that this resulted in some drift of the captured viewing angles over time. After switching the overlay to show the same, original target view for every recapture, there was a noticeable improvement in the consistency of captured views.

8.2 Limitations on Current Design

Some of the challenges we describe in this paper are fundamental and likely to persist in future systems designed for hand-held time-lapse capture. For example, the fact that nearby scenes are difficult to capture accurately is a direct consequence of epipolar geometry. However, some of the limitations we encountered can be at least partially addressed by improvements to real-time tracking and re-registration on mobile devices. We expect that as this technology improves, the tipping point for when a scene is easier to recapture with 3D guidance will shift, making an interface like our 3D Mode more appealing in more scenarios.

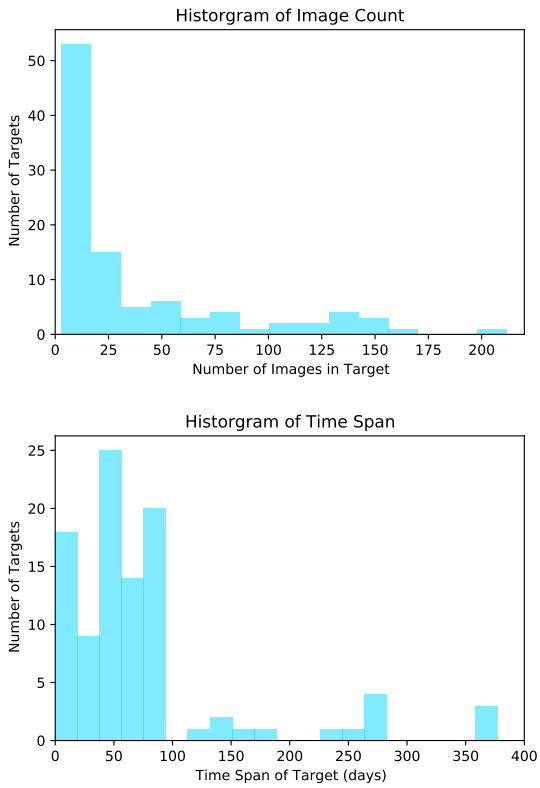

Figure 14: *ReCapture* Internal Usage Statistics: Our team has actively used ReCapture for over a year at the time of writing. These histograms give conservative estimates for the distributions of recaptures per target (top) and the span of time over which each target was captured (bottom).

It is also worth noting that the versatility of our current application design could, in theory, be achieved with a single capture mode that adapts guidance in real-time as tracking and re-registration succeed or fail. However, such a design poses significant challenges. Initializing tracking currently requires moving the camera in a particular way, which is different behavior from what users normally do in Overlay Mode. It may be possible to address this by quickly diagnosing a scene at the beginning of capture in a kind of metering process, but we leave exploring this kind of metering to future work.

9 CONCLUSION

Our work shows a great deal of potential in using interactive visual feedback to guide hand-held time-lapse capture on mobile devices. We have presented an extensive exploration of this task and outlined many of the important design considerations for future work in this area. We believe that systems like ReCapture could have substantial impact across a wide range of applications ranging from scientific field work, to historical and environmental preservation, community development, and the arts. We have also demonstrated that making this application practical requires careful consideration

of factors related to human interaction design, computer vision, visualization, and graphics. We believe this opens many opportunities for important future work on each of these aspects of the problem and are excited to see what other researchers and users will do with ReCapture.

REFERENCES

[1] Andrew Adams, Natasha Gelfand, and Kari Pulli. 2008. Viewfinder alignment. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 597–606. https://doi.org/10.1111/j.1467-8659.2008.01157.x

[2] Soonmin Bae, Aseem Agarwala, and Frédo Durand. 2010. Computational Rephotography. *ACM Trans. Graph.* 29, 3, Article 24 (jul 2010), 15 pages. https://doi.org/10.1145/1805964.1805968

[3] Eric P. Bennett and Leonard McMillan. 2007. Computational Time-Lapse Video. *ACM Trans. Graph.* 26, 3 (jul 2007), 102–es. https://doi.org/10.1145/1276377.1276505

[4] Abe Davis, Marc Levoy, and Fredo Durand. 2012. Unstructured Light Fields. *Comput. Graph. Forum* 31, 2pt1 (may 2012), 305–314. https://doi.org/10.1111/j.1467-8659.2012.03009.x

[5] Jane L. E, Ohad Fried, Jingwan Lu, Jianming Zhang, Radomír Měch, Jose Echevarria, Pat Hanrahan, and James A. Landay. 2020. Adaptive Photographic Composition Guidance. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. ACM, 1–13. https://doi.org/10.1145/3313831.3376635

[6] Jane L. E, Kevin Y. Zhai, Jose Echevarria, Ohad Fried, Pat Hanrahan, and James A. Landay. 2021. Dynamic Guidance for Decluttering Photographic Compositions. In *Proceedings of the 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21)*. ACM. https://doi.org/10.1145/3472749.3474755

[7] Minju Kim and Jungjin Lee. 2019. PicMe: Interactive Visual Guidance for Taking Requested Photo Composition. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300625

[8] Zhengqi Li, Wenqi Xian, Abe Davis, and Noah Snavely. 2020. Crowdsampling the plenoptic function. In *European Conference on Computer Vision*. Springer, 178–196.

[9] Ricardo Martin-Brualla, David Gallup, and Steven M. Seitz. 2015. 3D Time-Lapse Reconstruction from Internet Photos. In *2015 IEEE International Conference on Computer Vision (ICCV)*. 1332–1340. https://doi.org/10.1109/ICCV.2015.157

[10] Ricardo Martin-Brualla, David Gallup, and Steven M. Seitz. 2015. Time-Lapse Mining from Internet Photos. *ACM Trans. Graph.* 34, 4, Article 62 (jul 2015), 8 pages. https://doi.org/10.1145/2766903

[11] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. 2021. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*. https://doi.org/10.48550/arXiv.2008.02268

[12] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. 2019. Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines. *ACM Trans. Graph.* 38, 4, Article 29 (jul 2019), 14 pages. https://doi.org/10.1145/3306346.3322980

[13] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.

[14] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. 2011. KinectFusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. 127–136. https://doi.org/10.1109/ISMAR.2011.6092378

[15] Yogesh Singh Rawat and Mohan S. Kankanhalli. 2015. Context-Aware Photography Learning for Smart Mobile Devices. *ACM Trans. Multimedia Comput. Commun. Appl.* 12, 1s, Article 19 (oct 2015), 24 pages. https://doi.org/10.1145/2808199

[16] Michael Rubinstein, Ce Liu, Peter Sand, Frédo Durand, and William T. Freeman. 2011. Motion denoising with application to time-lapse photography. *CVPR 2011* (2011), 313–320. https://doi.org/10.1109/CVPR.2011.5995374

[17] Johannes Lutz Schönberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[18] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. 2016. Pixelwise View Selection for Unstructured Multi-View Stereo. In *European Conference on Computer Vision (ECCV)*. https://doi.org/10.1007/978-3-319-46487-9_31

[19] Yi Shih, Abe Davis, Samuel Hasinoff, Fredo Durand, and W.T. Freeman. 2012. Laser Speckle Photography for Surface Tampering Detection. *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 33–40. https://doi.org/10.1109/CVPR.2012.6247655

[20] Noah Snavely, Rahul Garg, Steven M. Seitz, and Richard Szeliski. 2008. Finding Paths through the World's Photos. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008)* 27, 3 (2008), 11–21. https://doi.org/10.1145/1360612.1360614

[21] Noah Snavely, Steven M. Seitz, and Richard Szeliski. 2006. Photo Tourism: Exploring Photo Collections in 3D. *ACM Trans. Graph.* 25, 3 (jul 2006), 835–846. https://doi.org/10.1145/1141911.1141964

[22] Kalyan Sunkavalli, Wojciech Matusik, Hanspeter Pfister, and Szymon Rusinkiewicz. 2007. Factored Time-Lapse Video. *ACM Trans. Graph.* 26, 3 (jul 2007), 101–es. https://doi.org/10.1145/1276377.1276504

[23] Nicole Tan, Rachana Sreedhar, Christian Vazquez, Jessica Stigile, Maria Gomez, Ashwin Subramani, and Shrenik Sadalgi. 2021. *An Augmented Reality Guided Capture Platform for Structured and Consistent Product Reference Imagery.* Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3447527.3474848

[24] Kathleen Tuite, Noah Snavely, Dun-yu Hsiao, Nadine Tabing, and Zoran Popovic. 2011. PhotoCity: Training Experts at Large-Scale Image Acquisition through a Competitive Game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 1383–1392. https://doi.org/10.1145/1978942.1979146

[25] Xiaojun Xiang, Hanqing Jiang, Guofeng Zhang, Yihao Yu, Chenchen Li, Xingbin Yang, Danpeng Chen, and Hujun Bao. 2021. Mobile3DScanner: An Online 3D Scanner for High-quality Object Reconstruction with a Mobile Device. *IEEE Transactions on Visualization and Computer Graphics* 27, 11 (2021), 4245–4255. https://doi.org/10.1109/TVCG.2021.3106491

Figure 15: Sample Visualizations of Data Acquired With Each of ReCapture's Three Capture Modes: (left) Wilting roses captured in 3D Mode, which supports accurate multi-view recapture for close-up objects. (middle) A time-lapse image composite of data taken in Overlay Mode, which facilitates robust recapture of more distant scenes (e.g., landscapes). (right) Images showing the estimated moses of a subject captured with Light Field Mode, which helps users capture data for synthesizing new views and 3D reconstruction. See additional results on our project website.