Evaluating the Impact of Recovery Density on Augmented Reality Tracking

Christopher Coffin* Cha Lee† Tobias Höllerer‡

Four Eyes Lab
Department of Computer Science
University of California Santa Barbara

ABSTRACT

Natural feature tracking systems for augmented reality are highly accurate, but can suffer from lost tracking. When registration is lost, the system must be able to re-localize and recover tracking. Likewise, when a camera is new to a scene, it must be able to perform the related task of localization. Localization and re-localization can only be performed at certain points or when viewing particular objects or parts of the scene with a sufficient number and quality of recognizable features to allow for tracking recovery. We explore how the density of such recovery locations/poses influences the time it takes users to resume tracking. We focus our evaluation on two generalized techniques for localization: keyframe-based and model-based. For the keyframe-based approach we assume a constant collection rate for keyframes. We find that at practical collection rates, the task of localization to a previously acquired keyframe that is shown to the user does not become more time-consuming as the interval between keyframes increases. For a localization approach using model data, we consider a grid of points around the model at which localization is guaranteed to succeed. We find that the user interface is crucial to successful localization. Localization can occur quickly if users do not need to orient themselves to marked localization points. When users are forced to mentally register themselves with a map of the scene, localization quickly becomes impractical as the distance to the next localization point increases. We contend that our results will help future designers of localization techniques to better plan for the effects of their proposed solutions.

Index Terms: B.8.0 [Computing Methodologies]: Image Processing and Computer Vision—Applications; I.6.3 [Computing Methodologies]: Simulation and Modeling—Applications

1 INTRODUCTION

Augmented reality (AR) applications are increasingly common on mobile devices. A few examples include Layar [11], Wikitude [2], and Junaio [1]. Most current systems rely on GPS and some combination of inertial and magnetic sensors in order to track the pose of the camera. However, such sensors do not allow for precise overlays of annotations and can suffer from latency. Vision-based tracking systems are commonly deemed a necessary ingredient to achieving satisfactory levels of augmented reality tracking accuracy. In spite of great advancements in this field, the robustness of markerless wide-area tracking is still a concern, and robustness problems can result in occasional or frequent loss of tracking, dependent on the adversity of the environment and the availability of, and integration with, complementary tracking methodologies.

In the case of tracking loss, registration needs to be recovered so that the mobile device is again registered with the scene. Vision-based tracking recovery can be achieved by comparing the current camera frame to previously gathered data. We distinguish two different approaches, based on what data we are comparing against: Keyframe-based approaches compare against data that was collected recently by that same mobile device (e.g. previously seen camera "keyframes" or natural features from those). Model-based approaches compare against data which has been collected at an earlier time, and stored off-line (e.g. models or panoramas). Depending on the quality and density of the data describing the scene, as well as the method used for the recovery, the spatial distribution of view poses that afford successful tracking recovery may vary significantly.

We are interested in the impact of the density or sparsity of such re-localization "pockets" on tracking performance, specifically on the time to resume tracking. We believe that our evaluation results will aid designers of mobile vision-based tracking systems in their decisions on localization techniques and keyframe or model considerations. Our evaluation is performed using a simulated recovery and tracking system for both the keyframe and simulated model-based approaches. Our goal is not to evaluate a specific tracking or recovery solution, but to provide more general results. By running the evaluation with a simulated tracking system, we provide more general results and obtain experimental control.

Our main contribution lies in analyzing the relationship between the density of localization positions and the time it takes users to recover tracking. We provide such analysis for two classes of tracking recovery methods (keyframe-based and model-based), both of which may require some user interaction for the conscious matching of current keyframes to stored data.

2 RELATED WORK

Relevant related work to the evaluations we present in this paper can be categorized into three groups:

The first two groups are comprised of systems which exemplify the types of recovery we evaluated in this study, namely keyframe-based and model-based recovery methods. The third group discusses papers which are concerned with similar spatial cognition questions to the ones we consider in our work.

2.1 Keyframe-based Recovery

There are a number of papers on vision-based tracking systems which use some form of localization in order to determine the position of the camera after normal frame to frame tracking has been lost. Keyframes are used, for example by Reitmayr et al. [14] in their model-based tracking system, by Klein and Murray in PTAM [10], and by Kim et al. in their modified version of Envisor [9]. Keyframe-based recovery involves storing individual frames or data from the camera's path at regular intervals. For example, such a system may collect and store a new camera image at every second,

*e-mail: ccoffin@cs.ucsb.edu
†e-mail: chalee21@cs.ucsb.edu
‡e-mail: holl@cs.ucsb.edu

or at every other frame, or at frames which are deemed to be especially robust, as in [10]. Keyframe-based recovery does not require a model of the scene. The data which is collected in the current session is assumed to be sufficient to resume tracking.

2.2 Model-based Recovery

There are a number of solutions which involve using a model of the scene, or large collections of spatially located information in order to localize the camera. The exact implementation of the localization process depends on the type of available information. For example, Schindler et al. perform localization using a database of images [15], Wu et al. perform localization using image patches derived from models of the scene [18], Karlekar et al. [8] use virtual building models in order to match silhouettes to aid in localization, and Yang et al. [5] use high quality terrestrial LIDAR data in order to perform recovery.

Of course, the quality of the localization will vary significantly depending on the type and quality of the existing data and the localization method used. We do not aim to directly evaluate any existing system. Our goal is to determine how fast users can resume tracking after loss of registration, given the density of usable localization points. If a system has a known average distance between localization points it should be possible to estimate the time it will take users to localize their view.

We seek to approximate such localization solutions in as general a form as possible. Therefore, we implement a generalization of recovery from stored model data, which we use to model various densities of usable localization positions. Variations in density of localization positions naturally occur due to differences in the quality of available model data and the localization technique used. If we can estimate the necessary density of camera poses for which recovery is possible based on a certain maximum time we want to spend on resuming the AR experience after loss of tracking, we can then make informed decisions on, for example, how frequently we have to store keyframes (for keyframe-based approaches), or (in the case of model-based approaches) how densely we have to survey an environment with panoramic imagery; or how accurately we have to model buildings and other reference objects geometrically; or, what feature types (e.g. SIFT [12] or VIP [19]) to use to index into image databases to provide the necessary off-axis recognition.

Vice versa, our results should prove useful for estimating the usability (time to tracking recovery) of a given system for a given set of data based on the expected density of localizable vantage points surrounding the user.

2.3 Spatial Cognition

Our work is also related to literature on spatial cognition, in particular map understanding and image-based homing. Both keyframe and model-based recovery methods are heavily dependent on people's ability to form spatial mental models [20] when it becomes necessary for users to consciously adjust camera poses to include specific imagery (modeled or seen previously). Our general keyframe alignment approach is similar to some research on image-based homing, such as [4]. However we are not aware of any work which focuses on the effects of varying the density at which representations are collected. Recovery in our abstracted model-based evaluation is dependent on the user's ability to quickly and accurately determine a localization point relative to a virtual representation of the scene. There are a number of studies which have evaluated various interfaces for perceptual navigation, particularly [17] and [13]. However, these studies evaluate different tasks and somewhat different interfaces than those we chose as the reference methods for our evaluations.

3 Experiment 1: Keyframe-based Recovery

For Experiment 1, we wanted to investigate how the frequency of collecting keyframes impacts the recovery time for a visual search task in augmented reality. Our goal was not to evaluate an existing feature based tracking system or an existing recovery solution. We consider recovery time to be inherent to any feature based tracking system and recovery solution. In order to run carefully controlled evaluation studies, we simulate tracking loss and tracking recovery. We seek general results, and since we were aiming for the loss of tracking to be statistically varied under our control, we use a robust sensor based tracking system (InterSense IS900) instead of a vision-based tracker.

Keyframe-based recovery involves storing individual frames or data from a camera's path at regular intervals. For example, such a system may collect a new sample at every second, or at every other frame (frequency). In our experiment we wanted to investigate how the frequency at which these samples are collected affects recovery time. Obviously, the more frequent the collection occurs the less the recovery time should be. We hypothesized that increasing the frequency of keyframe generation would not increase the recovery time linearly. Once tracking is lost, the user has to move to the position where the last keyframe was taken. For very high keyframe collection frequencies the users could simply backtrack slightly and resume tracking. In those lucky cases the tracking would resume automatically as users would already be overlapping a keyframe. However, localization for lower frequencies may not be a simple task. The user must first understand where to move to, which may not be easy if considerable time elapsed between the last logged keyframe and tracking loss. After understanding where to move to, the user must then position the camera in the correct pose to regain tracking. Our hypotheses for how recovery time would be affected by the keyframe frequency were:

H1.1: From a very high frequency of collecting keyframes (close to every frame) to a certain (still high) frequency of collecting keyframes, there would be no significant difference in the recovery time.

H1.2: From this high frequency of collecting keyframes to a low frequency, there would be a significant effect from decreasing the frequency time.

H1.3: From this low frequency to even lower sampling rates, the system would no longer be usable due to frustration by human users.

3.1 Task and Environment

While our true interest was in users' abilities to recover from tracking loss, we defined a simple AR search task for our study participants, and asked them to complete this task as speedily as they could. We explained to them that they would experience tracking loss occasionally, and also how they could cope with such a contingency.

Participants were asked to locate several virtual tags, one at a time, in a tabletop-sized model city, experiencing AR via a magic-lens-style handheld display (cf. Figures 1 and 3). The model city was at chest height as seen in Figure 2. Participants were asked to walk around the model, observing it constantly through the handheld display. This was necessary to locate the next virtual tag, since buildings could and would occlude the tags, and tags would therefore only be visible from certain vantage points. We designed this task to be a non-trivial example of an AR search task representative of common AR applications. It required the participant to continuously make the cognitive connection between the real world and the virtual components, a key factor for true AR tasks. The model city was set up through trial and error to be sufficiently complicated for

Figure 1: The display device used for both experiments. The device uses a MIMO iMO 10 inch USB display, a Point Grey FireFly camera, and an IS900 Headtrax device.

requiring the participant to stay aware of where they had previously searched. The handheld screen would always show the augmented camera feed of what the user pointed the 'magic lens' at, as well as four buttons labeled with 4-letter sequences, one of which corresponded with the virtual tag label currently pasted to a building wall somewhere within the city model. Once the label was spotted and the corresponding button pressed, a new tag was placed into the AR scene, and four new labels were displayed on the virtual handheld screen buttons. The model information was used in order to provide a task for the user to motivate movement. The model data was not used for the recovery. For the purposes of our evaluation, each of the virtual tags can be replaced with a physical object. We used virtual tags to allow only a single tag to be displayed at a random position for each iteration.

The models of the buildings themselves were obtained through the City of Sights project [6], which provides a number of virtual models as well as instructions for constructing physical representations of the models. To match the layout of the virtual models to the physical scene, we collected a screenshot of a top down view of a virtual layout of the scene using an orthographic projection and then printed the corresponding image onto a poster. The physical models were then aligned with the virtual models by placing the physical models over their corresponding outline on the poster. The small cityscape was then placed on a table with a bounding area of 745mm x 1160mm. In order to control for differences in our participants' body heights, we adjusted the height of the table relative to the height of the user. This was accomplished by inserting layers of Styrofoam padding beneath the platform holding the model. To determine the new position of the platform we measured the new transform of the platform precisely using a WorldViz PPT tracking system, and three infrared lights permanently fixed to the platform. The resulting transformation was accurate to the sub-millimeter range. An image of the physical setup can be seen in Figure 2.

During the task, artificial tracking loss was triggered based on user movement. After a tag was successfully identified, a new random distance was generated and their position was logged. When the user moved beyond that distance from their position, a tracking loss occurred. When tracking is lost the user is presented with a semi-transparent crossed out symbol to notify them that tracking is lost. We also display the image that the system is aiming to match against in the upper right hand corner of the screen. The center of the screen retains the live video feed. An example of the interface displayed on tracking loss is shown in Figure 3. The buttons to indicate the localization of a tag become unresponsive during this time, until recovery has been achieved.

Figure 2: The environment used for both experiments. The models were obtained from the City of Sights [6] project.

Figure 3: An example of tracking lost using the keyframe-based recovery method. Note the keyframe image displayed in the upper right hand corner of the screen. The keyframe image has been highlighted in this screenshot, but was not during the experiment.

One key design choice in this experiment was how to simulate keyframe-based recovery since we did not actually use a feature-based tracker. For our keyframe-based recovery we used what we determined to be a reasonable overlap between the current image and the keyframe images. For the keyframe-based orientation we limited the orientation to overlap within 20 degrees to ensure there was reasonable overlap at which some localization or recovery methods are known to operate. With regards to position the user was allowed to move 15 cm away from the localization point, provided they would maintain the orientation overlap.

3.2 Apparatus

The study was conducted on a Windows 7 computer and displayed on a MIMO iMo Monster ten inch USB powered touchscreen display with a 1024x600 resolution. Using this display allowed the graphical interface to be driven by a desktop machine avoiding the additional latency of a wireless transmission of sensor and camera data. We used a FireFly Camera from Point Grey running at 640x480 resolution and 60 fps to capture images of the scene. The

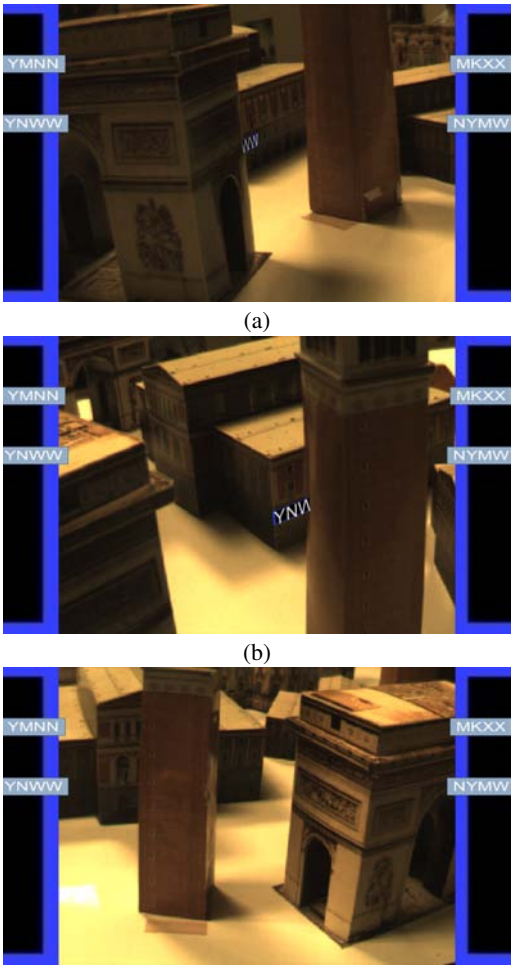


Figure 4: Several views of a tag illustrating occlusion by buildings in the scene. The participant must correctly pick the correct tag from four choices.

InterSense IS900 tracking system was used to track the tablet display. The IS900 has a 0.75 mm resolution and a static accuracy of 2.0 to 3.0 mm and a 25° RMS in Pitch and Roll, 0.50° RMS in Yaw. The update rate was 180Hz and the latency was typically 4 ms. The WorldViz PPT tracking system was used for calibrating the location of the models to the InterSense tracking system.

The display and attached camera and sensor can be seen in Figure1.

3.3 Experimental Design

We used a within-subjects experimental design. The independent variable was frequency of keyframe generation and the dependent variable was recovery time. After a formative evaluation study with several domain experts, we decided on frequencies of 1, 2, 3, and 4 seconds. For each trial, the user had to locate 10 tags which were attached to the side of a building (one at a time) (see Figure 4). The tags were texture mapped quads with a black background. Each tag displayed four randomly selected capital letters on a single side. The placement of the tags were selected from a set of 50 total placements. These placements were hand selected in order to ensure that they were reasonable. When a tag was found, the participant had to select the corresponding virtual button on the display, as shown in Figure 4. We used a random selection of placements for each trial to avoid the possibility of users learning the location of the tags. When selecting the next placement of a tag, we constricted the choice so that the next tag was rotated more than 45 degrees from the previous tag. This reduced the chance that the next tag would be visible from the location at which the previous tag was seen, to encourage participants to physically move in order to view the next tag. During a single trial the frequency of simulated keyframe logging was kept the same, but changed after each trial. The frequency was presented to each participant at the beginning of each trial through a textual note between trials. We informed the users of the frequency as we assumed that it was reasonable the users would be aware of the functionality of the tracking system they were using at a particular time. This would allow them to adjust their actions for recovery according to their expectations for the positions of the localization points. There were three sets of trials for each participant, with each set consisting of the four different frequencies. This resulted in 12 trials for each participant. A Latin square was used to remove ordering as a possible effect.

3.4 Participants and Procedure

Participants For the keyframe evaluation we evaluated twenty participants ranging in age between 18 and 31 years old with 17 females and 3 males. We will discuss the possible effects from gender differences in the conclusions. On a scale of 1 to 5 with 1 being the least familiar, users ranked their familiarity with modeling software as 1.3. On a similar scale users ranked their familiarity with games as 1.65, and their familiarity with augmented reality as 1.25. The users were more familiar with tracking systems such as provided by the Nintendo Wii, XBOX Kinect, and Playstation Move, with an average ranking of 2.85.

Procedure At the beginning of the study, participants were given a questionnaire collecting basic relevant information. The study administrator then explained the task to the participant and calibrated the city height for the participant if necessary. The model city was adjusted to chest level. Participants were then given a single trial for training purposes. For this, the frequency of the keyframes was set to 4 seconds, the most difficult. After completing the training phase, the study administrator explained the remaining tasks. After each set of trials, the participants were encouraged to take a break. At the conclusion of the third and final set of trials, the post-questionnaire was administered and the experiment concluded.

3.5 Results

The results from the experiment can be seen in Figure 5. From these results, we can see a clear difference between a keyframe frequency of 1 second and the three longer keyframe frequencies used. The results of a single-factor ANOVA indicate there was a significant effect of keyframe frequency on recovery time, $F(3,2068) = 20.592$, $P < 0.001$, and partial n^2=0.029. However we can also see that there was very little difference between 2, 3, and 4 second keyframe frequencies. As can be seen in Table 1, the results of a pairwise Tukey post-hoc evaluation indicate that there was only a significant difference between the frequency of 1 second and all other frequencies.

Given the frequency at which the keyframes are captured relative to the speed of movement of the users, it is possible that when tracking is lost, users are close enough to a localization position that the recovery seems to occur automatically. The user may not even become aware that tracking has been lost. Note that this is an actual possibility in the systems we are striving to model. These automatic localizations are measurable by inspecting localization time and account for part of the significant difference between the frequency of 1 second to the other frequencies.

In some cases, at short distances users are also able to quickly move their camera view back slightly along their path in order to intersect a previously viewed location. This can be accomplished

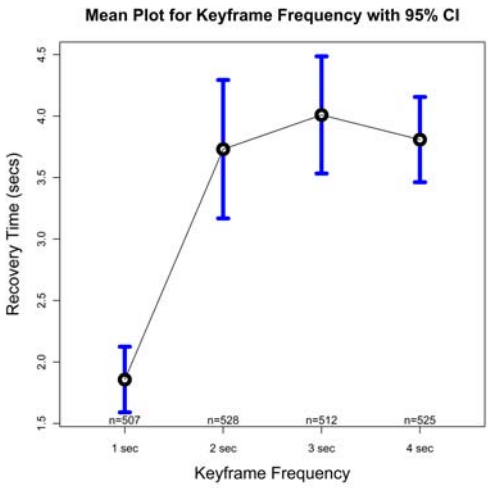


Figure 5: Mean plot of keyframe frequency over recovery time with a 95% CI. The mean recovery time for the keyframe frequencies were 1.86, 3.73, 4.01, and 3.81 seconds from shortest to longest frequency.

Frequencies	Diff	Lower	Upper	P Val <
2s to 1s	1.8732	1.0753	2.6711	0.0001
3s to 1s	2.1519	1.3479	2.9559	0.0001
4s to 1s	1.9515	1.1525	2.7505	0.0001
3s to 2s	0.2787	-0.5172	1.0746	0.8047
4s to 2s	0.0783	-0.7126	0.8692	0.9942
4s to 3s	-0.2004	-0.9975	0.5966	0.9168

Table 1: Table of the results from a Tukey post-hoc analysis of the keyframe experiment: multiple comparisons of means with a 95% family-wise confidence level.

by recognizing that tracking has been lost, but without the cognitive load of recognizing and matching the keyframe location based on the presented image.

The fact that none of the remaining differences are significant implies that once a user is presented with a keyframe to match, the cognitive load of matching their camera view to a physical representation is not dependent on the temporal or spatial distance to the keyframe. That is, participants are generally not more adept at matching nearby keyframes than keyframes that are a little further away.

3.6 Post Questionnaire Results

In the follow up questionnaire, we were interested in whether the participants experienced any discomfort, how difficult the task was, and any general comments users had about the experiment. We explicitly informed the participants to take any breaks if they felt any discomfort and all of these users finished the experiments. Four participants felt that they experienced some level of "dizziness" and five reported their arms being tired at the end. The most common comment was the annoyance the participants experienced with the experience of being tethered (an artifact of our design decision to forego wireless video transmission and tracking for the benefit of more reliably controlled and low-latency video feeds and a smaller IS900 sensor). As to the difficulty of the task, most users agreed that the task was relatively easy. Some reported that it became easier if they paid attention and remembered where they had been before. Overall, most users found the experience interesting.

3.7 Discussion

In our first hypothesis we suggested that from a constant sampling rate to some high frequency of collecting keyframes, there would be no significant difference in the recovery time. Our assumption for this statement was that within a certain frequency of keyframe collection, tracking recovery would seem to be automatic to the user. While in our evaluation we did not demonstrate this with our starting frequency of 1 second, our results suggest that if there is such an effect, it occurs below a keyframe interval of 1 second. Our second hypothesis stated that from some high frequency to some low frequency, there would be a significant effect in decreasing the frequency of the keyframe collection. This proved to be true in the case of the 1 second to 2 second increase of keyframe timing. In our final hypothesis, we stated that at some point, a low enough frequency would cause the system to be unusable due to frustration by human users. Our results indicated that there was no significant difference among the frequencies above 1 second; users were still able to use the system with even the 4 second interval between keyframes.

While there may well exist a larger interval that would exhibit another increase in recovery time, four seconds is already a very long time between keyframes, and so our results indicate that beyond a certain point, it may not be worth investing in increasing keyframe frequency. After a certain frequency the locomotive cost in moving to a location indicated by a keyframe matching was apparently insignificant compared to the cognitive load of identifying the matching location in our scenario. Therefore, if a matching system was generating a keyframe at every four seconds there would be little need to improve the system to generate keyframes at every two seconds. A much longer time period between keyframe generation may well cause walking time to again become significant; however, our results appear to indicate that at least in application cases similar to the one we evaluated, where the augmented environment is limited in scope/size, this would result in maximally a linear increase in time. It is also not clear that extending the interval beyond four seconds makes sense for practical use.

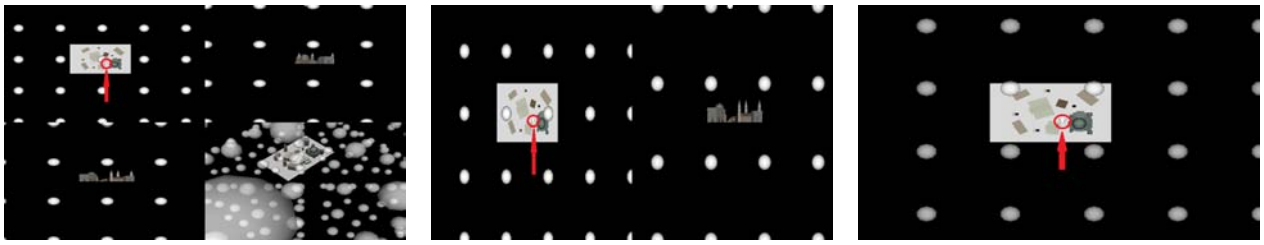
Note that it is possible for users to consciously remember the path that they have taken. This could possibly decrease recovery time as users would be less reliant on matching an image to the scene and more reliant on their motion memories. However, it is unreasonable to expect users to do this in a real world situation, where actively maintaining tracking is not an expressed goal of the application.

There are other methods for generating keyframes beyond temporal frequency. For example, Klein and Murray in PTAM [10] generate keyframes based on both the quality of the keyframes and the distance moved. We do not believe that the results of a distance-based keyframe generation procedure would differ significantly from our current results.

4 Experiment 2: Abstracted Model-based Recovery

Experiment 1 dealt with recovery systems using a traditional keyframe approach to recovery. Such approaches are necessary in situations where no pre-existing representation of the scene is present. However, in cases where stored data exists (in the form of point clouds, models, panoramas, etc.) additional approaches can be utilized. Some examples are [19] [7] [15]. The most significant difference between these approaches and a keyframe-based recovery solution is the ability to generate localization points at positions the user has not yet visited.

The density of the localization points can vary greatly depending on the dataset used and the exact nature of the localization approach. However, it is not clearly understood how in general the



(a) (b) (c)

Figure 6: Images showing the different interfaces tested for the abstracted model-based experiment. The initial interface tested used 4 views (top, side, side, perspective) similar to some modeling applications and is shown in (a). An interface showing only 2 views (top, side) is shown in (b). The final interface used in the experiment can be seen in (c). Both (a) and (b) proved too difficult during our expert user evaluations. On each image, the user's current position is indicated by a white cone. This is highlighted for the reader.

spacing of the localization points affects users' ability to recover tracking. It is possible that some systems would be largely unusable based on the density of the localization points provided for recovery. Our goal is therefore to provide a mapping between recovery point density and the ability of users to recover tracking. As in Experiment 1 we are not interested in evaluating a specific existing system. Instead, we simulated our tracking and recovery using the same physical configuration and user task as in Experiment 1.

Our hypotheses for Experiment 1 addressed the relationship of frequency of keyframes to recovery time. The frequency of keyframes was an indirect measure of distance, as participants were expected to be moving during the task. Experiment 2 measures distance as well, although directly instead. Therefore, our hypotheses regarding the abstracted model-based recovery solution were similar to the keyframe-based recovery experiment:

H2.1: From a spacing of distance zero to some small distance, we expected there to be no significant difference in the recovery time.

H2.2: From some small distance to a large distance (not specifically predicted, but hoped to be identified through controlled experimentation), we expected a significant effect from increasing the spacing distance.

H2.3: From this large distance forward, we expected that the system would no longer be usable due to frustration by human users.

Note that we do not seek to directly compare the interface or results from Experiment 1 to those from Experiment 2. Such a comparison would not be practical due to our variation of both the interface used for the localization as well as the underlying localization mechanism. Additionally, there is no clear mapping between the distances for each interface. Instead, we are aiming for general insights into the relationship between recovery point density and time to recover tracking.

Like for the keyframe-based evaluation, we simulated the both the tracking and the recovery process in our abstracted model-based recovery evaluation. In our simulations, we utilized a regular grid of localization points. Such a grid would not be found in most real-world localization solutions; however, it is necessary in order to obtain controlled measurements. Our goal is to evaluate the effect that the spacing of the localization points has on recovery. A grid provides a way to evaluate discrete settings of the density of points.

4.1 Task, Environment, and Apparatus

In the abstracted model-based localization task, we presented participants with the same task and setup as in Experiment 1. The only changes were to the solution for recovery and to the graphical interface provided when tracking was lost. This interface for the abstracted model-based localization method went through a substantial number of revisions in early pilot studies. The goal of our interface was to represent the positions of localization points surrounding the user. When tracking is working, the user is registered with the environment in both position and orientation. Once tracking is lost, we are therefore able to display the last known position and orientation of the user relative to the stored data. We represented the position and orientation using a cone with the user's view along the line from the from point to the base.

Note that the keyframe-based recovery interface and the interface for the abstracted model-based solution are very distinct. We do not seek to directly compare these solutions. Each interface is derived based on the type of available information assumed. In all of our iterations we have utilized the entire display area for the representation of the localizations points in the scene. We did not display a running video as there is little useful correlation between an individual view from the camera and the map of the localization points.

Our first attempt used localization points which were evenly placed around the participant in a three dimensional grid. Participants were shown a top down view of the scene during tracking loss, with small spheres which represented the localization points (as seen in Figure 6(c)). We evaluated this setup with seven users and quickly realized that participants had difficulty in determining the location of localization points under these conditions, as there were no cues to provide the height of the localization points. Postulating that a more well designed interface may have assisted the users in finding these points, we evaluated several alternative implementations.

Our early evaluation used a four panel configuration consisting of down, side, and front views of the scene using an orthographic projection along with a single perspective view taken at an angle, Figure 6(a). This interface is similar to what is found in most computer modeling applications. However, it is not readily suited for a mobile device with limited screen real-estate as the views can be difficult to clearly distinguish with multiple windows. We had experts from the CS and Psychology departments evaluate this setup and the general consensus was that it was difficult to use due to the small panels and information overload.

To alleviate these problems, our next interface used only the down and front orthographic views to represent the scene (Figure 6(b)). As the points were arranged in a regular grid it was possible for users to mentally align their camera along the XZ plane using the top down view then align along the Y axis by using the front view to determine the elevation of the balls. We evaluated this interface with nine participants. This proved to be too difficult for the participants, since many did not finish the experiment within the one hour time limit, and the constraints on the localization were further adjusted.

For a short evaluation of seven users, we eliminated height from the localization, in order to understand the effect this would have on user performance. For this evaluation, any height was acceptable and participants were only required to move to the correct XZ position to recover tracking. We were therefore able to use only a top down view of the scene to represent the localization points. While we found that this evaluation allowed users to localize within a reasonable time frame, eliminating an entire dimension is not a reasonable possibility for any past or future tracking system. However it is reasonable that a single height could be chosen in a realistic tracking system. An example would be choosing to use the average height of a person on a street level. We therefore limited all of the balls to a single height relative to the top of the board, and instructed users as to the location of that height. Once users were comfortable with the height of the localization points, only a top down view of the scene was needed Figure 6(c). This proved reasonably difficult for participants while being doable within our time constraints.

We used localization points with a radius of 10 cm. Given the scale of the model and based on the performance of existing localization methods, we determined that this would be a reasonable range over which localization would function. The scale of the recoverable region was not designed to match to any known tracking system. As stated earlier the density and size of the localization points varies greatly depending on the available data, and the localization method used. The important aspect is the spacing or distance between the localization points. For the abstracted model-based recovery, we did not restrict the orientation, provided the users were at least partially viewing the scene.

4.2 Experimental Design

As with Experiment 1, we also used a within-subjects experimental design. The independent variable was the spacing between localization points and the dependent variable was recovery time. We chose 0.1, 0.2, 0.4, and 0.8 meters to be the values of the distances.

Our decision to use 0.1, 0.2, 0.4, and 0.8 meters for the spacing was based on several factors. First was our decision to use localization points with a radius of 0.1 meters. This provided a lower-bound to determine performance in a very densely packed set of localization points. Our decision to increase the distance to 0.8 meters was based on early evaluations of our own performance indicating that as a very difficult distance for localization. The intervals were based on our desire to more clearly show where the transition between fast recoverability and a necessary cognitive load occurs.

4.3 Participants and Procedure

Participants For the grid-based recovery method we evaluated twenty participants ranging in age between 18 and 21 years old, with 16 females and 4 males. We will discuss the possible effects from gender differences in the conclusions section. On a scale of 1 to 5 with 1 being the least familiar, users ranked their familiarity with modeling software as 1.14. On a similar scale users ranked their familiarity with games as 1.47, and their familiarity with augmented reality as 1.10. The users were more familiar with tracking systems such as the Nintendo Wii, XBOX Kinect, and Playstation Move, with an average ranking of 2.61.

Procedure The procedure of Experiment 2 was overall very similar to that of Experiment 1. The only variation was in the implementation of the recovery method and interface. Similar to the training in Experiment 1, the training for the abstracted model-based recovery used the most difficult setting, 0.8 meters.

4.4 Results

The results from the experiment, as seen in Figure 7, indicate that recovery time increases after 0.2 meters of distance between the localization points. A single-factor ANOVA of the results showed a significant effect of distance on recovery time, $F(3,2092) = 197.91$, $P < 0.001$, and partial $n^2 = 0.221$. The graph also shows no visible difference in the recovery time between 0.1 meters and 0.2 meters distance between the localization points. A post-hoc analysis using Tukey's method, Table 4.4, showed a strong significant difference between all distances used except at 0.1 meters to 0.2 meters.

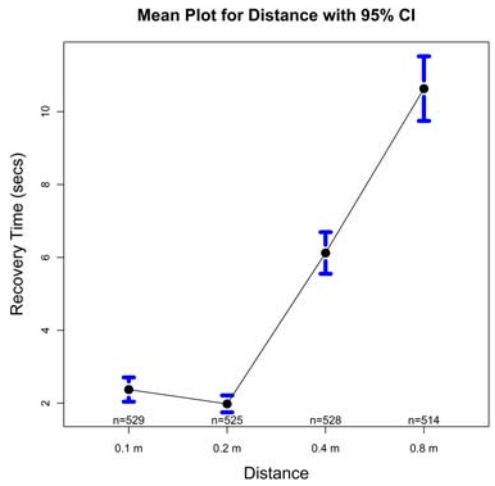


Figure 7: Graph of the mean recovery time for each distance, the abstracted model-based method, with a 95% confidence interval for the error bars.

Distances	Diff	Lower	Upper	P Val <
0.2m to 0.1m	-0.3929	-1.4264	0.6407	0.7624
0.4m to 0.1m	3.7494	2.7174	4.7815	0.0001
0.8m to 0.1m	8.2587	7.2196	9.2978	0.0001
0.4m to 0.2m	4.1423	3.1083	5.1763	0.0001
0.8m to 0.2m	8.6516	7.6106	9.6926	0.0001
0.8m to 0.4m	4.5093	3.4697	5.5488	0.0001

Table 2: Table of the results from a Tukey post-hoc analysis of the model-based recovery experiment: multiple comparisons of means with a 95% family-wise confidence level.

4.5 Post Questionnaire Results

The same questionnaire as in Experiment 1 was given to the participants of Experiment 2. Seven participants reported getting tired from holding the device and seven reported either strained eyes or slight dizziness. The wires were once again a common complaint. Most participants agreed the task was relatively easy except for when the "balls were far apart". Overall participants thought the experiment was interesting and fun.

4.6 Discussion

In our first hypothesis we suggested that from a distance of zero to some small distance there would be no significant difference in the recovery time. Our results have confirmed this in the case of 0.1 to 0.2 meters distance. For these values the localization does not involve a cognitive search task, and only requires that the users mentally disengage from their current task to perform localization.

In our second hypothesis we suggested that from some small distance to a large distance there would be a significant effect from increasing the spacing. We have shown this to be true in the case of

values greater than 0.2 meters. This is largely due to the increased cognitive load on users in locating a tracking position.

In our final hypothesis, we suggested that from a large distance forward the system would no longer be usable due to frustration by human users. This cannot be determined directly based on our evaluations. However, for values greater than 0.8 meters it is reasonable that users would be very disinclined to continue with localization. Our experience showed that it was both difficult and frustrating to do the task with greater distances.

The results from Experiment 2 are interesting and provide additional questions regarding the use of an abstracted model-based approach. At 0.1 and 0.2 meters of spacing between the localization points, the recovery times are not significantly different from each other. Since the radius of the localization points was 0.1 meters, these two conditions should have been very easy. In the 0.1 case the participants only had to move the display to the correct height. While the 0.2 meter case may have required some additional horizontal searching movement, both distances clearly did not require a cognitive matching step. Looking at the average time for localization in the final sets for both 0.1 and 0.2 meters, it seems that the average time for localization without a matching step is around 1.7 seconds. This reflects the user's need to mentally disconnect from their current task to the task of localization and the subsequent movement of the device. Note that this density of localization points is very generous. We therefore suggest that our results indicate the minimum amount of time necessary for users to perform such recovery assuming they become cognitively aware of the lost tracking.

We did not test users for spacings beyond 0.8 meters, because based on several pilot evaluations we don't expect useful data from beyond that range. We would expect that as the spacing grew larger, the recovery time would also increase until the curve flattened out. Beyond a certain point users would simply be lost or would simply have to walk to a "general" area and refine their position. At 0.8 meters, the participants averaged 10.6 seconds of recovery time and if the trends hold true it would be frustrating to ask users to do more.

5 General Discussion

Performance Trends In both of our experiments, we had hypothesized that user performance would follow an S curve. We initially stated that we would expect no significant differences at small localization point distances or high keyframe frequencies. Then we expected the recovery time to increase as the distances increased or as the frequency decreased. And finally we expected the recovery time to level off at a certain point for both distance and frequency. We predicted that the only significant differences would come from the middle region of this S curve. In both our experiments we were not able to show the entire S curve. In Experiment 1 (Figure 5), we showed the tail end of such a curve. We did not believe that capturing keyframes at a faster rate than 1 second would make practical sense and we expect that this is indeed the end of the first part of the S curve. At a mean recovery time of 1.8 seconds, it is difficult to imagine improving the recovery time. We observed that at 1 second of keyframe frequency, the recovery was almost automatic in most cases anyways. In Experiment 2, we showed the first half of the S curve. As can be seen in Figure 7, we did not reach a point where recovery time leveled off. We examined this in the discussion session of Experiment 2. In our initial evaluations, we found that users frequently could not complete the trials in a reasonable amount of time with distances greater than 1 meter.

Gender-related Issues In discussing our results from each experiment, we not yet considered effects which could have arisen from the fact that both sets of users in our experiments were mostly female (17 out of 20 for Experiment 1 and 16 out of 20 for Experiment 2). We address this issue here. The relationship between spatial understanding differences between genders is complex and cannot be generalized completely [3]. In very loose terms one might expect from the literature that males perform better with spatial tasks in both simulated and real environments while females generally perform better in spatial tasks involving maps. It can also be expected that on average males perform better with wayfinding tasks in general. In our experiments we used an AR environment which combines real and simulated content. For the most part it could be considered a real environment, since users walked around in the actual scene. The required spatial understanding task can be categorized as a wayfinding task since we are showing users a location via either an image or 3D view and asking them to navigate to that location. In both tasks, users had to move to the correct location and orient the display to roughly the correct orientation which involves the use of Visuo-Spatial Working Memory (where males in general show better results). As a result of these points, we recognize that if the gender of our subjects had been more balanced, their recovery time (as a means) may have improved but we would not expect a change in the general patterns we observed..

6 Conclusions

In summarizing our results, it is important to note that our simulations were perfect with regards to recovery success, which is to say that recovery never failed when it was in the correct position as indicated by our interface. This is not necessarily the case in practical systems. While keyframes and localization points can be selected based on their usefulness for recovery, naturally occurring changes to the scene such as differences in lighting or novel objects can result in recovery failure. The exact effect of such unreliable tracking is a subject of future work.

However, we can make some assumptions based on our current results. Using the keyframe-based approach as presented, a failure to localize would result in an additional localization step, where the user is asked to match a second image to the scene, if additional keyframes were kept. Each subsequent failure would naturally result in additional localization time. For the abstract model-based approach, a failure would require the user to find a second localization point after the user would have realized that localization could not be achieved. It seems reasonable to expect that this would not be a significant issue for densely packed localization points. For sparser points, each additional failure would require the user to cognitively locate a new point, move to it, and attempt localization.

In our abstract model-based approach, we generously constrained a single dimension (height) to make localization easier for the user. Our results indicate that even for this scenario, localization point spacings of more than 0.4 meters apart take longer to localize than with a basic keyframe-based approach. This points to the recommendation that if a specific model-based recovery approach would result in sparse recovery layouts, with localization point spacings of 0.4m and higher, it may be beneficial for users to localize based on matching images rather than attempting matching with the model-based approach at hand.

Note that this study does not consider the tolerance of specific individual users for various frequencies of tracking loss. This would also play a large role in the adoption of a tracking system by users. Even with excellent recovery, if tracking is poor, the system may be considered unresponsive and unsatisfactory to users. Previous research [16] has shown that users may be willing to tolerate up to 10 seconds of tracking loss, and even longer if there is an understood "average" time, depending on the task and application. We leave this type of evaluation for future work.

In conclusion, we have have evaluated two recovery methods to obtain how information about the density of localization points for similar recovery methods can affect recovery time. We evaluated a keyframe-based recovery method and discussed guidelines for implementing similar systems. We evaluated a representative method for model-based recovery in a controlled experimental setup and

found that such an approach requires significant time for spatial understanding, and can benefit from a well designed interface for improved usability. In future work, we would like to evaluate the differences that different recovery user interfaces can bring about in terms of user performance for typical AR tasks.

ACKNOWLEDGEMENTS

This work was supported in part by Office of Naval Research (ONR) grant N00014-09-1-1113 and US National Science Foundation (NSF) CAREER grant IIS-0747520.

We would like to thank Jonathan Ventura and Steffen Gauglitz for their time and expertise in developing the interfaces used in this paper. Lastly, we would also like to thank Dr. Mary Hegarty from the Hegarty Spatial Thinking Lab at UCSB for advice on work in spatial understanding differences between genders.

REFERENCES

[1] Junaio. http://www.junaio.com/.

[2] Wikitude. http://www.wikitude.org.

[3] E. Coluccia and G. Louse. Gender differences in spatial orientation: A review. *Journal of Environmental Psychology*, 24(3):329–340, 2004.

[4] M. O. Franz, B. Schölkopf, H. A. Mallot, and H. H. Bülthoff. "where did i take that snapshot? scene-based homing by image matching". *Biological Cybernetics*, 79:191–202, 1998.

[5] G. Y. Gehua Yang, J. Becker, and C. V. Stewart. Estimating the location of a camera with respect to a 3d model. In *3DIM '07: Proceedings of the Sixth International Conference on 3-D Digital Imaging and Modeling*, pages 159–166, Washington, DC, USA, 2007. IEEE Computer Society.

[6] L. Gruber, S. Gauglitz, J. Ventura, S. Zollmann, M. Huber, M. Schlegel, G. Klinker, D. Schmalstieg, and T. Höllerer. The city of sights: Design, construction, and measurement of an augmented reality stage set. In *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*, pages 157 –163, 2010.

[7] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:2599–2606, 2009.

[8] J. Karlekar, S. Zhou, W. Lu, Z. C. Loh, Y. Nakayama, and D. Hii. Positioning, tracking and mapping for outdoor augmentation. In *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*, pages 175 –184, oct. 2010.

[9] S. Kim, C. Coffin, and T. Höllerer. Relocalization using virtual keyframes for online environment map construction. In *VRST'09: Proc of the 16th ACM Symposium on Virtual Reality Software and Technology*, pages 127–134. ACM, 2009.

[10] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, November 2007.

[11] Layar. http://www.layar.com/, 2009.

[12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004.

[13] S. H. Park and J. C. Woldstad. Multiple two-dimensional displays as an alternative to three-dimensional displays in telerobotic tasks. *Human Factors*, 42(4):592–603, 2000.

[14] G. Reitmayr and T. W. Drummond. Going out: Robust tracking for outdoor augmented reality. In *Proc. Fifth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'06)*, pages 109–118, October 22–25 2006.

[15] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. pages 1 –7, jun. 2007.

[16] B. Shneiderman. Response time and display rate in human performance with computers. *ACM Comput. Surv.*, 16:265–285, September 1984.

[17] M. Tory, T. Moller, M. S. Atkins, and A. E. Kirkpatrick. Combining 2d and 3d views for orientation and relative position tasks. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04, pages 73–80, New York, NY, USA, 2004. ACM.

[18] C. Wu, B. Clipp, X. Li, J.-M. Frahm, and M. Pollefeys. 3d model matching with viewpoint-invariant patches (vip). *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8, 2008.

[19] C. Wu, F. Fraundorfer, J.-M. Frahm, and M. Pollefeys. 3d model search and pose estimation from single images using vip features. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pages 1 –8, 2008.

[20] J. M. Zacks, J. Mires, B. Tversky, and E. Hazeltine. Mental spatial transformations of objects and perspective. *Spatial Cognition and Computation*, 2:315–332, May 2001.