Bae *et al. mUX: The Journal of Mobile User Experience* (2016) 5:4
DOI 10.1186/s13678-016-0005-0

mUX: The Journal of Mobile User Experience

RESEARCH Open Access

# Fast and scalable structure-from-motion based localization for high-precision mobile augmented reality systems

Hyojoon Bae[1], Michael Walker[2], Jules White[2], Yao Pan[2*], Yu Sun[2] and Mani Golparvar-Fard[3]

*Correspondence:
yao.pan@vanderbilt.edu
[2]Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, USA
Full list of author information is available at the end of the article

## Abstract

A key problem in mobile computing is providing people access to cyber-information associated with their surrounding physical objects. Mobile augmented reality is one of the emerging techniques that addresses this problem by allowing users to see the cyber-information associated with real-world physical objects by overlaying that cyber-information on the physical objects' imagery. This paper presents a new vision-based context-aware approach for mobile augmented reality that allows users to query and access semantically-rich 3D cyber-information related to real-world physical objects and see it precisely overlaid on top of imagery of the associated physical objects. The approach does not require any RF-based location tracking modules, external hardware attachments on the mobile devices, and/or optical/fiducial markers for localizing a user's position. Rather, the user's 3D location and orientation are automatically and purely derived by comparing images from the user's mobile device to a 3D point cloud model generated from a set of pre-collected photographs. Our approach supports content authoring where collaboration on editing the content stored in the 3D cloud is possible and content added by one user can be immediately accessible by others. In addition, a key challenge of scalability for mobile augmented reality is addressed in this paper. In general, mobile augmented reality is required to work regardless of users' location and environment, in terms of physical scale, such as size of objects, and in terms of cyber-information scale, such as total number of cyber-information entities associated with physical objects. However, many existing approaches for mobile augmented reality have mainly tested their approaches on limited real-world use-cases and have challenges in scaling their approaches. By designing a multi-model based direct 2D-to-3D matching algorithms for localization, as well as applying a caching scheme, the proposed research consistently supports near real-time localization and information association regardless of users' location, size of physical objects, and number of cyber-physical information items. Empirical results presented in the paper show that the approach can provide millimeter-level augmented reality across several hundred or thousand objects without the need for additional non-imagery sensor inputs.

**Keywords:** Mobile augmented reality, Structure-from-motion, Direct 2D-to-3D matching, Image-based localization

## Introduction

Augmented Reality (AR) is an emerging technique that allows users to see real-world physical objects and their associated cyber-information overlaid on top of imagery of them. Mobile augmented reality is a variant of augmented reality that uses a mobile device's camera to capture real-world imagery and a mobile device's sensors to derive what cyber-information should be visible in the camera imagery, as shown in Fig. 1. A key challenge of mobile augmented reality is that it relies on precisely localizing a user in order to determine what is visible in their camera view. The localization must be performed in the field without constraining the individual's whereabouts to a specially equipped area such as custom augmented reality "caves" with pre-deployed external infrastructure for location tracking. In other words, mobile augmented reality must work regardless of users' location and environment, and deliver relevant cyber-information precisely and quickly.

Several key characteristics directly determine the reliability and utility of mobile augmented reality approaches: 1) user localization, which determines the users' viewpoint and derives what real-world physical objects are in the current scene, in order to interpret the user's surrounding contexts and deliver relevant cyber-information, 2) the speed of determining which cyber-information is associated with physical objects in order to deliver/visualize the cyber-information in the correct position, 3) the robustness of the system and ability to work with dynamically changing environments, and 4) the scalability of the cyber-physical information association system, both in terms of physical scale, such as size of objects, and in terms of cyber-information scale, such as total number of cyber-information entities associated with physical objects. The purpose of this paper is to address some of key research gaps in each of these areas that are not filled by current state-of-the-art mobile augmented reality research approaches.

A key differentiator of this research is its use of image-based localization from smartphone camera sensors and ability to localize users with respect to arbitrary marker-less 3D objects. The proposed mobile augmented reality approach, called as Hybrid 4-Dimensional Augmented Reality (HD$^4$AR), that was first developed in our prior work [1–5], provides reliable identification of the location and orientation of the user based on photographs taken by existing and already available commodity smartphones. Videos of the commercial implementation of the technology by Cloudpoint Inc., are available on YouTube: https://www.youtube.com/user/PARworks. HD$^4$AR not only provides the

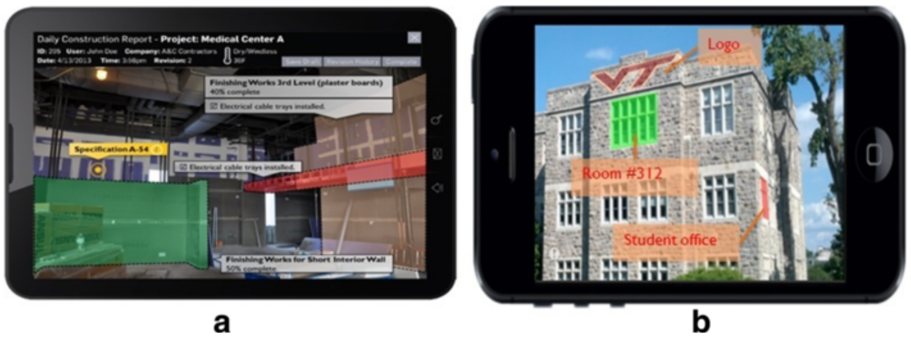

Daily Construction Report - Project: Medical Center A
ID: 205 User: John Doe Company: A&C Contractors Dry/Windless
Date: 4/13/2013 Time: 3:56pm Revision: 2 36F
Save Draft
Revision History
Complete
Finishing Works 3rd Level (plaster boards)
40% complete
☑ Electrical cable trays installed.
Specification A-54
☑ Electrical cable trays installed.
Finishing Works for Short Interior Wall
50% complete
a
Logo
Room #312
Student office
b

**Fig. 1** An example of mobile augmented reality applications. **a** Building facility management application, **b** Tourism application

location and orientation of the user, but also provides high-precision visualization of semantically-rich 3D cyber-information over real-world imagery in an augmented reality (AR) format. Rather than using imprecise mobile GPS and/or wireless sensors, as in existing mobile AR approaches, HD$^4$AR allows users to take pictures using smartphones for accurate localization in 3D and high-precision augmentation.

A limitation of prior work was that the system was difficult to scale to multiple objects without external non-imagery sensor inputs, such as GPS. These inputs were required in order to determine which model to augment against. This paper extends our prior work on HD$^4$AR in the following ways: 1) the localization speed is further increased by designing and developing a caching approach for direct 2D-to-3D matching and 2) a new multi-model augmentation approach that scales to hundreds or thousands of 3D point clouds in the system is implemented and tested. Further, the new multi-model augmentation approach does not require external non-imagery sensor inputs and is based purely on a new 2D-3D matching algorithm. The enhanced localization speed and impact of multi-model based localization are discussed in Sections 'Cached k-d tree generation for fast direct 2D-to-3D matching in model-based localization' and 'Multi-model image-based localization for blind localization requests'.

The remainder of this paper is organized as follows: Section 'Related work' discusses prior work on mobile augmented reality and open research challenges, Section 'Cached k-d tree generation for fast direct 2D-to-3D matching in model-based localization' discusses technical details of the HD$^4$AR and its caching approach, Section 'Multi-model image-based localization for blind localization requests' introduces our new multi-model scalable augmentation approach that relies on combining and/or clustering the 3D point cloud models used in the HD$^4$AR, Section 'Experimental results and validation', presents empirical results showing the speed and scalability improvements of the new approach, and Section 'Conclusion' presents concluding remarks.

## Related work

Over the past decade, many research projects related to mobile augmented reality have focused on accurate user localization to realize mobile augmented reality on various types of mobile devices. Based on the techniques used for estimating the location and pose of the user's mobile device, prior work on user localization can be roughly categorized into: 1) sensor-based localization which tracks the position using GPS and/or inertial, geomagnetic sensors attached to users, 2) marker-based localization which identifies the mobile device's camera position and orientation by leveraging pre-defined optical markers and image processing techniques, 3) visual simultaneous localization and mapping (visual SLAM) which utilizes parallel threads for simultaneously tracking and mapping visual features from images, and 4) model-based localization which uses pre-constructed 3D models of the physical world as a priori information to identify relative location and orientation of mobile devices. Table 1 summarizes and evaluates each category of prior research and presents qualitative assessment on localization accuracy and computational complexity.

The majority of prior work on user localization has relied on positioning systems, such as GPS or WLAN sensors [6, 7], or combined it with inertial measurers such as gyroscope sensors [8, 9]. Exploiting GPS sensors works well in outdoor environments but does not support indoor environments, and is unreliable in dense urban environments

**Table 1** Qualitative comparison of localization techniques for mobile augmented reality systems

| Metrics | Sensor-based | Marker-based | Visual SLAM | Model-based |
|---|---|---|---|---|
| Localization accuracy | 1.5 – 35 m[a] | 0.5 – 2 mm[b] | 0.5 – 20 mm[c] | 0.5 – 20 mm[c] |
| Localization speed | 100 – 200 msec | 20 – 140 msec | 20 – 40 msec | 5 – 240 sec |
| External infrastructure | GPS satellite | Optical markers | Not needed | Not needed |
| Resistant to drifts | × | √ | × | √ |
| Scale well to large scene | × | × | × | √ |
| Supports mobility | √ | √ | √ | × |

[a]GPS Covered area;[b]Markers within 3 m distance;[c]Objects within 10 m distance

where a clear line of sight to the GPS satellite is unavailable. In addition, the use of GPS and inertial sensors in commodity smartphones introduces significant challenges due to the limited accuracy of a GPS receiver and the noise presented in sensor data [10]. For example, the noise in geomagnetic heading values can cause jitter in onscreen information presentation. The indoor environment also imposes various challenges on location discovery due to dense multipath effects and building material dependent propagation effects. There are many potential technologies and techniques that are suggested to offer the same functionality as a GPS indoors, such as WLAN, Ultra-Wide Band (UWB) and Indoor GPS. By tagging users with appropriate receivers/tags and deploying a number of nodes (e.g., access points, receivers, transmitters, etc.) at fixed positions indoors, the location of tagged users can be tracked by triangulation [8, 11]. However, the accuracy of using network infrastructure for image-based localization is still questionable and their reliance on pre-installed infrastructures causing challenges in scalability.

In the meantime, several research groups have proposed marker-based mobile augmented reality to remove the dependency on mobile sensors or pre-installed network infrastructures [11–16]. These works track users' position and orientation using image processing techniques, i.e., matching the image captured by users' mobile devices to special, pre-defined 2D patterns (markers). Although marker-based localization has been shown to work well in both indoor and outdoor environments and does not require additional sensors, visual markers need to be attached to every real-world physical object of interest. Tagging hundreds to thousands of objects with 2D markers in the case of large-scale environments, such as street scenes, or construction site, is impractical and does not scale well to handle various distances to objects.

The advent of computer vision methods over the past decade has led to new research on the application of image-based localization methods for marker-less mobile augmented reality systems. Due to the dependency on pre-installed infrastructure, inertial measurers, and/or optimal markers, vision-based localization methods have gained significant attention in the computer vision community, as well as in the augmented reality community [12, 17–29]. A group of these works have focused on visual Simultaneous Localization and Mapping (SLAM) [19, 22, 24], which simultaneously constructs a sparse 3D map from visual features and localizes a device using generated map, with parallel threads of tracking and mapping (PTAM) [21] method. However, visual SLAM methods mostly focus on small-scale environments, such as a user's office, and suffer from inconsistent loop closure problems when the scale becomes larger, such as outdoor buildings on the street. Visual SLAM also requires the previously generated point cloud model to be hosted on the client device. This minimizes opportunity for collaboration on editing the content stored in the 3D point cloud and thus content added by one user can not be immediately accessible by others. In addition, in the context of augmented reality, visual SLAM methods are difficult to associate arbitrary 3D cyber-information with physical objects as the 3D coordinates of the map vary from the devices and their initial locations of calibration. As a consequence, visual SLAM methods require either an offline-learned 3D model or manual association of 3D cyber-information, whenever users initiate the SLAM method with different devices. Another drawback of visual SLAM methods is that the performance of localization depends on the used devices. All the computations need to be done on-board the devices, and thus, the localization speed relies on the computing power of mobile devices. The dependency on mobile devices makes visual SLAM methods difficult to scale to large-scale mobile augmented reality systems.

Another category of computer vision based work has shown that a set of overlapping images can be used to extract very accurate 3D geometry of stationary subjects, such as buildings under construction, in form of 3D point cloud model. After extracting the 3D point cloud of the subjects through a Structure-from-Motion (SfM) algorithm that estimates the 3D position of the visual features through image feature extraction, pair-wise matching, initial triangulation, and the Bundle Adjustment [25] optimization process, a 3D point cloud model can be used as a prior knowledge to compute 2D-to-3D correspondences for precisely localizing mobile camera imagery [26–29]. Using a 3D point cloud for user localization, i.e., model-based localization, permits mobile augmented reality systems to accurately estimate the 3D position and 3D orientation of the new photograph purely based on the image [1–5], and therefore, it does not have any hardware constraints on mobile devices, such as stereo cameras, GPS sensors, or motion tracking sensors. Furthermore, recent advances in SfM [30–32] enable the easy creation of large scale 3D point clouds from an unordered set of images and extend model-based localization methods to large scenes such as street-level or city-level scale.

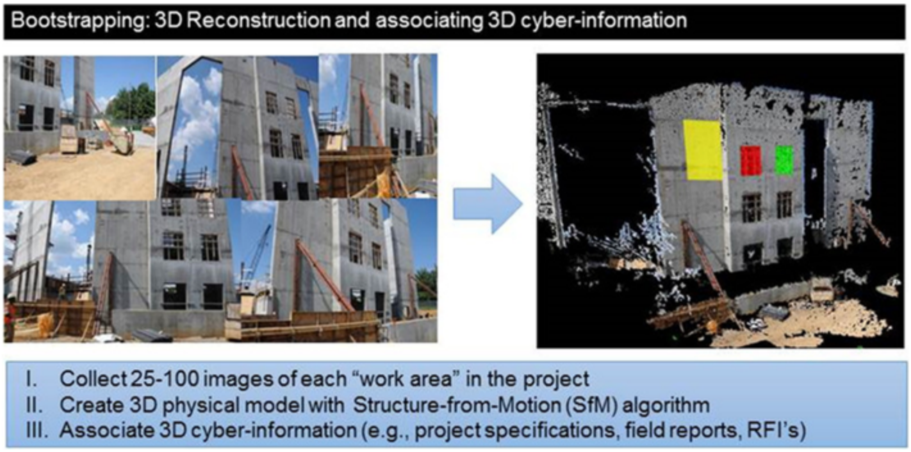
Although this body of computer vision research has shown the potential and high-accuracy of model-based reasoning, the low speed of model-based localization due to resource-intensive algorithms, such as feature extraction and 2D-to-3D matching, and the lack of on-device localization methods make them difficult to use for mobile augmented reality. In addition, very little research has examined the scalability issues of mobile augmented reality and fast cyber-physical information association with model-based localization. For example, Lim et al. [28] and Sattler et al. [29] proposed near real-time model-based localization methods. However, their test cases consist of only a single 3D point cloud model at room-level scale and their approaches were not true mobile augmented reality as they were unable to provide cyber-information delivery/visualization functionality on a mobile device. Several other recent efforts are focused on fast image-based localization using previously generated point cloud models [29, 33, 34]. These methods however do not address the problem of content authoring and their architecture does not provide an opportunity for collaborative interaction among users (by content authoring and query of information in near real-time). Applications of model-based localization methods in augmented reality systems can be found in [17, 18]. These systems were designed for context-aware architecture/engineering/construction and facility management applications to enhance construction progress monitoring processes. The 3D point cloud model is generated from pre-collected photographs of a construction site and the system uses the extracted model at street-level scale to localize users. Although their systems precisely determine the users' location and deliver relevant construction project information to end-users, it can not conduct user localization in the field for on-site decision making purposes. With their systems, field personnel have to take photographs and bring them back to the office to process each photograph. Even after field personnel bring photographs back to the office, localizing a single photograph to see the cyber-information overlaid on top of imagery takes *tens of seconds* with a high-end personal computer at the office. Considering the applications and the current limits from these works, a new approach, which takes at most 1–3 seconds regardless of operating scales and provides augmented reality with commodity smartphones is needed.

Since model-based localization methods provide sufficient accuracy for high-precision cyber-physical information association scenarios, such as identifying the buttons on a car dashboard, overlaying construction information on walls, etc., this paper will focuses on model-based localization techniques for high-precision mobile augmented reality systems. Further, these techniques do not require tagging physical objects or constraining augmentation to 2D targets. However, these techniques have not been shown to work on mobile devices at scales of hundreds of objects, which is a problem that we address in this paper. As a consequence, the objectives of this paper are to approaches that we have developed to overcome the challenges in model-based localization methods by optimizing both 3D reconstruction and the localization processes to make it possible to scale them up to hundreds or thousands of objects and deliver augmented reality to mobile devices in near real-time.

## Cached k-d tree generation for fast direct 2D-to-3D matching in model-based localization

HD$^4$AR [1–5] is an approach for augmented reality that delivers annotated photos to a user's phone, as shown in Fig. 1. Photos are captured with user's mobile devices and uploaded to server. The server uses computer vision techniques to compare the photo to the physical model and then determines what cyber-information is in view and where it should appear. Cyber-information can include various forms such as textual information (object purposes, price, building codes for construction elements), videos, audios, etc. The server then sends the photo back to the user along with the associated cyber-information. The final result displayed on the client side will be an annotated photo providing rich cyber-information. One motivation of HD$^4$AR is to help field engineering in construction sites. For example, a field engineer is concerned about the construction progress and quality of concrete foundation wall. It would be beneficial if the field engineer can query for the needed building plan information directly from the site using a picture of the foundation wall. With mobile augmented reality, the field engineer can use mobile devices to localize his position with respect to the environment and view relevant cyber-information overlaid on top of each associated construction elements. The overall procedure of HD$^4$AR is summarized in Figs. 2 and 3.

HD$^4$AR works by generating a 3D point cloud from a set of overlapping photographs of real-world physical objects with feature extraction, matching and SfM algorithms. The 3D reconstruction process combines different image features descriptors, and operate across cores in a multi-core CPU and GPU architecture for fast operations. Once the 3D physical model is available, a user can take a photo with a mobile device at a random location. HD$^4$AR uses a new image-based localization approach, which takes advantage of a



Bootstrapping: 3D Reconstruction and associating 3D cyber-information

I. Collect 25-100 images of each "work area" in the project
II. Create 3D physical model with Structure-from-Motion (SfM) algorithm
III. Associate 3D cyber-information (e.g., project specifications, field reports, RFI's)

**Fig. 2** Initial 3D reconstruction: bootstrapping of HD$^4$AR

pre-constructed 3D point cloud of target scene to identify a mobile device's relative location and orientation. The localization process compares the new photo to the generated 3D physical model and estimates the extrinsic camera parameters to find the relative position of the user's camera. In addition, the HD$^4$AR uses the client-server architecture to further increase the localization speed. The smartphone as the client uploads new photographs to the server for localization and the major image processing load is located on the server. The localization method using a direct 2D-to-3D matching algorithm takes at most few seconds to localize a photograph. After recovering a complete pose of the user's camera, the server can decide what cyber-information should appear in the user's photograph and send the cyber object and their associated information to the client. The client app will then draw cyber objects on top of the photograph.

Despite the accuracy and near real-time performance of HD$^4$AR, however, the localization speed needs to be further accelerated to provide a better user experience. With binary feature descriptors, HD$^4$AR still takes a few seconds to localize a single photograph [3, 4] against a single target model.

To support near real-time cyber-physical information association at dynamically varying environmental scales, we present a new approach for further accelerating the HD$^4$AR

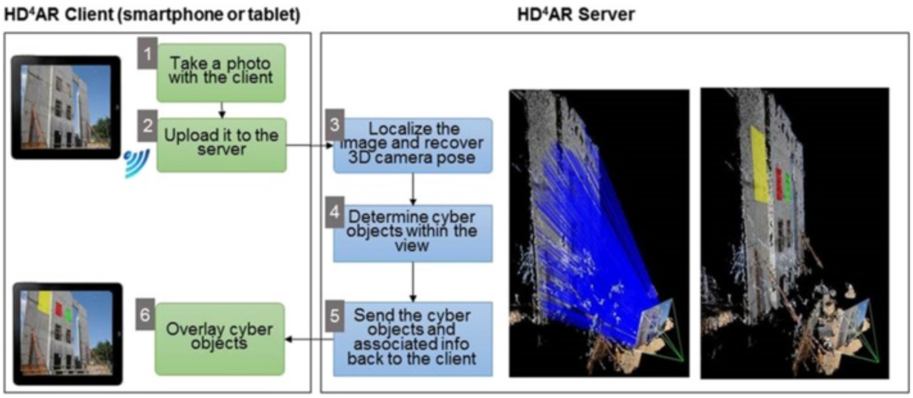

HD4AR Client (smartphone or tablet)
HD4AR Server
1 Take a photo with the client
2 Upload it to the server
3 Localize the image and recover 3D camera pose
4 Determine cyber objects within the view
5 Send the cyber objects and associated info back to the client
6 Overlay cyber objects

**Fig. 3** Image-based localization and augmentation of HD$^4$AR

localization/augmentation speed using an adaptive descriptor caching scheme. The original $HD^4AR$ localization process requires a set of resource-intensive algorithms, such as direct 2D-to-3D matching algorithms with performance that depends on the number of 3D points in the 3D physical model. As a consequence, visually rich scenes, such as outdoor data sets, typically have longer localization and augmentation times since the resulting 3D physical models are dense due to a large number of textures from the scene's physical objects.

The matching complexity of the direct 2D-to-3D matching with a k-d tree proposed in [3, 4] depends on the number of 3D points and the number of feature descriptors from a new image to be localized. Specifically, the upper bound of this matching complexity is:

$$O(MlogN) \tag{1}$$

where N is the number of 3D points in the point cloud and M is the number of feature descriptors from a new image. For outdoor data sets, the value of N is typically in the range between 30,000 and 200,000, while the value of M is 10,000–20,000. As shown in Eq. 1, larger values of N result in longer matching times. If users create a 3D physical model of a street or city using several hundred pre-collected photographs, the resulting model will consist of hundreds of thousands 3D points, and thus, a direct 2D-to-3D matching algorithm may take tens of seconds. Therefore, methods of reducing the complexity of this direct 2D-to-3D matching are needed.

### Adaptively caching 3D image descriptors using localization patterns

Removing the dependency on the number of 3D points in 1 can be expected to significantly reduce the overall matching time. To remove this dependency, we developed a new approach that generates a constant size cached k-d tree from a set of 3D representative descriptors and use it for direct 2D-to-3D matching. By caching and maintaining highly queried 3D points in a smaller k-d tree, the matching time and localization time can be reduced. Further, the cache can automatically adapt itself to cache the descriptors most commonly visible in the locations that users commonly capture imagery for augmentation. That is, the cache can adapt itself to actual usage patterns to accelerate the most common mobile augmentation perspectives.

With the proposed caching approach, a key question then becomes how to select which 3D points and their corresponding representative descriptors should be located in a cached k-d tree to provide a high localization success-ratio and accurate localization results. To provide fast and reliable localization results, therefore, the proposed approach exploits the fact that 1) $HD^4AR$ accurately and rapidly localizes a new photograph with a small number of 2D-to-3D correspondences and 2) localization requests from users may have a geospatial pattern, e.g., taking a picture of a single side of a building from the side-walk and not arbitrary locations. As a consequence, the most frequently matched 3D points during the previous localizations and their corresponding 3D representative descriptors are likely to be needed again in the future and are cached for future direct 2D-to-3D matching.

The procedure of caching 3D points and their corresponding representative descriptors can be summarized as follows:

1. After the 3D reconstruction process of HD$^4$AR, create a "cache" list with size equal to the number of 3D points in the 3D physical model. Each element of the list consists of (hit count, Index of 3D point) pair. The list will is continually updated as HD$^4$AR is used to localize images.
2. After the direct 2D-to-3D matching stage in HD$^4$AR localization, increase the hit count by 1 for all 3D points which have 2D-to-3D correspondences with the newly localized photo.
3. Sort the "cache" list in decreasing order. The upper part of the list contains the most frequently matched 3D points.
4. Extract 3D points and their corresponding representative descriptors according to the point indices of first N elements of the "cache" list. We have found that ranges of N from 1000–10,000, depending on the size of the 3D physical model, are most effective.
5. Generate a cached k-d tree using the extracted 3D representative descriptors and use it for fast direct 2D-to-3D matching.

The localization process of HD$^4$AR is slightly modified to handle fast direct 2D-to-3D matching with a cached k-d tree. Upon receiving a new photograph from the client device, the HD$^4$AR server first matches image feature descriptors of the new photograph against a cached k-d tree to find 2D-to-3D correspondences. If the number of correspondences is less than 16 or HD$^4$AR was unable to calibrate the camera with the resulting correspondences, HD$^4$AR runs a full image-based localization, as discussed in [3, 4], as a fallback solution. After the localization process, HD$^4$AR asynchronously updates the "cache" list and re-generates the cached k-d tree using the updated information.

With a cached k-d tree, the complexity of direct 2D-to-3D matching is reduced to:

$$O(MlogN) \longrightarrow O(M) \tag{2}$$

as N is constant and based on the size of the cache. Since M is the number of feature descriptors of the new photograph to be localized and is completely a random number, it is difficult to remove the dependency of the matching algorithm on M. However, by creating and using a constant number 3D points in the cached k-d tree, the proposed approach can reduce matching time and maintain localization success rates and accuracy, as shown by the empirical results in Section 'Experimental results and validation'.

Figure 4 visualizes the cached 3D points after 25 random localization requests from client devices for a building on the Virginia Tech campus. The size of the cache was set to 5000 points so that the number of nodes in the cached k-d tree could not exceed 5000. From Fig. 4(b), we can infer that the user localization requests mostly took place at the one side of the building in this test scenario and indeed had a geospatial pattern. The cached k-d tree improves HD$^4$AR localization performance by up to 262 %. The details of the experimental results used to evaluate the cached k-d tree approach are discussed in Section 'Experimental results and validation'.

### Multi-model image-based localization for blind localization requests

Our past work on mobile augmented reality presented in [1–4] assumed that there is only a single 3D physical model in the system or users know which model should be used for localization and augmentation, which is a significant limitation. For example, let us

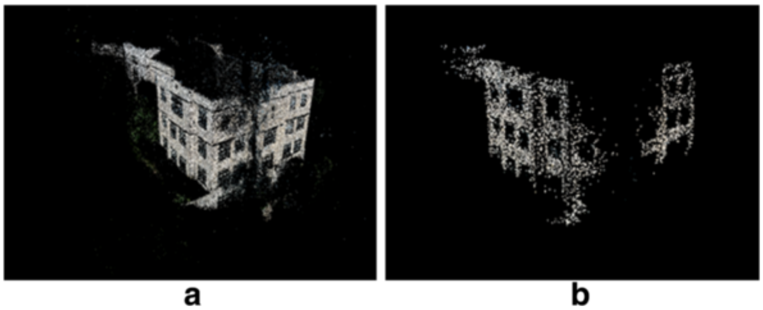


**Fig. 4** An example of a cached 3D physical model. **a** original 3D physical model, **b** caching the most frequently matched 3D points from 25 localization requests. The size of the cache is fixed at 5000 points

assume that separate point cloud models were created for different locations/objects, such as for dashboards of different cars. Users are required to choose the model from a list on the client device to enable model-based localization with respect to the corresponding 3D physical model (e.g., show me augmented reality information related to a 2011 Honda Pilot EX). This strategy is impractical when the number of physical models is enormous and/or users do not know which model should be used for localization and augmentation. To overcome this issue and provide near real-time localization/augmentation service in the presence of multiple 3D physical models, we developed a new approach, which can handle localization requests that do not know the target physical model for localization. Throughout this section, we will refer to localization requests that do not indicate the target 3D physical model as "blin" localization requests.

### Double-stage matching algorithm with a single indexed k-d tree

Our prior approach for finding an appropriate model for blind localization required matching a new image from a user's mobile device to all 3D physical models in the server. If any model successfully localized the photo (typically the first localizing model), that model was used as the basis for the localization information returned to the mobile device. Obviously, this sequential matching approach is very time-consuming and is inefficient if there are a large number of potential target models. Specifically, the upper bound of this sequential matching complexity is:

$$O(KMlogN) \tag{3}$$

where K is the number of models that exist in the server, N is the number of 3D points in each physical model, and M is the number of feature descriptors from a new image to be localized. For outdoor data sets we studied in [1–5], the value of N is typically in the range between 30,000 and 200,000 while the value of M is 10,000–20,000.

Instead of time-consuming sequential matching, our new approach creates a single indexed k-d tree and uses a matching heuristic algorithm to find the target model for blind localization requests. Specifically, a single k-d tree is created by concatenating all 3D representative descriptors from multiple models and model index information is imposed on each 3D representative descriptor to track its model of origin. When a new image is matched against this combined index, each time a descriptor matches the new image, a match count is incremented for the corresponding model that contributed that feature descriptor to the k-d tree. After the image is matched against this single indexed k-d tree, either the single 3D physical model that has the largest number of 2D-to-3D matches or all models above a threshold number of matches (e.g., for scenes with elements from multiple models) are used for localization and augmentation. Then, the image-based localization or caching approach discussed in Section 'Cached k-d tree generation for fast direct 2D-to-3D matching in model-based localization' can be used to localize a given photograph within the model(s) selected for use as the basis of augmentation. The procedure of this double-stage matching algorithm with a single indexed k-d tree can be summarized as follows:

1. Concatenate all 3D representative descriptors from all 3D physical models in the HD$^4$AR server. Also, the model index contains a mapping from each descriptor to its model of origin.
2. Upon receiving a blind localization request from the client, perform direct 2D-to-3D matching between the given image and the generated index k-d tree.
3. Using the derived 2D-to-3D correspondences from the matching process and the model index information, count the number of 2D-to-3D matches for each 3D physical model.
4. Take N models that meet a selection criteria, such as a "most matches" or "above threshold" and then perform the image-based localization for each model in parallel.
5. Select localization results which are within a re-projection error threshold and return them to the client.

The proposed double-stage matching algorithm can be reduced to a single-stage matching as the result of first-stage matching already including the 2D-to-3D correspondences of the target model (e.g., reuse the first stage descriptor matches in the second stage localization process rather than recomputing these matches from the model's individual k-d tree). However, the reason for the double-stage matching is that several models can have very similar visual features and thus are not clearly distinguished from each other through first-stage matching, which may reduce the number of matches against any given individual model. For example, if two 3D physical models A and B are created for different sides of the same building, it is possible that some of 2D-to-3D correspondences found in the first-stage matching correspond to physical elements that appear in both models, however only the strongest matching descriptor in the entire k-d tree is considered a match against the new image, even if multiple models have descriptors that could match against the imagery of that physical element in the client's photo. The reduced number of 2D-to-3D correspondences then decreases the accuracy of localization. Therefore, we utilize the first-stage matching results only for finding candidate target models and perform the second-stage matching in parallel to get the most accurate localization results possible.

With the proposed approach, the complexity of blind localization is reduced to:

$$O(KMlogN) \longrightarrow O(MlogK + 2MlogN) \tag{4}$$

where K is the number of models that exist in the server, N is the number of 3D points in each physical model, and M is the number of feature descriptors from a new image. The details of the performance gain provided by the proposed single indexed k-d tree approach will be fully discussed in Section 'Experimental results and validation'.

## Experimental results and validation

### Experiment with cached k-d trees

This section presents results from experiments with the proposed caching approach for fast model-based localization using direct 2D-to-3D matching. In order to assess improvements provided by the proposed approach, HD$^4$AR image-based localization was performed on both cached models and non-cached models. In addition, only outdoor models were considered during this experiment as the outdoor models typically have larger number of 3D points and are more computationally expensive to use for localization compared to indoor models. The details of the 3D physical models used in this experiment are summarized in Table 2. In order to minimize feature extraction time during localization, the BRISK (Binary Robust Invariant Scalable Keypoint) [35] descriptor was used in this experiment. The outdoor photographs were collected by smartphones to generate 3D physical models for this experiment and half of the photographs were randomly selected to pre-train the "cache" list discussed in Section 'Multi-model localization for blind localization requests'. All experiments were conducted on a single Amazon EC2 instance server with 22.5 GB memory and two Intel Xeon X5570 processors running Ubuntu version 12.04. An NVIDIA Tesla M2050 graphic card was used for GPU computations. The fallback solution - returning to normal model-based localization when the proposed caching approach failed to localize the photograph - was disabled during the experiment to assess the effect of the cache size on the localization success ratio. During the experiment, different cache sizes, i.e., 1000, 2000, 5000, and 10,000 points, were tested to validate the effect of the cache size on the performance, localization success rate, and the accuracy of the localization results.

Table 3 compares the results of the caching approach on the "patton" model, which is a building on Virginia Tech's campus with 46,318 3D points. As shown in Table 3, the proposed caching approach achieved the fastest localization time with the smallest cache size, while mean re-projection error remained at a similar level to that of localizations without cache. However, the localization success-ratio with smallest cache size, i.e., 1000–2000 points, was lower than with the non-cached localization. This reduction in the localization success rate is due to the fact that a pre-trained cache does not properly cover the entire target scene as we selected random photographs for caching 3D points. Nevertheless, the caching approach achieved an 80–98 % localization success ratio and was 118-126 % faster than the non-cache localization in all cases. This means that the cached approach can speed up 80–98 % of request by at least 118 %.

To further demonstrate the performance improvement of the cache-based matching, we also measured elapsed times for each step in localization, i.e., feature extraction time, and the matching/calibration time. As shown in Table 4, the matching and calibration speed is improved by the caching approach, while the feature extraction time remains constant. Therefore, we can conclude that the proposed approach, which uses a cached

**Table 2** 3D physical models tested for direct 2D-to-3D matching with a cached k-d tree approach

| Model name | Number of base images | Number of 3D points | Mean re-projection error from 3D reconstruction |
|---|---|---|---|
| patton | 40 | 46318 | 0.498 pixels |
| knu | 50 | 33122 | 0.552 pixels |
| parliament | 52 | 234343 | 0.606 pixels |

**Table 3** Performance comparison of image-based localization approaches for the "patton" model

| Package | HD$^4$AR | HD$^4$AR with caching approach | | | |
|---|---|---|---|---|---|
| Cache size | - | 1000 | 2000 | 5000 | 10,000 |
| Localization | 49/50 | 40/50 | 44/50 | 49/50 | 49/50 |
| success-ratio | (98 %) | (80 %) | (88 %) | (98 %) | (98 %) |
| Mean number of | 2145 | 134 | 228 | 438 | 748 |
| 2D-to-3D matches | | | | | |
| Mean re-projection error | 0.812 pixels | 0.962 pixels | 0.927 pixels | 1.047 pixels | 1.060 pixels |
| Mean localization time | 2.312 sec | 1.314 sec | 1.484 sec | 1.692 sec | 1.836 sec |
| (sequential requests) | (1×) | (1.760×) | (1.558×) | (1.366×) | (1.259×) |
| Mean localization time | 0.754 sec | 0.477 sec | 0.547 sec | 0.583 sec | 0.627 sec |
| (parallel requests)[a] | (1×) | (1.581×) | (1.378×) | (1.378×) | (1.203×) |

[a] Handled 4 concurrent requests by multi-threading

k-d tree for matching, reduces overall localization time by reducing the search space of direct 2D-to-3D matching. If we only consider the direct 2D-to-3D matching procedure, the matching/calibration time was up to 2.887 times faster than the non-cache localization, which is an even more significant speedup. Figure 5 visualizes the cached 3D physical models with different cache sizes. As expected, the smaller cache sizes produced sparser 3D point clouds, but the proposed approach successfully localized most of the photographs even with these sparse cached 3D point clouds.

Tables 5 and 6 compare the detailed results of the caching approach on the "knu" model, which 33,122 3D points. Again, the caching approach achieved the fastest localization time with the smallest cache size, while mean re-projection error was slightly increased. For the "knu" model, however, the localization success-ratio did not decrease for the smaller cache sizes. As shown in Fig. 6, the cached 3D models were well-trained and covered the entire target scene even when cache size was 1000 points. The performance gain of the caching approach is 118-158 % on localization and 131–226 % on direct 2D-to-3D matching. As the "knu" model has fewer 3D points than "patton" model, the performance gain was slightly lower. However, the proposed approach was faster than the non-cached localization approach and achieved an overall localization time under 1 sec for the "knu" model.

Finally, the proposed caching approach was applied to a large-scale model, i.e., the "parliament" model. The number of 3D points in the "parliament" model is 234,343 points. Tables 7 and 8 compares the results of the caching and non-caching approaches on the "parliament" model and Fig. 7 presents the cached 3D physical models with different cache sizes. As shown in Tables 7 and 8, the cache-based localization significantly improved the localization speed and matching speed for "parliament" model. The proposed approach was 196–262 % faster than the non-cached localization approach and the direct 2D-to-3D matching was up to 465 % faster. In addition, the mean re-projection

**Table 4** Details of localization time for sequential requests on "patton" model

| Package | HD$^4$AR | HD$^4$AR with caching approach | | | |
|---|---|---|---|---|---|
| Cache size | - | 1000 | 2000 | 5000 | 10,000 |
| BRISK feature extraction time | 0.785 sec | 0.785 sec | 0.785 sec | 0.785 sec | 0.785 sec |
| Matching/localization time | 1.527 sec | 0.529 sec | 0.698 sec | 0.907 sec | 1.050 sec |
| (performance gain) | (1×) | (2.887×) | (2.188×) | (1.684×) | (1.454×) |

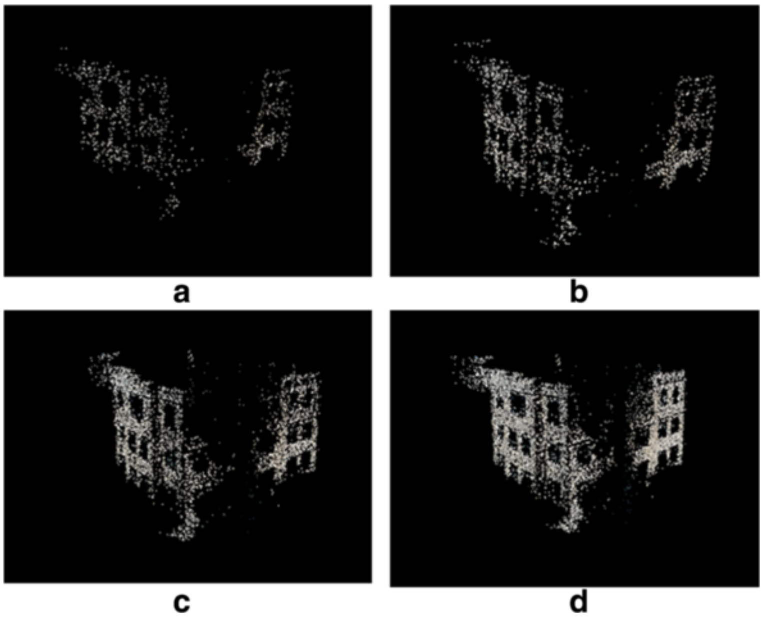


**Fig. 5** Cached 3D physical models of the "patton" model. **a** cache size = 1000 points, **b** cache size = 2000 points, **c** cache size = 5000 points, and **d** cache size = 10,000 points

error was similar to that of the non-cache localization even with a cache size of 1000 points. From these results, we can conclude that the proposed caching approach improve the performance of image-based localization on large-scale physical models and provides reliable and accurate localization results.

To illustrate the outputs of these experiments in a mobile augmented reality format, the 3D physical models associated with 3D cyber-information are shown in Fig. 8(a). Figure 8(b) illustrates the HD$^4$AR localization results in 3D space and corresponding augmented photographs are shown in Fig. 8(c). In addition to experimental results shown in Tables 3, 4, 5, 6, 7 and 8, the augmented photographs empirically show that camera poses were successfully recovered, and thus the cyber-information, e.g., window information on the "patton" model, is precisely overlaid on photographs from different viewpoints.

**Table 5** Performance comparison of image-based localization approaches for the "knu" model

| Package | HD$^4$AR | HD$^4$AR with caching approach | | | |
|---|---|---|---|---|---|
| Cache size | - | 1000 | 2000 | 5000 | 10,000 |
| Localization success-ratio | 50/50 (100 %) | 49/50 (98 %) | 50/50 (100 %) | 50/50 (100 %) | 50/50 (100 %) |
| Mean number of 2D-to-3D matches | 1204 | 87 | 157 | 338 | 561 |
| Mean re-projection error | 1.070 pixels | 1.457 pixels | 1.504 pixels | 1.536 pixels | 1.396 pixels |
| Mean localization time (sequential requests) | 1.347 sec (1×) | 0.854 sec (1.577×) | 0.959 sec (1.405×) | 1.033 sec (1.304×) | 1.138 sec (1.184×) |
| Mean localization time (parallel requests)(a) | 0.507 sec (1×) | 0.386 sec (1.313×) | 0.414 sec (1.225×) | 0.440 sec (1.152×) | 0.470 sec (1.079×) |

**Table 6** Details of localization time for sequential requests on "knu" model

| Package | $HD^4AR$ | $HD^4AR$ with caching approach | | | |
|---|---|---|---|---|---|
| Cache size | - | 1000 | 2000 | 5000 | 10,000 |
| BRISK feature extraction time | 0.462 sec | 0.462 sec | 0.462 sec | 0.462 sec | 0.462 sec |
| Matching/localization time | 0.886 sec | 0.392 sec | 0.497 sec | 0.572 sec | 0.677 sec |
| (performance gain) | (1×) | (2.260×) | (1.783×) | (1.549×) | (1.309×) |

**Multiple-model based localization**

Multi-model based localization was tested with the proposed double-stage matching algorithm using a single indexed k-d tree discussed in Section 'Multi-model image-based localization for blind localization requests'. To emulate an environment where multiple 3D physical models exist in the server, we used a total of 200 physical models generated from the $HD^4AR$ 3D reconstruction process [3, 4]. The details of test scenarios are summarized in Table 9. The server-side of the $HD^4AR$ for localization was running on Ubuntu version 12.04 with 8 GB memory and a 4-core Intel i5-2520M processor. Also, BRISK descriptors were used for this experiment.

In order to validate that the proposed double stage matching approach can successfully find target models for blind localization requests, a group of photos that could be successfully localized against any single model were tested without designating the target models. In addition, only the performance of sequential localizations from a single client device were measured. Table 10 shows the overall results of the proposed double-stage matching approach for multi-model based localizations. As shown in Table 10, the proposed double-stage matching algorithm with a single indexed k-d tree approach successfully found target models for all blind localization requests regardless of the number of models in the system. In addition, the proposed approach rapidly and accurately localized all tested photographs even in the presence of 200 models in the system. In comparison, all

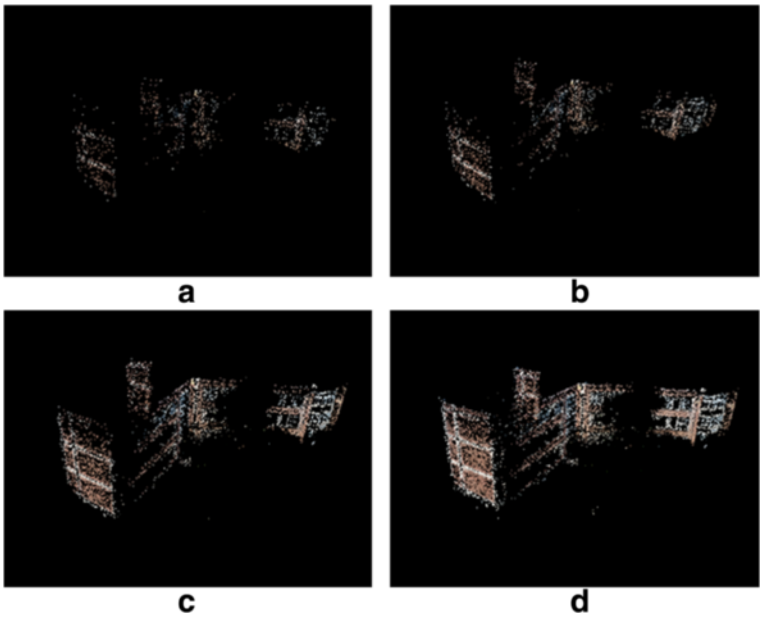


**Fig. 6** Cached 3D physical models of the "knu" model. **a** cache size = 1000 points, **b** cache size = 2000 points, **c** cache size = 5000 points, and **d** cache size = 10,000 points

**Table 7** Performance comparison of image-based localization for "parliament" model

| Package | HD$^4$AR | HD$^4$AR with caching approach | | | |
|---|---|---|---|---|---|
| Cache size | - | 1000 | 2000 | 5000 | 10,000 |
| Localization success-ratio | 40/40 (100 %) | 37/40 (92.5 %) | 37/40 (92.5 %) | 40/40 (100 %) | 40/40 (100 %) |
| Mean number of 2D-to-3D matches | 465 | 104 | 178 | 337 | 442 |
| Mean re-projection error | 0.897 pixels | 0.990 pixels | 0.906 pixels | 0.858 pixels | 0.872 pixels |
| Mean localization time (sequential requests) | 2.693 sec (1×) | 1.027 sec (2.622×) | 1.134 sec (2.375×) | 1.301 sec (2.070×) | 1.377 sec (1.956×) |
| Mean localization time (parallel requests)(a) | 0.847 sec (1×) | 0.345 sec (2.455×) | 0.369 sec (2.295×) | 0.415 sec (2.041×) | 0.439 sec (1.929×) |

past model-based mobile augmented reality techniques published in prior work have been demonstrated with only a single model. The mean localization times for multi-model based localizations were in the range between 1.360–2.623s and the mean re-projection errors were within 1.507–1.532 pixels.

To further demonstrate the scalability improvement of the double stage matching approach, we also measured the elapsed times for each step in localization, i.e., target model searching time, feature extraction time, and the matching/calibration time. As shown in Table 11, the target model searching time, which corresponds to the first-stage matching time in the proposed approach, only took 0.482–1.799 sec in our test scenarios where the number of models are varied from 10 to 200. As expected in Section 'Multi-model image-based localization for blind localization requests', the target model searching time is not proportional to the number of models. The number of models increased 20× and the search time only increased by roughly 3×. Even in the presence of 200 models, the target model search and localization with the proposed approach took under 2 sec. From experimental results shown in this section, we can conclude that the proposed approach successfully handles blind localization requests and provides near real-time localization/augmentation in the presence of multiple 3D physical models in the system. In addition, the experimental results imply that the double-stage matching algorithm with a single indexed k-d tree approach can be extended to hundreds of 3D physical models without significantly reducing localization performance. We believe that the technique could scale to 1000s of models, but time constraints on collecting and building models prevented us from testing above 200 models.

## Conclusion

In this paper, we presents a new vision-based context-aware approach for mobile augmented reality that allows users to query and access semantically-rich 3D cyber-information related to real-world physical objects and see it precisely overlaid on

**Table 8** Details of localization time for sequential requests on "parliament" model

| Package | HD$^4$AR | HD$^4$AR with caching approach | | | |
|---|---|---|---|---|---|
| Cache size | - | 1000 | 2000 | 5000 | 10,000 |
| BRISK feature extraction time | 0.571 sec | 0.571 sec | 0.571 sec | 0.571 sec | 0.571 sec |
| Matching/calibration time (performance gain) | 2.122 sec (1×) | 0.456 sec (4.654×) | 0.563 sec (3.769×) | 0.730 sec (2.907×) | 0.806 sec (2.633×) |

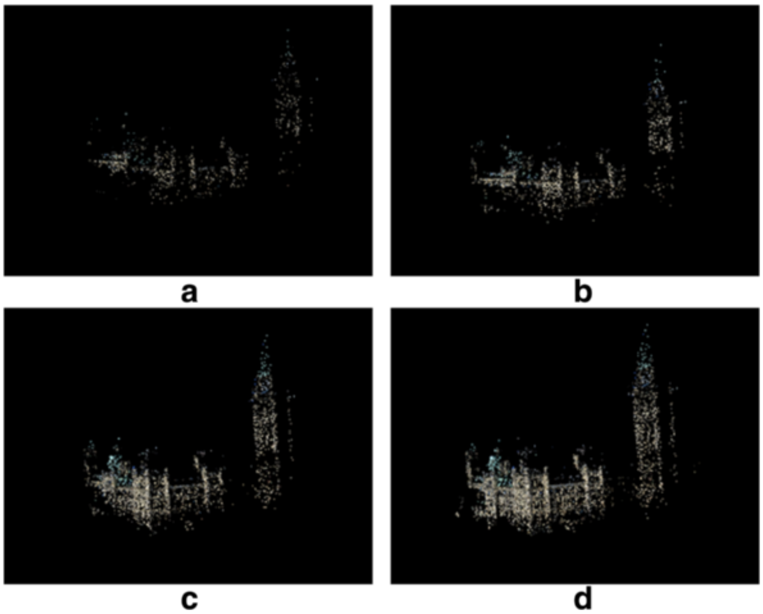


**Fig. 7** Cached 3D physical models of the "parliament" model. **a** cache size = 1000 points, **b** cache size = 2000 points, **c** cache size = 5000 points, and **d** cache size = 10,000 points

top of imagery of the associated physical objects. We design a multi-model based direct 2D-to-3D matching algorithms for localization and apply a caching scheme. The approach supports near real-time localization and information association regardless of size of physical objects, users location, and number of cyber-physical information items.

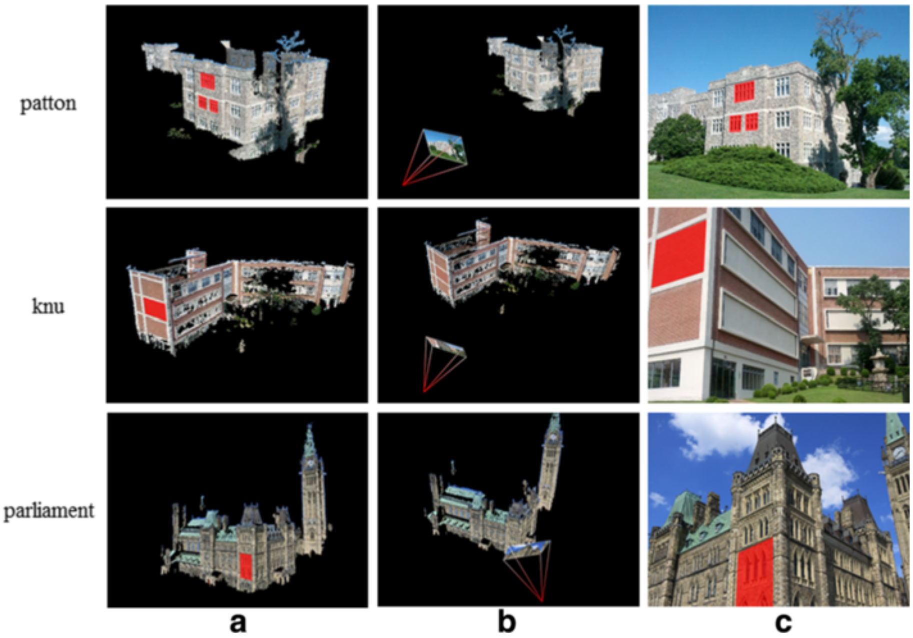


**Fig. 8** Localization/augmentation results for building-scale outdoor data sets. **a** Target 3D model associated with 3D cyber-information, **b** Image-based localization result from the HD$^4$AR server, and **c** Augmentation results from the HD$^4$AR mobile client. Point cloud models shown are improved for density using multi-view stereo algorithm

**Table 9** 3D physical model specifications for multi-model based localization experiments

| Number of 3D models | Total number of 3D points | Total point cloud size |
|---|---|---|
| 10 | 484006 | 201.21 MB |
| 20 | 1238784 | 503.33 MB |
| 60 | 2647207 | 1.07 GB |
| 100 | 3374138 | 1.38 GB |
| 200 | 4095305 | 1.70 GB |

Based on results presented in this paper, we can conclude that the cached k-d tree generation approach can significantly accelerate model-based mobile augmented reality approaches. No existing prior work to date attempts to improve the speed of model-based localization by tackling the complexity of direct 2D-to-3D matching. By removing the dependency on number of 3D points, the proposed approach provides near real-time localization/augmentation results regardless of number of 3D points in the 3D physical model. With the proposed approach, the localization time now takes at most 1.5 sec for large-scale physical models. In addition, it still achieves high-precision localization with an augmented reality overlay visualization error of at most a few pixels.

The results also show that the proposed double-stage matching algorithm using a single indexed k-d tree can scale up to mobile augmented reality experiences that simultaneously rely on hundreds of 3D physical models. Prior approaches have only scaled up to a single model. As shown in Section 'Experimental results and validation', the proposed double-stage matching algorithm can rapidly find target models for blind localization requests and successfully localize the photographs under 3 sec with 200 physical models in the system.

By combining these solution approaches, which simplify and speed up the process of accurately obtaining relevant cyber-information for mobile augmented reality experiences, the output of this research can be used in many practical context-aware mobile applications, such as construction progress monitoring. Since the solution approaches work with commodity smartphones and do not depend on external devices, such as GPS satellites, optical markers, or geomagnetic sensors, highly-contextual mobile experiences can be built simply and cheaply.

### Future work

While this research presents promising results toward near real-time high-precision mobile augmented reality by developing hybrid mobile/cloud model-based localization on SfM-based 3D physical models, some research challenges need to be addressed to further improve these mobile augmented reality experiences:

**Table 10** Performance comparison of multi-model based localization

| Number of models | 10 | 20 | 60 | 100 | 200 |
|---|---|---|---|---|---|
| Localization<br>Success-ratio | 235/235<br>(100 %) | 235/235<br>(100 %) | 235/235<br>(100 %) | 235/235<br>(100 %) | 235/235<br>(100 %) |
| Mean number of 2D-to-3D matches | 523 | 524 | 537 | 537 | 537 |
| Mean re-projection error | 1.531 pixels | 1.532 pixels | 1.513 pixels | 1.511 pixels | 1.507 pixels |
| Mean localization time | 1.360 sec | 1.568 sec | 2.054 sec | 2.343 sec | 2.623 sec |

**Table 11** Details of localization time from the proposed single indexed k-d tree approach

| Number of models in the system | 10 | 20 | 60 | 100 | 200 |
|---|---|---|---|---|---|
| Target model searching time | 0.482 sec | 0.738 sec | 1.222 sec | 1.509 sec | 1.799 sec |
| BRISK feature extraction time | 0.570 sec | 0.573 sec | 0.571 sec | 0.578 sec | 0.566 sec |
| Matching/calibration time | 0.308 sec | 0.257 sec | 0.261 sec | 0.256 sec | 0.258 sec |

1. *Real-time localization/augmentation*: although the HD$^4$AR achieves near real-time localization regardless of environmental constraints, some applications, such as AR-based video gaming, may require real-time augmented reality rather than still photo AR. A possible solution is to develop a hybrid approach that uses HD$^4$AR for the at-scale search, identification of target models in the scene, and initial 6DOF positioning and then a faster on-device tracking approach to provide real-time visualization. For example, key frames in the camera video stream could be localized through the model-based approach proposed in this study while intermediate frames are localized through an on-device tracking approach that relies on the HD$^4$AR localization and model search results.
2. *Minimal number of base images*: we typically collected 50–100 images or 3-5s of 1080p video for each target scene to produce 3D physical models. These imagery capture heuristics came from experience, and therefore, the relationship between number of base images and the quality of 3D point cloud should be further analyzed to guide users to in determining the minimal number of base images needed for reliable model-based localization.
3. *Robustness against reflective or translucent surfaces*: HD$^4$AR is based on intensity-based image feature descriptors, such as SIFT, SURF, FREAK, or BRISK, which compare the intensity of pixels to discover orientation and response of feature points. As a consequence, the proposed approach may not work well with images that only show reflective surfaces such as metals, mirrors, or glass curtain walls of a building. These surfaces reflect all surrounding scenes and make the system difficult to find correspondences among the images. One possible method to address this is to require images to be taken farther from these elements so other non-reflective elements can also be presented in the scene.

Videos of the commercial implementation of the technology by Cloudpoint Inc., are available on YouTube: https://www.youtube.com/user/PARworks.

**Acknowledgements**
This paper is based upon work supported by the National Science Foundation under National Science Foundation (NSF) Civil, Mechanical, and Manufacturing Innovation CMMI-1360562 award. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the writers and do not necessarily reflect the views of the National Science Foundation.

**Authors' contributions**
HJ designed and implemented HD$^4$AR system, designed the experiments and drafted the manuscript. MW helped carry out experiments. JW participated in the design of the system and provide critical revision to the manuscript. YP helped conduct experiments and revised the manuscript. YS helped collect experiment data and revised the manuscript. MG participated in the design of the system and provide some critical revision to the manuscript. All authors read and approved the final manuscript.

**Competing interests**
The authors declare that they have no competing interests.

**Author details**
[1]Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, USA. [2]Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, USA. [3]Department of Civil and Environmental Engineering and the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, USA.

Received: 22 February 2016 Accepted: 18 May 2016
Published online: 19 July 2016

**References**

1. Bae H, Golparvar-Fard M, White J (2012) Enhanced hd4ar (hybrid 4-dimensional augmented reality) for ubiquitous context-aware aec/fm applications. In: Proceedings of 12th International Conference on Construction Applications of Virtual Reality (CONVR 2012). National Taiwan University Press, Taiwan. pp 253–26
2. Bae H, Golparvar-Fard M, White J (2013) High-precision and infrastructure-independent mobile augmented reality system for context-aware construction and facility management applications. In: Proceeding of the 2013 ASCE International Workshop on Computing in Civil Engineering. American Society of Civil Engineers, Los Angeles. pp 637–644
3. Bae H, Golparvar-Fard M, White J (2013) High-precision vision-based mobile augmented reality system for context-aware architectural, engineering, construction and facility management (aec/fm) applications. Visual Eng 1(1):1–13
4. Bae H, Golparvar-Fard M, White J (2014) Rapid image-based localization using clustered 3d point cloud models with geo-location data for aec/fm mobile augmented reality applications. In: in Proceedings of the International Conference on Computing in Civil and Building Engineering. American Society of Civil Engineers, Orlando. pp 841–849
5. Bae H, Golparvar-Fard M, White J (2014) Image-based localization and content authoring in structure-from-motion point cloud models for real-time field reporting applications. J Comput Civil Eng. ASCE, Reston. B4014008
6. Behzadan AH, Kamat VR (2007) Georeferenced registration of construction graphics in mobile outdoor augmented reality. J Comput Civil Eng 21(4):247–258
7. Khoury HM, Kamat VR (2009) High-precision identification of contextual information in location-aware engineering applications. Adv Eng Inform 23(4):483–496
8. Akula M, Dong S, Kamat VR, Ojeda L, Borrell A, Borenstein J (2011) Integration of infrastructure based positioning systems and inertial navigation for ubiquitous context-aware engineering applications. Adv Eng Inform 25(4):640–655
9. Ojeda L, Borenstein J (2007) Personal dead-reckoning system for gps-denied environments. In: IEEE International Workshop on Safety, Security and Rescue Robotics, SSRR 2007. IEEE, Rome. pp 1–6
10. Gotow JB, Zienkiewicz K, White J, Schmidt DC (2010) Addressing challenges with augmented reality applications on smartphones. In: Mobile Wireless Middleware, Operating Systems, and Applications. Springer, Berlin Heidelberg. pp 129–143
11. Chen Y, Kamara JM (2011) A framework for using mobile computing for information management on construction sites. Autom Constr 20(7):776–788
12. Arth C, Schmalstieg D (2011) Challenges of large-scale augmented reality on smartphones. In: the 10th IEEE International Symposium on Mixed and Augmented Reality Workshop (ISMAR). IEEE, Basel
13. Feng C, Kamat VR (2012) Augmented reality markers as spatial indices for indoor mobile aecfm applications. In: Proceedings of 12th International Conference on Construction Applications of Virtual Reality (CONVR 2012). National Taiwan University Press, Taiwan. pp 235–242
14. Lee S, Akin Ö (2011) Augmented reality-based computational fieldwork support for equipment operations and maintenance. Autom Constr 20(4):338–352
15. Yabuki N, Miyashita K, Fukuda T (2011) An invisible height evaluation system for building height regulation to preserve good landscapes using augmented reality. Autom Constr 20(3):228–235
16. Hakkarainen M, Woodward C, Billinghurst M (2008) Augmented assembly using a mobile phone. In: 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, 2008. ISMAR 2008. IEEE, Cambridge. pp 167–168
17. Golparvar-Fard M, Peña-Mora F, Savarese S (2012) Automated model-based progress monitoring using unordered daily construction photographs and ifc as-planned models. ASCE J Comput Civil Eng 2012:04014025
18. Carozza L, Tingdahl D, Bosché F, Gool L (2014) Markerless vision-based augmented reality for urban planning. Comput Aided Civ Infrastruct Eng 29(1):2–17
19. Davison AJ, Reid ID, Molton ND, Stasse O (2007) Monoslam: Real-time single camera slam. IEEE Trans Pattern Anal Mach Intell 29(6):1052–1067
20. Dong Z, Zhang G, Jia J, Bao H (2009) Keyframe-based real-time camera tracking. In: 2009 IEEE 12th International Conference on Computer Vision. IEEE, Kyoto. pp 1538–1545
21. Klein G, Murray D (2007) Parallel tracking and mapping for small ar workspaces. In: 6th IEEE and ACM International Symposium on Mixed and Augmented Reality. IEEE, Nara. pp 225–234
22. Salas-Moreno RF, Newcombe RA, Strasdat H, Kelly PH, Davison AJ (2013) Slam++: Simultaneous localisation and mapping at the level of objects. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Portland. pp 1352–1359
23. Wagner D, Reitmayr G, Mulloni A, Drummond T, Schmalstieg D (2010) Real-time detection and tracking for augmented reality on mobile phones. IEEE Trans Vis Comput Graph 16(3):355–368
24. Ufkes A, Fiala M (2013) A markerless augmented reality system for mobile devices. In: 2013 International Conference on Computer and Robot Vision (CRV). IEEE, Regina, SK. pp 226–233
25. Triggs B, McLauchlan PF, Hartley RI, Fitzgibbon AW (2000) Bundle adjustment a modern synthesis. In: Vision Algorithms: Theory and Practice. Springer, Berlin Heidelberg. pp 298–372
26. Gordon I, Lowe DG (2006) What and where: 3d object recognition with accurate pose. In: Toward Category-level Object Recognition. Springer, Berlin Heidelberg. pp 67–82
27. Irschara A, Zach C, Frahm JM, Bischof H (2009) From structure-from-motion point clouds to fast location recognition. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, Miami. pp 2599–2606
28. Lim H, Sinha SN, Cohen MF, Uyttendaele M (2012) Real-time image-based 6-dof localization in large-scale environments. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Providence. pp 1043–1050

29. Sattler T, Leibe B, Kobbelt L (2011) Fast image-based localization using direct 2d-to-3d matching. In: 2011 IEEE International Conference on Computer Vision (ICCV). IEEE, Barcelona. pp 667–674
30. Agarwal S, Furukawa Y, Snavely N, Simon I, Curless B, Seitz SM, Szeliski R (2011) Building rome in a day. Commun ACM 54(10):105–112
31. Frahm JM, Fite-Georgel P, Gallup D, Johnson T, Raguram R, Wu C, Jen YH, Dunn E, Clipp B, Lazebnik S (2010) Building rome on a cloudless day. In: Computer Vision–ECCV 2010. Springer, Berlin Heidelberg. pp 368–381
32. Snavely N, Seitz SM, Szeliski R (2008) Modeling the world from internet photo collections. Int J Comput Vis 80(2):189–210
33. Lu G, Ly V, Shen H, Kolagunda A, Kambhamettu C (2013) Improving image-based localization through increasing correct feature correspondences. In: Advances in Visual Computing. Springer, Berlin Heidelberg. pp 312–321
34. Sattler T, Leibe B, Kobbelt L (2012) Towards fast image-based localization on a city-scale. In: Outdoor and Large-Scale Real-World Scene Analysis. Springer, Berlin Heidelberg. pp 191–211
35. Leutenegger S, Chli M, Siegwart RY (2011) Brisk: Binary robust invariant scalable keypoints. In: 2011 IEEE International Conference on Computer Vision (ICCV). IEEE, Barcelona. pp 2548–2555