# Real-Time Image Localization and Registration with BIM Using Perspective Alignment for Indoor Monitoring of Construction

Khashayar Asadi, S.M.ASCE[1]; Hariharan Ramshankar[2]; Mojtaba Noghabaei, S.M.ASCE[3]; and Kevin Han, A.M.ASCE[4]

**Abstract:** Construction performance monitoring has been identified as a key component that leads to the success of a construction project. Real-time and frequent monitoring will enable early detection of potential schedule delays and facilitate the communication of progress information accurately and quickly. To facilitate as-built and as-planned data comparison, this paper proposes an automated registration of a video sequence (i.e., a series of image frames) to an as-planned building information model (BIM) in real time. This method discovers the camera poses of image frames in the BIM coordinate system by performing an augmented monocular simultaneous localization and mapping (SLAM) and perspective detecting and matching between the image frames and their corresponding BIM views. The results demonstrate the effectiveness of real-time registration of images with BIMs. The presented method can potentially fully automate past studies that automate progress inference, given visual representation of as-built models aligned with BIM. Moreover, it will facilitate communication on jobsites by associating quality and progress with visuals that are in the BIM coordinate system. **DOI: 10.1061/(ASCE)CP.1943-5487.0000847.** © *2019 American Society of Civil Engineers.*

## Introduction

Systematic and frequent monitoring of construction progress has been repeatedly reported as a major component of project controls that can help reduce schedule and cost overruns (Golparvar-Fard et al. 2015; Turkan et al. 2013; Navon 2007). Despite recent advances in information technology (IT), prevailing monitoring systems are still dominated by traditional approaches, e.g., construction managers and superintendents observe and compare as-built conditions to schedules. This process involves manual data collection and immense data extraction from schedules, construction drawings, and daily reports, which are error-prone, costly, and labor intensive. Moreover, the aforementioned processes are time-consuming and take 20% to 30% daily efforts to update the construction activities (Solihin and Eastman 2015; Golparvar-Fard et al. 2011).

Building information model (BIM) has become an essential step to the digital management of construction projects, helping facility and construction managers with decision making. As-planned (e.g., nD BIM) and as-built documentation [e.g., images, videos, and three-dimensional (3D) point clouds] provide a significant opportunity for analysis, sensing, and communication of construction performance. Collecting visual data and updating BIM are the most time-consuming parts of as-built documentation (Han and Golparvar-Fard 2017). Some efforts to automate these time-consuming tasks include vision-based methods for comparing as-built conditions with as-planned data (i.e., schedule and BIM) (Han et al. 2018; Han and Golparvar-Fard 2015a; Bosché et al. 2015; Pučko et al. 2018). The three dominant types of as-built representations are laser scanned point clouds (Bosché et al. 2015; Turkan et al. 2013), images (Kim et al. 2013a; Yang et al. 2015; Asadi and Han 2017, 2018; Brilakis et al. 2011), and image-based point clouds (Han and Golparvar-Fard 2015a, 2017).

Each of these types has its own limitations and advantages in a particular environment. For instance, although 3D laser scanners with a stationary line-of-sight provide more accurate and dense point clouds compared with the image-based point clouds, the data collection process requires sufficient knowledge of surveying theory, expensive components, and multiple individuals. Moreover, the process is time-consuming. Inefficiencies during inclement weather (e.g., rainfall) are also a limitation (Golparvar-Fard et al. 2011).

Nowadays, high resolution and cheap digital cameras have increased the frequency of collecting data (Bohn and Teizer 2010). Camera networks on construction sites, as a promising form of as-built data acquisition method, have received increasing attention in recent years (Han et al. 2018; Han and Golparvar-Fard 2017; Yang et al. 2015; Golparvar-Fard et al. 2015). Moreover, they have reduced the price of acquiring and maintaining videos and photographs (Han and Golparvar-Fard 2017). Recently, advanced data collection methods that leverage camera-equipped unmanned vehicles (UVs) for visually monitoring the construction process are proposed (Asadi et al. 2018c, d; Ham et al. 2016; Siebert and Teizer 2014). Compared to conventional manual practices, the UV applications for collecting as-is information has been proven to be safe, efficient, and cost-effective (Ham et al. 2016; Han et al. 2015).

Digital cameras enable an as-built scene to be observed from a wide range of angles and viewing position during construction.

[1]Ph.D. Student, Dept. of Civil, Construction, and Environmental Engineering, North Carolina State Univ., Raleigh, NC 27606 (corresponding author). ORCID: https://orcid.org/0000-0003-0665-1579. Email: kasadib@ncsu.edu

[2]Master Student, Dept. of Electrical and Computer Engineering, North Carolina State Univ., Raleigh, NC 27606.

[3]Ph.D. Student, Dept. of Civil, Construction, and Environmental Engineering, North Carolina State Univ., Raleigh, NC 27606. ORCID: https://orcid.org/0000-0002-2248-1840

[4]Assistant Professor, Dept. of Civil, Construction, and Environmental Engineering, North Carolina State Univ., Raleigh, NC 27606. ORCID: https://orcid.org/0000-0002-2995-8381

Note. This manuscript was submitted on September 17, 2018; approved on January 10, 2019; published online on June 13, 2019. Discussion period open until November 13, 2019; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Computing in Civil Engineering***, © ASCE, ISSN 0887-3801.

However, unless the captured images and videos are aligned with respect to as-planned data (BIM), they cannot be used for automated inference of construction progress. BIM involving the generation and management of digital representations of physical and functional characteristics of places. So, there is an essential demand for automated visual data localization with respect to BIM in real time, especially for robotic applications in the construction industry such as autonomous data collection and navigation. For this purpose, many research studies have focused on registering images and image-based point clouds to BIM (see section "Registration of As-Built and As-Planned Models"). These registered (or integrated) models have the following limitations: (1) the registration of visual data (i.e., images and videos) with BIM is not fully automated and is not in real time; and (2) the image-based 3D reconstruction in these studies has not been validated for indoor environments.

## Focus of This Study

To address the aforementioned limitations, this paper presents a framework to automate the registration of as-built video frames to as-designed BIM. This framework will serve as an enabling factor for fully automating past studies on vision-based progress inference (Golparvar-Fard et al. 2009; Han and Golparvar-Fard 2015a; Han et al. 2018). The proposed study is illustrated in Fig. 1. The video frames that are taken from a monocular camera are registered in real time to a BIM. This registration happens by perspective detection and matching between frames and their corresponding BIM views. The proposed method has a synergistic opportunity with autonomous data collection using UVs by providing real-time localization and registration with BIM. By using BIM as a priori geometry to be reconstructed, the method recovers the camera poses corresponding to the keyframes with respect to the BIM coordinate system. It also has the potential to be used as a complementary localization method to improve camera trajectory before a loop closure in SLAM (to be further detailed in the "Background" section). This improvement happens at the end when SLAM localization is finished, similar to the global bundle adjustment approach with a loop closure.

The proposed method is validated through two case studies with challenging environments for image-based localization/SLAM (i.e., lack of visual features) and perspective detection (i.e., cluttered scene): a simple hallway and an indoor construction site, respectively. The results demonstrate the robustness and feasibility of real-time localization and registration with BIM for automated construction monitoring of indoor environments.

## Background

The proposed method performs automatic registration of image sequences to BIM in real-time leveraging BIM's geometry and computer vision techniques, such as camera localization and mapping and perspective detection. This section is dissected into these two vision techniques and registration of visual data with BIM.

### *Vision-Based Camera Localization and Mapping*

The process of image-based 3D reconstruction includes camera motion estimation and as-built point cloud generation. Some 3D reconstruction techniques are not in real time but provide more accurate camera poses and dense point clouds (also called maps) (Snavely et al. 2006; Fuhrmann et al. 2014; Wu et al. 2011a). Some are in real time but provide less accurate camera poses and sparse maps (Fuentes-Pacheco et al. 2015). Certain similarities and differences between these two approaches are presented subsequently.

VisualSFM (Wu et al. 2011a) and OpenSfM (Mapillary 2018) are offline approaches to reconstruct camera motions and the structure of a scene using either ordered or unordered photographs without any strong semantic or geometric priors (Agarwal et al. 2011; Snavely et al. 2008). This automatic process leads to the interrelation of all images with each other using advanced feature descriptors, such as SIFT (Lowe 2004). These features make the image matching task to be invariant to scale and camera orientation. To minimize reprojection errors (i.e., the distance, in pixels, between a feature and its reprojected point back on the image using the estimated camera pose), a final global optimization procedure [i.e., bundle adjustment (BA) (Wu et al. 2011b)] is applied, which adjusts reconstructed features (i.e., point clouds) and camera poses in 3D.

Similarly, visual SLAM estimates camera poses of image sequences (or video) and maps an environment, but in real time. This map consists of feature correspondences from the video frames. With a rapid increase in processing power and recent advances in computer vision, visual SLAM that requires heavy computing power can now run in real time (Mur-Artal et al. 2015; Engel et al. 2014; Bresson et al. 2015; Mur-Artal and Tardós 2016; Concha and Civera 2015; Qin et al. 2017).

Large-scale direct (LSD)-SLAM (Engel et al. 2014) is based on direct image alignment using image intensities and it does not use features. Conversely, feature-based monocular (single lens) SLAM performs feature detection and matching to recover a structure from motion (Mur-Artal et al. 2015; Bresson et al. 2015; Mur-Artal and Tardós 2016).

Oriented FAST and rotated BRIEF (Binary Robust Independent Elementary Features) (ORB)-SLAM and ORB-SLAM2 are based

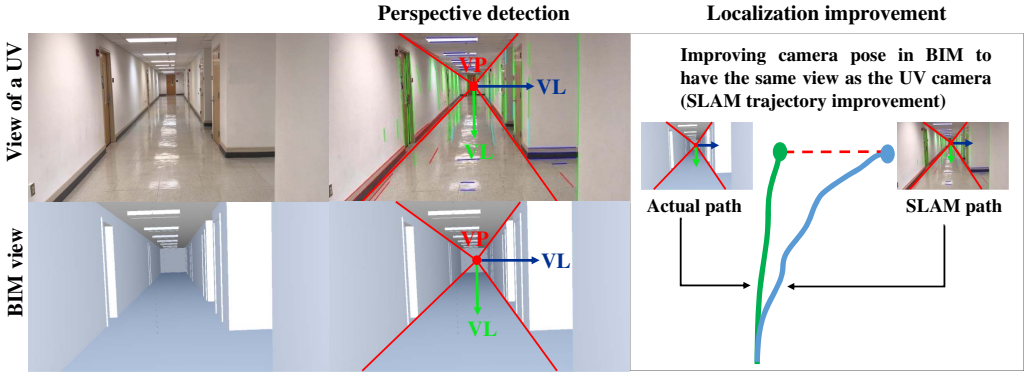


**Fig. 1.** Localizing the camera with respect to the BIM and improving the camera pose through perspective matching.

on ORB features (Mur-Artal et al. 2015; Mur-Artal and Tardós 2016) that can run in real time. For efficient and reliable computation, they use the same features for tracking, mapping, relocalization, and loop closure. A bag of words place recognition module performs loop detection and relocalization. These visual words known as visual vocabulary are created offline with the ORB descriptors extracted from a large set of images.

Monocular visual-inertial system (VINS-mono) (Qin et al. 2017) is a visual-inertial approach that uses an optimization-based sliding window formulation for providing high-accuracy visual-inertial odometry (i.e., estimating the 3D pose (translation + orientation) of a moving camera relative to its starting position, using visual features) consisting of a camera and a low-cost inertial measurement unit (IMU). It utilizes a loosely-coupled sensor fusion where the IMU is treated as an independent module to assist vision-only pose estimates obtained from the visual structure from motion. This algorithm performs best when the system uses a global shutter camera and a synchronized high-end IMU.

These visual SLAM methods generate a very sparse map, compared to the output of non-real-time methods [i.e., VisualSfM (Wu et al. 2011a) and OpenSfM (Mapillary 2018)]. One of the major challenges of visual SLAM is to deal with scaling and drift issues, especially with rotating motion, before a loop closure. When a loop is closed (e.g., a camera returns to the starting position) BA optimizes the path and minimizes scaling errors and drift. The odometry is not accurate especially in indoor scenes due to lack of light and features (e.g., a hallway with painted walls). Moreover, all monocular feature-based approaches generate point clouds and odometry in local coordinate systems, meaning that all distances are unitless.

To address these challenges, three main contributions regarding localization and mapping are provided in this paper as follows: (1) an augmented visual SLAM to improve point clouds and create more accurate camera trajectory, compared with ORB-SLAM2, (2) odometry and point cloud in the real world without postprocessing, and (3) minimized drifts before a loop closure.

### Vanishing Points Estimation

Extracting 3D information from two-dimensional (2D) images is one of the major tasks in the computer vision community. For instance, there is a line of research that estimates vanishing points (VPs) for detecting geometry of an interior space (Hedau et al. 2009; Lee et al. 2009; Zou et al. 2018). When projecting a perspective on a 2D image, parallel lines intersect at a vanishing point either inside or outside of the image plane (Rother 2002). For instance, the parallel lines can be detected by intensity gradients of the pixel units (Tuytelaars et al. 1998; Coughlan and Yuille 1999) or edge detection (Košecká and Zhang 2002; Rother 2002; Denis et al. 2008; Hedau et al. 2009).

In this paper, the Hedau et al. (2009) algorithm is implemented, which finds three vanishing points corresponding to three mutually orthogonal directions in an input image with a robust search and voting scheme.

### Registration of As-Built and As-Planned Models

During the past decade, research studies have addressed vision-based approaches for registering visual data and BIM. These methods are based on conformity of geometric primitives such as points, lines, and planes to determine the translation, rotation, and scale in reference to as-planned models. The primary limitations of these methods are manual operations and predefined viewpoint assumption (known position and orientation) about the camera (Lukins and Trucco 2007), such as installation of cameras on jobsites (Podbreznik and Rebolj 2005; Lukins and Trucco 2007) and finding correspondences between visual data and BIM (Rebolj et al. 2008).

Lukins and Trucco (2007) used a fixed camera and a pose estimation algorithm (David and DeMenthon 2005) to classify scene changes as structural changes related to the 3D model. To address the failed alignment due to the complex building models and cluttered scenes, they performed an evolutionary optimizer known as particle swarm optimization (PSO) (Shi 2004). Ibrahim et al. (2009) proposed a method that acquired prior knowledge of building components and occupancy within a scene from a four-dimensional (4D) building model registered to the camera. The paper argued that the fixed cameras are inefficacious for tracking progress on construction sites due to inflexibility in response to changing structures.

In contrast to previously described methods, Golparvar-Fard et al. (2009) used automatically reconstructed point clouds to manually register a large set of unordered images to a 4D BIM. Bosche et al. (2009) registered laser scanner's point clouds to the 3D building model using manually selected point correspondences. Bosché (2010) extended this method to make it fully automated. The registration process split into a coarse and a fine step based on an iterative closest point (ICP) method. A good initial guess was necessary for the ICP algorithm to converge (Yang et al. 2011), which was the primary constraint in this automated method. Bosché (2012) made the coarse registration step semiautomated by one-click random sample consensus (RANSAC)-based scan plane extraction method. Kim et al. (2013a) provided a common data layout in a fully automated registration similar to Bosché (2010) in a different manner by resampling the 3D building model and the point cloud. This method also was not applicable in several cases (Bellekens et al. 2014). In this context, Pătrăucean et al. (2015) provided a general overview of the as-built generation of BIM in the availability of as-designed BIM with the focus on the geometric modeling aspects.

A number of other research contributions dealt with the estimation of the camera pose for each image using perspective-'n'-point/line (PnP/PnL) method (Chen 1990). This algorithm is based on correspondences between the 3D model features and their projections in the image. A set of at least three correspondences is necessary to solve this problem. The accuracy improves by increasing the number of features, except adding outliers correspondences that negatively affect the accuracy (Přibyl et al. 2015). Liu et al. (1990) used eight line correspondences in their unique solution. This solution was improved by decreasing the number of line correspondences to four (Ansar and Daniilidis 2003), and the effect of outliers decreased in Mirzaei and Roumeliotis (2011). Zhang et al. (2012) presented an efficient and accurate solution for the PnL problem with imprecise line correspondences. In spite of the fact that PnL solution in Přibyl et al. (2015) was very robust to outliers, this method needed at least nine line correspondences, whereas the algorithm in Xu et al. (2017) was based on both line and points. Fischler and Bolles (1987) proposed a RANSAC paradigm to solve a PnP problem for given correspondences in cartography purposes. This method rejected the outliers by determining the pose of the camera.

In addition to the previously mentioned cases, a majority of image to model registration approaches leveraged both correspondences establishment and camera pose estimation (David et al. 2003, 2004; Diaz and Abderrahim 2007; Moreno-Noguer et al. 2008; Brown et al. 2015; Rossi et al. 2005; Xia et al. 2012; Karsch et al. 2014; Han and Golparvar-Fard 2015b). David et al. (2004) combined the correct point correspondences (Gold et al. 1995) and the pose

estimation proposed in Dementhon and Davis (1995). A local search was conducted that converged to a minimal solution for an initial guess of an object pose. To provide a globally optimal solution with no prior pose estimation information, a random start scheme was proposed that had the disadvantage of a significant increase in the search space. David et al. (2003) modified this method using line features that needed fewer initial guesses in the random start approach compared to the use of point features. Diaz and Abderrahim (2007) modified the initial algorithm and addressed the 3D object tracking requirements. Other relevant studies tried to reduce the search space using wide variety of strategies from branch and bound (Brown et al. 2015), Gaussian mixture models (Moreno-Noguer et al. 2008), and evolutionary algorithms (Rossi et al. 2005) to differential evolution (Xia et al. 2012). Prior pose information made these methods more robust to clutter, occlusion, and repetitive patterns (Moreno-Noguer et al. 2008).

Han and Golparvar-Fard (2015b) and Karsch et al. (2014) initiated the structure-from-motion (SfM) procedure leveraging BIM as a priori. By interactively guiding BIM into a few images that had significant overlap with the rest of the images, the BIM overlaid on the remaining site images. The primary constraints with this method are the necessity of complete BIM data, which is typically only available for commercial construction sites, high computation time, which prevents the method to be applicable in real-time, and being prone to failure in cases where an anchor camera (i.e., an image that has significant overlap with majority of the images) is chosen inappropriately.

Kropp et al. (2018) conducted a study on the image to 4D BIM registration using camera pose discovery for each image frame with respect to the BIM coordinate system. This method used line segments as the features. Although the initial registration needed manual intervention, the consecutive images were registered in an automatic manner. A SfM algorithm was used for the rough camera motion estimation, which was necessary for the main registration pipeline. For this reason, the whole registration pipeline was not applicable in real time. During the trajectory reconstruction of the video sequence, the scale varied, especially during rotation motion, which could lead to bad results. These limitations are addressed in this paper by proposing an augmented monocular SLAM algorithm.

## Technical Challenges and Objectives

The continuous development of visual SLAM enables real-time estimation of locations and orientations of a camera while incrementally reconstructing a 3D scene. However, visual SLAM localizes a camera to an arbitrary local coordinate system and produces a low-resolution and noisy point cloud that is not usable for comparison of as-built point cloud with as-planned building model. Additionally, the SLAM trajectory (i.e., camera poses in a sequence) is not accurate and suffers from scaling and drift issues, especially in indoor scenes with lack of light and features (e.g., concrete wall and painted wall).

The proposed method improves SLAM trajectory while registering a camera to a BIM. Currently, available methods do not offer continuous and automated registration in real time. Moreover, the majority of the existing methods deal with outdoor construction environments. There is a need for investigating indoor localization and registration in real time. Therefore, the proposed method addresses the following primary research objectives:

- Improving camera localization by minimizing the SLAM drift before loop closure using perspective detection and matching;
- Developing a new SLAM method for robust indoor performance; and
- Performing SLAM and automatic registration of image-to-BIM in real time.

Regarding the first objective, improving the camera poses and minimizing the SLAM drift are implemented at the end of the localization process. In this process, the perspective of the keyframes and their corresponding BIM views are first detected and then matched to improve the estimated camera poses from SLAM. At the end, all the improved camera poses are overwritten to the estimated poses from SLAM, similar to the global bundle adjustment approach with a loop closure.

The augmented SLAM that has been mentioned in the second objective is used for camera pose estimation for each keyframe in the first objective. It has robust performance even in indoor environments with few salient features by using a custom bag-of-word vocabulary (to be further detailed in the "Method" section). The augmented SLAM generates a global map at the first data collection and is restricted to relocalize in the same global map with the same scale in further data collections. This ability to relocalize into the global map enables the augmented SLAM to generate point clouds and odometry in the real-world units in every run.

The third objective is the primary contribution to civil engineering, which enables automatic registration of as-build images to as-planned BIM in real time. By giving visual representation of as-built models aligned with BIM, the presented method will serve as an enabling factor for fully automating past studies on vision-based progress inference, such as Golparvar-Fard et al. (2009), Han and Golparvar-Fard (2015a), and Han et al. (2018). Moreover, it will facilitate communication on jobsites by associating quality and progress with visuals that are aligned with BIM, presenting as-built versus as-planned conditions and supporting decision making in a timely manner.

## Method

This section provides an overview of the proposed method from data acquisition to fine camera pose estimation (Fig. 2). The following subsections correspond to the four steps in Fig. 2.

### *Initial Setup*

The initial setup consists of three steps: building model creation, camera calibration, and as-built data acquisition. The primary geometry information from a BIM such as visible lines in the model is retrieved as an input. MATLAB camera calibration (Fetić et al. 2012) is applied to obtain camera intrinsic parameters, such as a focal length, coordinates of the optic center, and skew parameter. These parameters are used for 3D reconstruction and camera pose estimation within the BIM. As-built data at a given time can be collected using a camera while performing the proposed SLAM for the localization and mapping process during the inspection.

### *Initial Frame Registration and Camera Localization*

The initial frame registration delivers an accurate camera pose that corresponds to the first frame of an image sequence. The proposed method is for a periodic monitoring system and assumes that the first frame to be registered. This is the only manual step. Afterward, the camera poses of the consecutive monitoring are recovered automatically in real time.

An augmented monocular SLAM algorithm that builds on ORB-SLAM2 (Mur-Artal and Tardós 2016) is proposed for camera localization and mapping. To enhance the performance of SLAM in

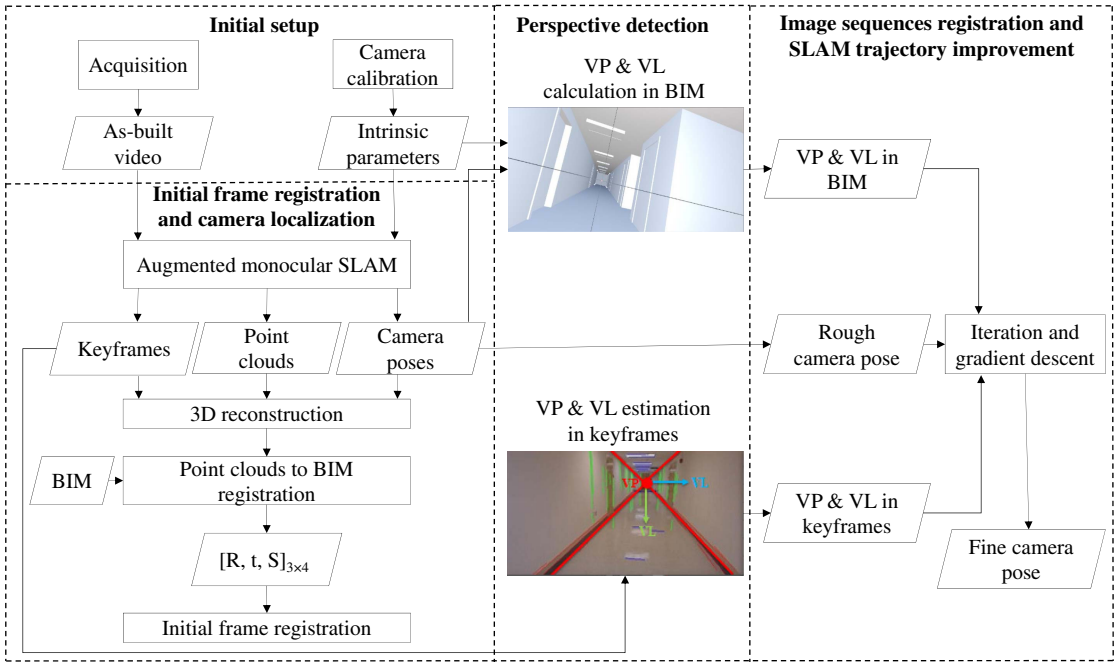


**Fig. 2.** Concept overview for image sequences to BIM registration and SLAM-derived camera trajectory improvement.

indoor environments even before a loop closure, a custom bag-of-words vocabulary is generated from the video using the DBow2 library (Gálvez-López and Tardós 2012). The custom vocabulary improves the tracking performance of SLAM. In the first data collection for each new environment (e.g., hallway and construction site in this paper), SLAM localizes the camera to an arbitrary local coordinate system and creates the map of the environment with a sparse point cloud. To localize the camera with respect to the BIM coordinate system, a transformation matrix between the point cloud and BIM is needed. For this purpose, a dense point cloud is generated using multiview environment (MVE) technique (Fuhrmann et al. 2014). After manually selecting corresponding features (i.e., corners) between the point cloud and BIM, a similarity transformation is performed to align them (Han et al. 2018; Han and Golparvar-Fard 2015b). The aligned model serves as a global map in the real-world scale. In this map, the scale factor is serialized to disk as a binary file using the serialization feature of the Boost C++ libraries (Ramey 2004). This binary file is then loaded later for further data collection. Finally, the transformation matrix including rotation, translation, and scale ($[R,t,s]_{3\times4}$) is applied to the first camera pose from SLAM and generates the BIM view corresponding to the first video keyframe [selected image frames during the localization and mapping process (Mur-Artal et al. 2015)].

These steps happen once during the first data collection as a preprocessing step. In the further data collections, SLAM is restricted to relocalize within the global map using the saved scale factor from the binary file. As an added advantage, when relocalizing, the local mapping thread does not run, leading to faster processing times on the same hardware. A video (Asadi 2018b) is prepared to illustrate the process of a global map generation during the first data collection, followed by SLAM relocalization in the same map with the same scale for further data collections. Note that if the system is in mapping mode, after relocalization in the global map, the SLAM algorithm can create new keyframes even if it is revisiting some places that have been already mapped. As it is shown in the video, more points are also added to the global map, which provides a denser global map after each run.

### *Vanishing Points/Lines Estimation in Video Keyframes*

Vanishing points and vanishing lines (VLs) are considered as correspondences between an as-built scene and its corresponding BIM view. The proposed approach takes images with red, green, and blue color channels (i.e., RGB images ) as input and delivers pixel location of three vanishing points corresponding to three mutually orthogonal directions of a scene and related vanishing lines in each direction, similar to Hedau et al. (2009). In this method, Canny edge detector (Canny 1986) detects edges. Then, the three mutually orthogonal directions are detected through a voting and searching scheme. Edges that are longer than a predefined threshold are considered as lines. Depending on views, the line intersection in each direction forms either finite or infinite vanishing point, independent of the camera direction and orientation. The proposed method estimates a triplet of vanishing points that correspond to the three principal orthogonal directions of the scene. The crossproduct of two vanishing points on a plane provides a vanishing line. Fig. 3 illustrates two perpendicular lines, corresponding to the horizontal and vertical vanishing lines, respectively. The dot in the intersection of the vanishing lines shows the finite vanishing point.

### *Vanishing Points/Lines Calculation on BIM Views*

The 2D pixel coordinate of each 3D point is calculated by perspective projection. During this calculation, three transformations that correspond to the subsequent three equations [adopted from Snavely et al. (2006)] are applied. Eq. (1) is a rigid transformation (rotation + translation) that transfers the world coordinate system (3D model) to the camera coordinate system. In this equation, $P^C$ and $P^W$ are the same physical points, described in camera and BIM coordinate systems, respectively; $t_W^C$ is a translation vector between the camera origin and the origin of the BIM coordinate system; and $R_W^C$ is the $3 \times 3$ rotation matrix of the BIM axis with respect to the camera axis.

The perspective division transfers the camera coordinate system to the 2D image plane [Eq. (2)], where $P_z^C$ is the third $(z)$ coordinate of $P^C$. Eq. (3) is an affine transformation and describes the

**Fig. 3.** Illustration of vanishing point/lines estimation using edge detection in an example frame.

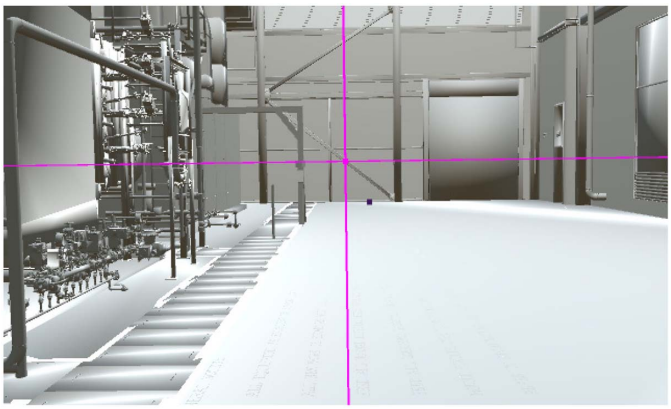

**Fig. 4.** Illustration of vanishing point/lines calculation in an example BIM view.

transformation between the image plane and pixel arrays, where $f$ and $c$ are the focal length and image center vectors, respectively; and $r(p)$ is a function [Eq. (4)] that computes a scaling factor that removes the radial distortion using radial distortion coefficients ($k_1$ and $k_2$)

$$P^C = R_W^C \cdot P^W + t_W^C \tag{1}$$

$$p = -P^C / P_z^C \tag{2}$$

$$p' = f \cdot r(p) \cdot p + c \tag{3}$$

$$r(p) = 1 + k_1 \cdot \|p\|^2 + k_2 \cdot \|p\|^4 \tag{4}$$

To transform the BIM to the 2D view of the camera, the intrinsic and extrinsic camera parameters (i.e., camera properties, location, and orientation) are used to bring the BIM into the new position and its image plane. For instance, the method starts with the known first frame for both the camera and BIM. Then for the second frame, SLAM returns a new camera pose (i.e., extrinsic parameters). This new pose is used to coarsely move the BIM to the new camera location (referred to as a rough camera pose in this paper). Then the three previous equations are applied to retrieve the 2D view from this location. From this 2D BIM view, parallel lines in the same direction intersect at the vanishing point in that direction. The vanishing lines' equations are calculated similar to the previous step (see the horizontal and vertical vanishing lines in Fig. 4).

### *Image Sequence Registration and SLAM Trajectory Improvement*

In this step, the perspective information (vanishing points/lines) extracted from the keyframes and their corresponding BIM views are aligned. This alignment improves the estimated camera pose from SLAM (rough camera pose) and recovers a fine camera pose. The camera location and orientation for all the keyframes are improved during a gradient descent–based iterative process as illustrated by Fig. 5 and further detailed subsequently. Gradient descent is a first-order iterative optimization algorithm that finds the values of a function's parameters (in this case, location and orientation of the camera) to find the minimum of a cost function (in this case, perspective alignment error functions).

Perspective alignment has two types of errors between the image frame and its corresponding BIM view. Eq. (5) calculates a mean square error (or distance error) in pixels between the finite vanishing points in a keyframe ($X_{keyframe}$, $Y_{keyframe}$) and its corresponding BIM view ($X_{\text{BIM}}$, $Y_{\text{BIM}}$). Eq. (6) shows an angular error in degree, which represents the angle between the slope of $VL_{keyframe}$ (e.g., the slope of a horizontal vanishing line) on a keyframe and its corresponding slope of $VL_{\text{BIM}}$ on BIM view (Fig. 5)

$$\Delta d = \sqrt{(X_{\text{BIM}} - X_{keyframe})^2 + (Y_{\text{BIM}} - Y_{keyframe})^2} \tag{5}$$

$$\Delta\theta = \tan^{-1}\left(\frac{VL_{\text{BIM}} - VL_{keyframe}}{1 + VL_{\text{BIM}} \cdot VL_{keyframe}}\right) \tag{6}$$

The iterative process returns a fine camera pose with respect to the BIM. The gradient descent–based algorithm is implemented to reduce the aforementioned errors after each iteration (Fig. 5). To process this iteration in real time, a maximum number of iterations ($\eta$) and two thresholds for distance and angular errors ($\delta$ and $\epsilon$, respectively) are defined. Fig. 6 demonstrates the process of fine pose estimation for each arriving keyframe with respect to the BIM. The gradient decent stops when either the errors are below the threshold or the maximum number of iterations is reached. Then, the fine camera pose with the minimized errors is returned.

The reason behind processing only the keyframes comes from the purpose of registration. Processing only the keyframes instead of all the frames has the advantage of preventing unnecessary processing for registering almost similar as-built data to the as-planned model. For example, in the case of static camera without any movement, the camera captures many similar images per second; however, there is no need for registering all these similar frames to the BIM. By changing the camera view a new keyframe is generated and the proposed method enables the new keyframe to be registered to the BIM in real time.

## Experimental Setup and Results

### *Initial Setup*

The proposed method is validated in two different indoor scenes: a hallway with featureless walls, and a construction site. A BIM of the hallway was modeled and a BIM of the construction site was given by a construction company. A unmanned ground vehicle (UGV) with a monocular camera (i.e., a webcam) from a previous study (Asadi et al. 2018d) was used for data collection. This UGV moved in a path while recording 30 frames per second (fps) video with 1,920 × 1,080 resolution. The length of the videos for the hallway and the constriction site are 70 s (2,100 frames including

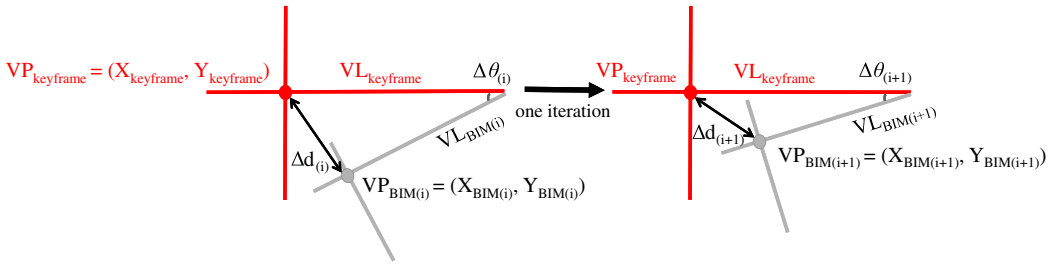


**Fig. 5.** Iterative process for optimizing the distance error ($\Delta d$) between vanishing points ($VP_{keyframe}$ and $VP_{\mathrm{BIM}}$) and the angular error ($\Delta\theta$) between vanishing lines with the slopes of $VL_{keyframe}$ and $VL_{\mathrm{BIM}}$.

**Input:** $IM_{keyframe} \in Video$: Key video frames of SLAM
  $[K]$: Camera intrinsic matrix from camera calibration
  $[R|t]_{keyframe}$: Rough camera pose of $IM_{keyframe}$
  $\delta$ and $\epsilon$: Thresholds for distance and angular errors, respectively
  $\eta$: Max number of iterations
**Output:** Fine camera pose per $IM_{keyframe}$ in the BIM coordinate system
**foreach** $IM_{keyframe} \in Video$ **do**
  $IM_{BIM}$ = Project BIM view using $[K]$ and $[R|t]_{keyframe}$
  $(VP_{keyframe}, VL_{keyframe})$ = Perform perspective detection on $IM_{keyframe}$
  $(VP_{BIM}, VL_{BIM})$ = Perform perspective detection on $IM_{BIM}$
  $\Delta d_{keyframe} = VP_{keyframe} - VP_{BIM}$: Distance error
  $\Delta\theta_{keyframe} = tan^{-1}[(VL_{BIM} - VL_{keyframe})/(1 + VL_{BIM} \cdot VL_{keyframe})]$ : Angular error
  **while** $iter<\eta$ *or* $(\Delta d > \delta$ *and* $\Delta\theta > \epsilon)$ **do**
    $([R|t]_{iter}, \Delta d_{iter}, \Delta\theta_{iter})$ = Perform gradient descent to estimate a new pose
    $iter++$
  **end**
  $[R|t]_{iter}$: Return fine camera pose
**end**

**Fig. 6.** Fine pose estimation based on perspective alignment.

52 keyframes) and 120 s (3,600 frames including 59 keyframes), respectively. The camera had a fixed focal length. The intrinsic parameters are determined through camera calibration as previously mentioned in the "Method" section.

### Initial Frame Registration and Camera Localization

The proposed SLAM is performed by an NVIDIA Jetson TX1 (NVIDIA 2017) on the UGV, which localizes the UGV with respect to the BIM coordinate system. To evaluate the performance of the proposed SLAM, ORB-SLAM2 and the proposed SLAM are compared by generating and comparing the camera trajectories and 3D reconstructed scenes of a hallway (Fig. 7). This hallway has featureless walls and repetitive features (i.e., doors, ceiling tiles, light fixtures, etc.). ORB-SLAM2 was unable to detect two 90° turns accurately. Instead, ORB-SLAM2 created two different hallways [Fig. 7(a)]. However, there was only one hallway as shown in Fig. 7(c). Fig. 7(c) presents an improved trajectory before a loop closure by the proposed SLAM.

Creating a global map of such scenes with ORB-SLAM2 results in high error for point clouds and estimated camera poses. The unacceptable point cloud and camera trajectory makes the process of recovering the initial corresponding 2D view challenging. As it is shown in Fig. 7(c), the augmented SLAM with custom vocabulary, shows improved tracking performance. In the further data collection SLAM is restricted to run in this accurate global map and the initial corresponding 2D view is recovered automatically and accurately.

The first keyframes of the video sequences taken from the hallway and the construction site are shown in Fig. 8(a). Fig. 8(b) presents the SLAM trajectory and point clouds generation. Fig. 8(c) shows the dense point clouds that are improved using the MVE method, which can be used for semiautomated registration (Han et al. 2018). The transformation matrix from the latter step is applied to the first camera pose from SLAM and generates the BIM view corresponding to the first video keyframe [Fig. 8(d)].

### Vanishing Points/Lines Estimation in Video Keyframes

To evaluate real-time performance and accuracy, keyframes are downsampled to two different image resolutions. Two sets of video keyframes with 640 × 360 and 1,200 × 673 pixels are tested. All the processes are performed using an NVIDIA Jetson TX1 (NVIDIA 2017) as a processing unit.

The perspective estimation process time for each video keyframe varies by the image resolution and the number of lines that need to be extracted from that keyframe. The average process times for keyframes with 640 × 360 resolution from the hallway and the construction site are about 0.5 and 1.5 s per keyframe, respectively. These values for the keyframes with 1,600 × 675 resolution from the hallway and the construction site are about 1,600 and 2,500 s per keyframe, respectively. Although the estimation process for the images with a higher resolution is more accurate, the need for real-time performance restricts the image resolution to be 640 × 360.

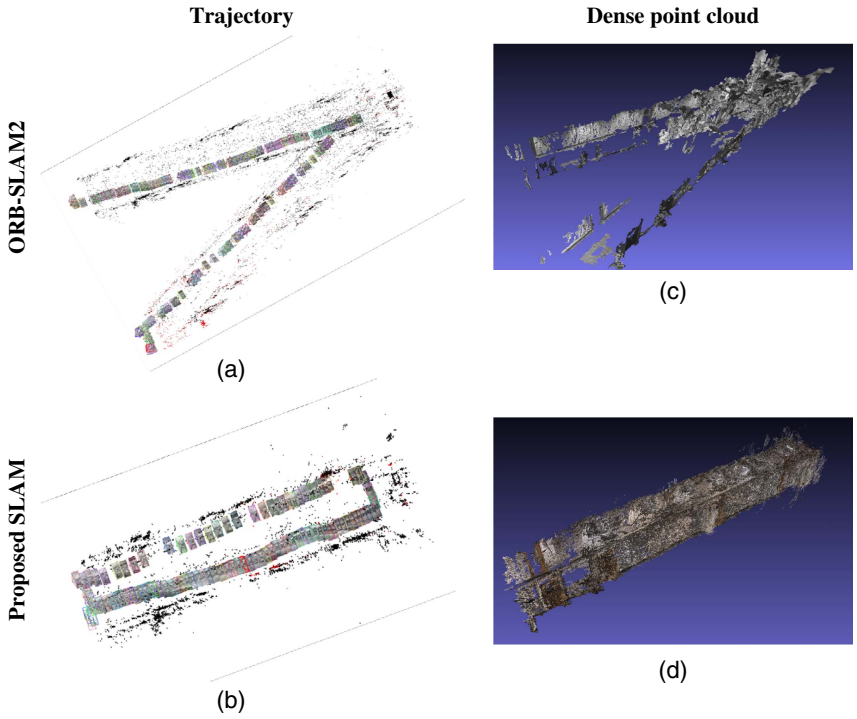


**Fig. 7.** (a and b) Camera trajectory; and (c and d) scene 3D reconstruction of a simple hallway with repetitive features using (a and c) ORB-SLAM2 and (b and d) the proposed SLAM.

### *Vanishing Points/Lines Calculation in BIM*

The vanishing points' pixel location and vanishing lines' equation in the 2D BIM views are calculated and visualized. Fig. 11(c) shows two perpendicular lines that depict the horizontal and vertical vanishing lines. The dot at the intersection of the vanishing lines shows the finite vanishing point. Because the 3D information (or 2D coordinates after projection) is available from the BIM, vanishing points/lines can be directly computed without detecting lines.

### *Image Sequence Registration and SLAM Trajectory Improvement*

In this section, the accuracy and robustness of the improved SLAM are evaluated. Therefore, the deviations of the estimated camera poses from SLAM are defined by calculating the perspective alignment errors as described in the section "Image Sequence Registration and SLAM Trajectory Improvement." The iteration parameters (i.e., maximum number of iteration and thresholds for distance and angular errors) are defined as follows: $\eta = 500$, $\delta = 1$ pixel, and $\epsilon = 1°$. The errors after the iteration are significantly reduced, returning fine camera poses. Figs. 9 and 10 show the perspective alignment errors (distance errors and angular errors, respectively) that correspond to each keyframe before and after iteration. The average distance error before iteration for the keyframes from the construction site is much higher than the keyframes from the hallway (see the triangles in Fig. 9). According to Fig. 8(b), the camera trajectory in the hallway scene is almost straight; however, the camera trajectory in the construction site is almost circular at the beginning (up to half of the path), followed by a straight movement for the remaining path. The first 28 camera poses of the keyframes had high rotation motion, which is associated with a high drift issue (inherent limitation of SLAM) and consequently high distance errors for these keyframes. Remaining camera poses had a straight movement with substantially lower distance errors as can be seen in Fig. 9. Angular errors in keyframes from both scenes are very low (less than 3°) even before the iteration, which are reduced to less than 0.01° after the iteration.

Fig. 11 shows the registration of six keyframes and their corresponding BIM views. Three keyframes are from the hallway and the other three are from the construction site. Fig. 11(a) shows six keyframes. Fig. 11(b) presents the estimated vanishing points/lines. Fig. 11(c) presents the corresponding BIM views using the rough camera poses (derived from SLAM) and the vanishing points/lines that are calculated in each BIM view. Fig. 11(d) presents the fine camera poses that are recovered after perspective matching through the iterative process.

A video (Asadi 2018a) is prepared to show the robustness of the proposed method. This video shows the registration of all the keyframes from the construction site scene (59 keyframes) to the BIM. The keyframe sequences and their corresponding and recovered BIM views are displayed in the left and right windows, respectively.

### *Computation Time*

This section further explains the average computation time required for processing each keyframe of the hallway and the construction site in $640 \times 360$ resolution. The computation time represents the processing time for estimating a rough camera pose using SLAM, estimating VP/VL for the keyframe, and recovering a fine camera pose through the iterative perspective alignment process. The latter process includes VP/VL calculation in the corresponding BIM view during the iteration.

The number of features in each frame is a factor that affects the performance of the system greatly. More features mean more time spent on feature extraction and a large number of matches to compute during the iteration process before returning a rough camera pose. As can be seen in Fig. 12, the average computation time for each keyframe from the hallway and the construction site are 657

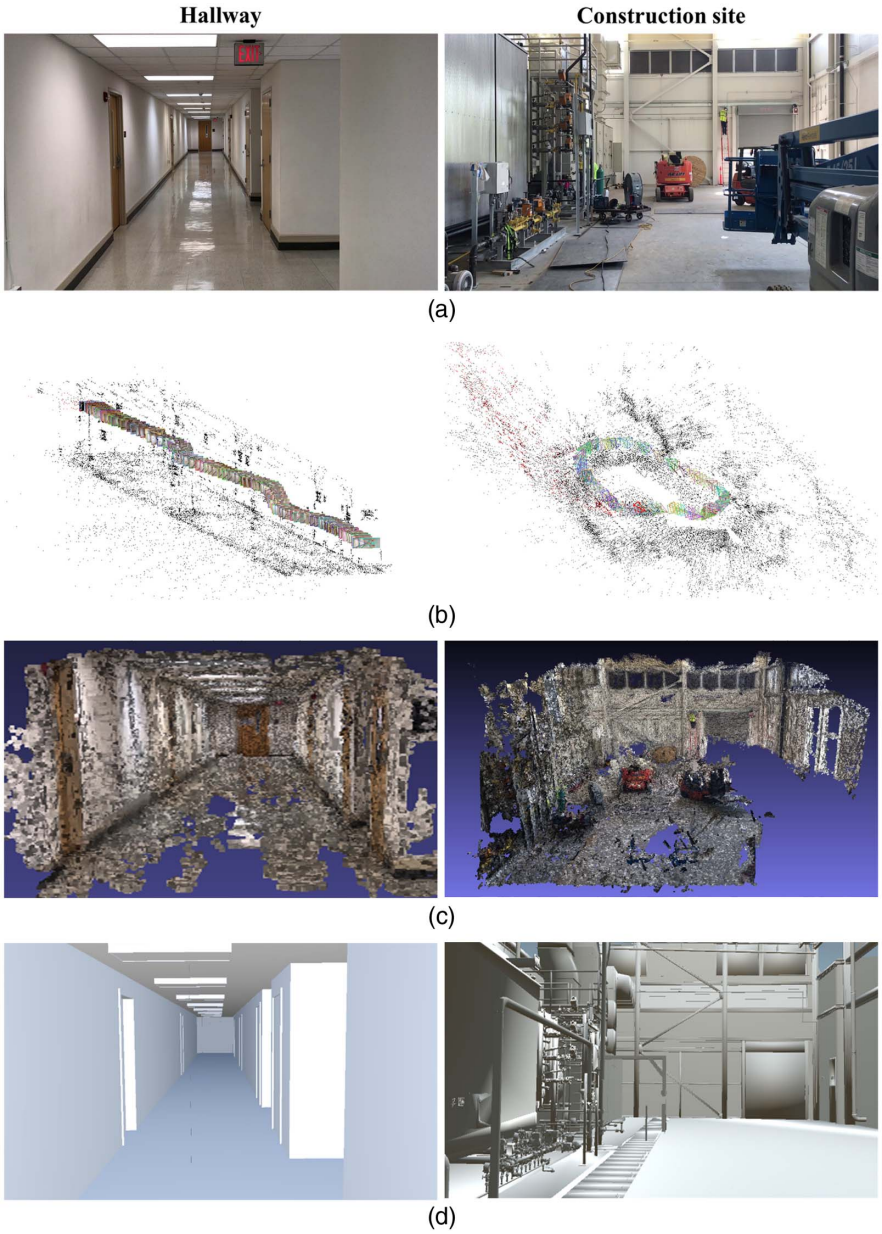


**Fig. 8.** First keyframe registration: (a) first keyframes; (b) trajectory and sparse point clouds from SLAM; (c) dense point clouds after 3D dense reconstruction; and (d) recovered BIM views corresponding to the first keyframes.

and 1,904 ms, respectively. The computation time for the first two processes (estimating rough camera pose and VP/VL) are higher for the keyframes from the construction site. The reason is that these keyframes have more features and lines, requiring more time to process each keyframe. As shown in Fig. 9, the average of the distance errors before the iteration is higher for the keyframes from the construction site. The higher errors, in this case, led to high numbers of iterations, often equals to the chosen threshold $\eta$ when recovering the fine camera pose.

## Discussion

### *Impact of Perspective Estimation Accuracy*

During the processing of the video keyframes, it could be observed that perspective estimation in some keyframes from the cluttered scene and with $640 \times 360$ resolution can lead to bad results (inaccurately recovered camera poses). Because the proposed method uses perspective detection and matching for recovering the corresponding BIM view for each keyframe, accurate estimation of VP/VL for the keyframes is vital for this method and counts as a critical assumption. The accuracy of the perspective estimation process largely depends on the quality of the extracted low-level image features, which is susceptible to scene clutters, occlusion, and local noise. The findings indicate that the perspective estimation method is unable to detect accurately lines in the directions of VP/VL, which makes the VP/VL estimation process error prone.

Fig. 13 shows the mean square error in pixels between the estimated VP and the ground truth from the hallway (squares) and the construction site (triangles), with $640 \times 360$ resolution. These errors for the hallway scene are negligible. However, about 15 keyframes from the construction site (a cluttered scene) have high errors. Performing perspective estimation of the keyframes with

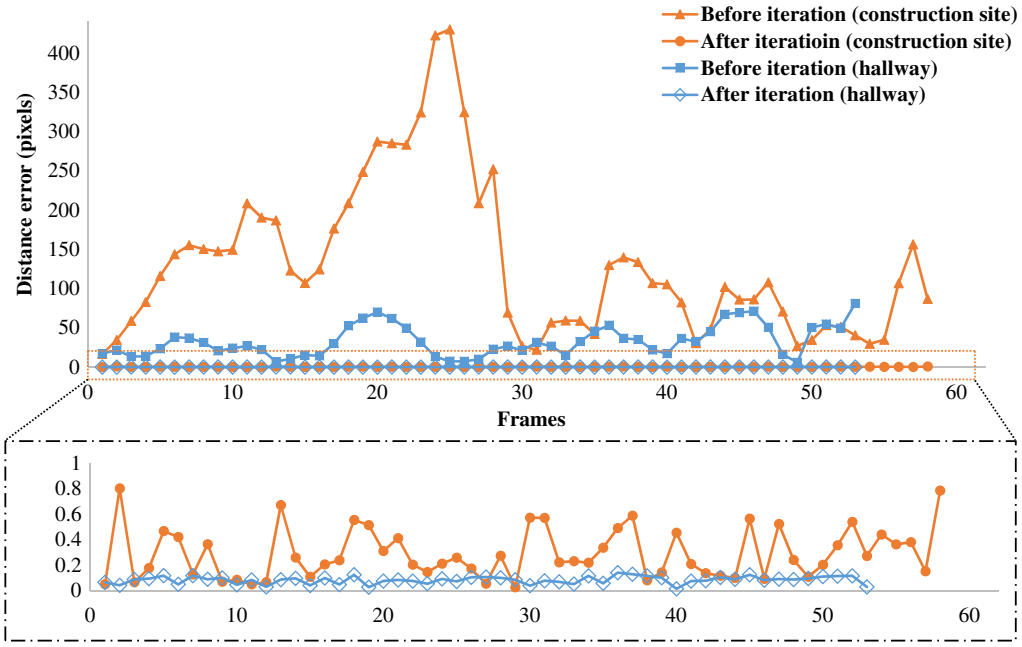


**Fig. 9.** Distance errors before and after the iteration process corresponding to the keyframes from the hallway and construction site scenes. Errors after iteration have been magnified for better visualization.

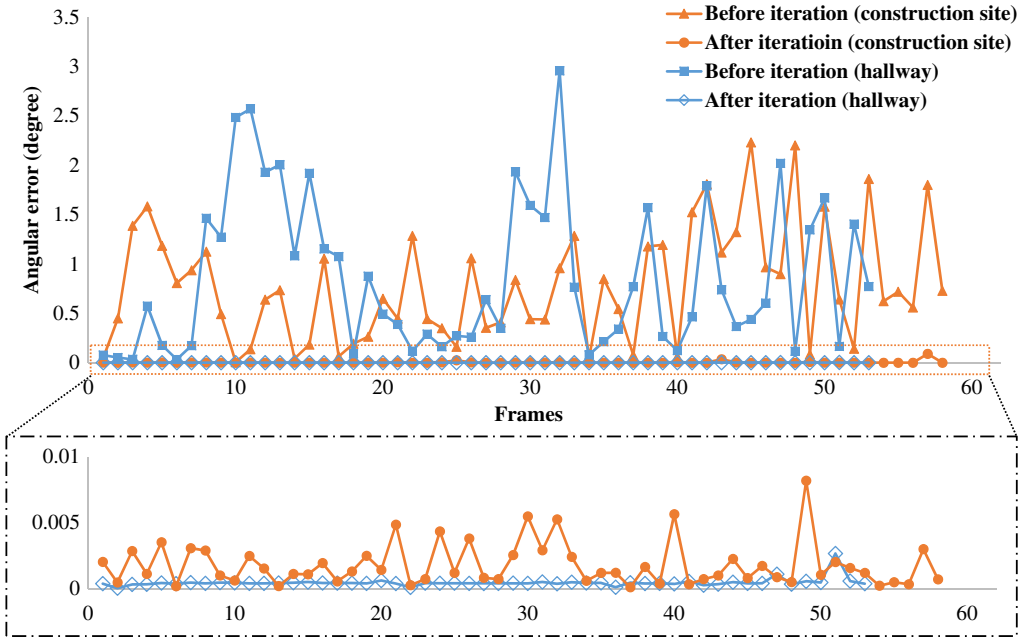


**Fig. 10.** Angular errors before and after the iteration process corresponding to the keyframes from the hallway and construction site scenes. Errors after iteration have been magnified for better visualization.

1,200 × 675 resolution yielded much higher accuracy (almost equal to the ground truth) (Fig. 14). However, as previously reported, processing images with 1,200 × 675 resolution takes an average of 2,500 s per keyframe (see "Vanishing Points/Lines Estimation in Video Keyframes" subsection under "Experimental Setup and Results" section).

In computer science, a real-time computing (RTC) system is one that guarantees a response before a previously set deadline (Furht et al. 1991; Stankovic 1988; Krishna 1999). For a stream of images, a real-time system is one that guarantees it has finished processing each image before some previously set timing constraint on latency. Regarding real-time performance for the current experiment, the average computation time for processing each keyframe and recovering its corresponding camera pose was reported in the preceding section. A 70-s video from the hallway with 52 keyframes out of 2,100 frames (0.74 keyframes per second) had the average computation time of 0.65 s per keyframe. Because the processing time for each keyframe is lower than the number of keyframes per second, the proposed registration method ran in real-time for the hallway.

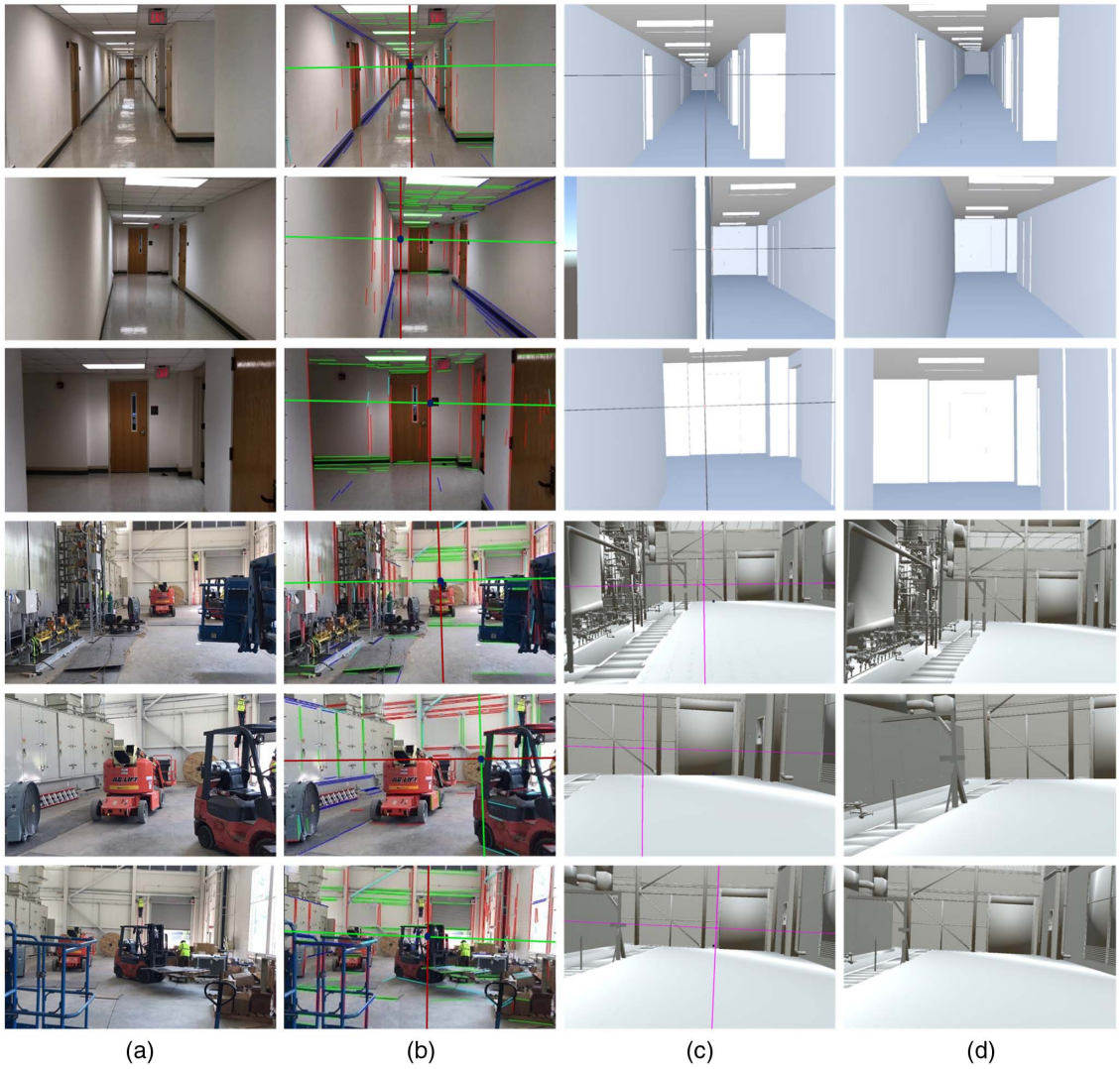


**Fig. 11.** Example images of (a) video keyframes; (b) estimating VP/VL in the video keyframes; (c) calculating VP/VL in the initial 2D BIM views from SLAM (inaccurate and not usable for as-built/as-planned comparison); and (d) recovering fine camera poses using perspective matching through the iterative process.

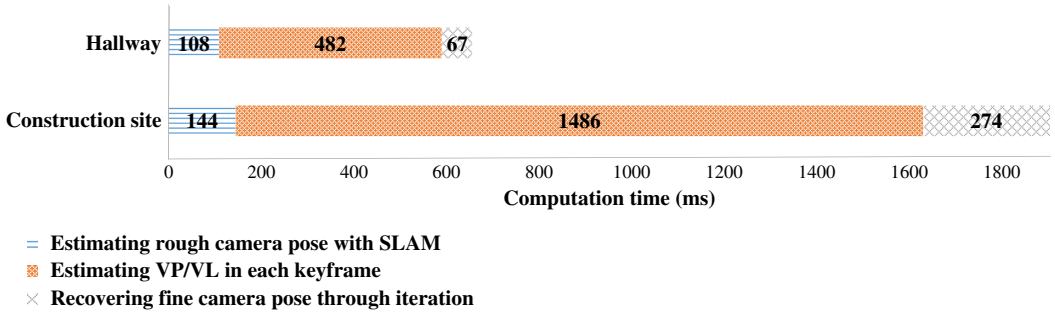


**Fig. 12.** Average computation time required for processing each keyframe from the hallway and the construction site.

A 120-s video from the construction site with 59 keyframes out of 3,600 frames (0.5 keyframes per second) had the average computation time of 1.9 s per keyframe using a Jetson TX1 on the UGV. Because the process time for each keyframe is higher than the number of keyframes per second, the UGV had to move at a much slower speed in order to maintain enough overlaps between video frames to localize and register in real time.

These findings from the hallway and construction site indicate that the speed of a UV is not a constraint in a scene that is not cluttered (i.e., a hallway with a less number of features). However,

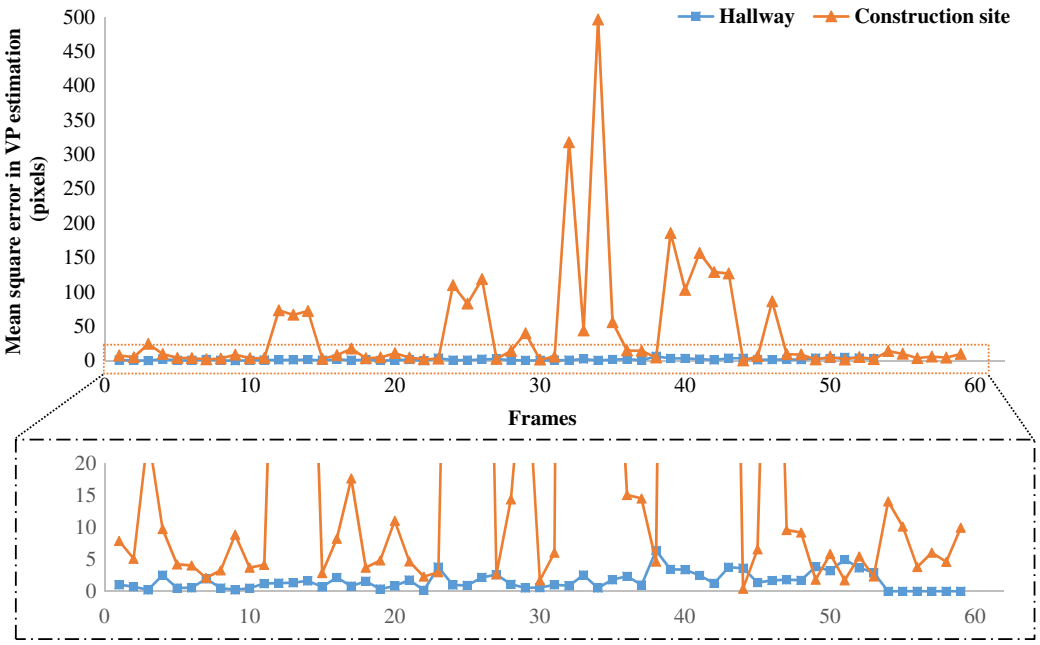


**Fig. 13.** Mean square error in VP estimation corresponding to the video keyframes with the resolution of 640 × 360 pixels from the hallway scene (squares) and the construction site (triangles). Small errors have been magnified for better visualization.

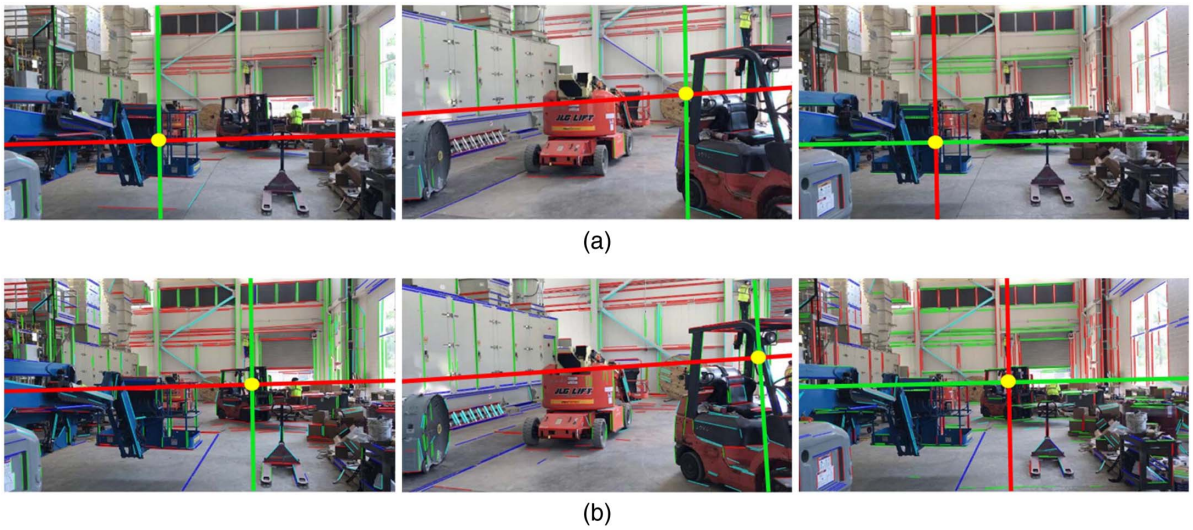


**Fig. 14.** (a) Examples of inaccurate perspective estimation in images with the resolution of 640 × 360; and (b) accurate perspective estimation in images with the resolution of 1,200 × 673.

in a cluttered scene (i.e., a construction site with many objects), the speed of a UV can be a constraint as VP/VL estimation is a bottleneck as can be seen in Fig. 12.

Another way to ensure the real-time operation is the use of a more powerful computer. For instance, a desktop with an Intel i7 processor (3.4 GHz with 6 cores) can process a keyframe from the construction site in less than 0.2 s.

### *Practical Implications*

The bottleneck of past studies on vision-based as-built and as-planned data comparison [e.g., Golparvar-Fard et al. (2009), Han and Golparvar-Fard (2015a), and Han et al. (2018)] in automating their monitoring framework was data collection and registration with BIM. Therefore, this study will serve as an enabling factor for fully automating these methods by providing an automated and accurate registration. Fig. 15 shows two examples of registered keyframes to their corresponding BIM views. For better visualization, a Canny edge detector (Canny 1986) is applied to the BIM views to highlight edges [Figs. 15(c and d)]. While most of the back-projected images aligned accurately with their corresponding BIM views [Figs. 15(a and c)], there were some instances of minor misalignments [see lines in Figs. 15(b and d)]. Figs. 15(b and d) shows the distance error of 18 pixels (Fig. 13) in 17th keyframe form the construction site video.

Even with these minor misalignments, construction project participants can still discuss construction performance (especially progress) and any issues found on these images to make decisions.

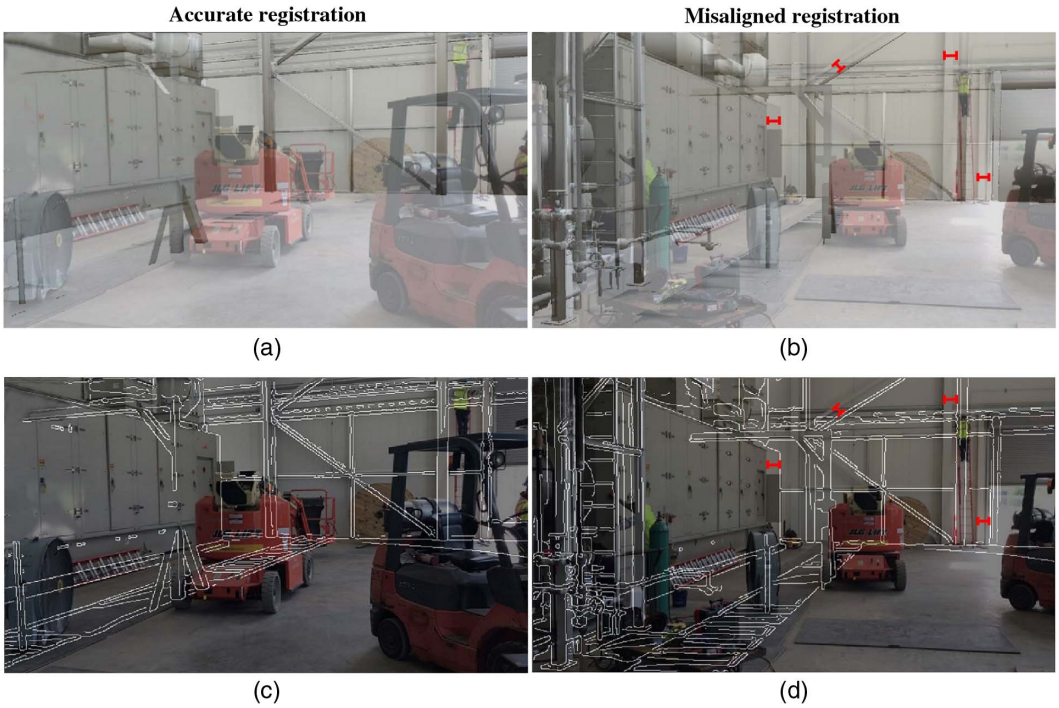


**Fig. 15.** Examples of accurate and misaligned image-to-BIM registration: (a) accurate registered image to its BIM view; (b) registered image to its BIM view with minor misalignment; and (c and d) BIM elements shown as edges for better visualization. Lines on (b and d) indicate misalignments.

The proposed method can potentially serve as a decision-making tool, similar to Reconstruct Inc. (Reconstruct 2018) and Navisworks (Autodesk 2018) that can compare as-built and as-planned conditions. For further data analytics (i.e., quality assessment), denser point clouds and more accurate registration through postprocessing are necessary. Even with postprocessing, the proposed method reduces the overall computational time through its real-time registration that produces initial alignment.

## Conclusion and Future Research

This paper presents a novel method that increases the degree of automation in construction monitoring applications. Each keyframe of a video sequence is registered to the BIM in real time by aligning the perspective of the keyframe with its corresponding BIM view. This process provides the BIM views in the views of keyframes. The proposed SLAM is used for estimating a rough camera pose information for each keyframe. These poses are improved during the gradient descent-based iteration process. The two case studies present promising results, demonstrating accurate registration and real-time performance in indoor environments.

The proposed method is based on perspective detection that estimates vanishing lines and points by detecting straight edges on images. Therefore, a scene with curved walls or arches is prone to higher error compared to a scene with straight edges. Addressing this limitation will allow applying the proposed method to buildings with unique shapes.

Moreover, after the rough camera pose estimation using the output of SLAM, the fine camera pose estimation using perspective detection improves the camera location in two axes that are on the image frame [i.e., lines in Fig. 11(b)]. However, the fine camera pose estimation is not able to improve camera poses in the direction that is normal to the frame, only relying on the SLAM output in this direction. This limitation can be addressed by driving a UV with a fixed velocity (SLAM yields less drift in this direction with the fixed velocity). Another way to address this limitation is the use of a secondary positional sensor [i.e., an inertial measuring unit (IMU) and ultrawide band (UWB)] as a complimentary localization method.

## References

Agarwal, S., Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. 2011. "Building Rome in a day." *Commun. ACM* 54 (10): 105–112. https://doi.org/10.1145/2001269.2001293.

Ansar, A., and K. Daniilidis. 2003. "Linear pose estimation from points or lines." *IEEE Trans. Pattern Anal. Mach. Intell.* 25 (5): 578–589. https://doi.org/10.1109/TPAMI.2003.1195992.

Asadi, K. 2018a. "Real-time image localization and registration with BIM using perspective alignment." Accessed October 29, 2018. https://youtu.be/sQFZNDo6zPk.

Asadi, K. 2018b. "Restricting SLAM to re-localize in a global map with the same scale." Accessed October 29, 2018. https://youtu.be/SqbofWqoWXQ.

Asadi, K., and K. Han. 2017. "Perspective-based image-to-BIM alignment for automated visual data collection and construction performance monitoring." In *Proc., Int. Workshop on Computing in Civil Engineering 2017*, 171–178. Reston, VA: ASCE.

Asadi, K., and K. Han. 2018. "Real-time image-to-BIM registration using perspective alignment for automated construction monitoring." In *Proc., Construction Research Congress 2018*, 388–397. Reston, VA: ASCE.

Asadi, K., H. Ramshankar, H. Pullagurla, A. Bhandare, S. Shanbhag, P. Mehta, S. Kundu, K. Han, E. Lobaton, and T. Wu. 2018c. "Building an integrated mobile robotic system for real-time applications in construction." Preprint, submitted April 18, 2018. http://arxiv.org/abs/1803.01745.

Asadi, K., H. Ramshankar, H. Pullagurla, A. Bhandare, S. Shanbhag, P. Mehta, S. Kundu, K. Han, E. Lobaton, and T. Wu. 2018d. "Vision-based integrated mobile robotic system for real-time applications in construction." *Autom. Constr.* 96 (Dec): 470–482. https://doi.org/10.1016/j.autcon.2018.10.009.

Autodesk. 2018. "Navisworks: Project review software for AEC professionals." Accessed December 11, 2018. https://www.autodesk.com/products/navisworks/overview.

Bellekens, B., V. Spruyt, R. Berkvens, and M. Weyn. 2014. "A survey of rigid 3D point cloud registration algorithms." In *Proc., AMBIENT 2014: The 4th Int. Conf. on Ambient Computing, Applications, Services and Technologies*, 8–13. Wilmington, DE: IARIA XPS Press.

Bohn, J. S., and J. Teizer. 2010. "Benefits and barriers of construction project monitoring using high-resolution automated cameras." *J. Constr. Eng. Manage.* 136 (6): 632–640. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000164.

Bosche, F., C. T. Haas, and B. Akinci. 2009. "Automated recognition of 3D CAD objects in site laser scans for project 3D status visualization and performance control." *J. Comput. Civ. Eng.* 23 (6): 311–318. https://doi.org/10.1061/(ASCE)0887-3801(2009)23:6(311).

Bosché, F. 2010. "Automated recognition of 3D CAD model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction." *Adv. Eng. Inf.* 24 (1): 107–118. https://doi.org/10.1016/j.aei.2009.08.006.

Bosché, F. 2012. "Plane-based registration of construction laser scans with 3D/4D building models." *Adv. Eng. Inf.* 26 (1): 90–102. https://doi.org/10.1016/j.aei.2011.08.009.

Bosché, F., M. Ahmed, Y. Turkan, C. T. Haas, and R. Haas. 2015. "The value of integrating scan-to-BIM and scan-vs-BIM techniques for construction monitoring using laser scanning and BIM: The case of cylindrical MEP components." *Autom. Constr.* 49 (Jan): 201–213. https://doi.org/10.1016/j.autcon.2014.05.014.

Bresson, G., T. Féraud, R. Aufrère, P. Checchin, and R. Chapuis. 2015. "Real-time monocular SLAM with low memory requirements." *IEEE Trans. Intell. Transp. Syst.* 16 (4): 1827–1839. https://doi.org/10.1109/TITS.2014.2376780.

Brilakis, I., M. W. Park, and G. Jog. 2011. "Automated vision tracking of project related entities." *Adv. Eng. Inf.* 25 (4): 713–724. https://doi.org/10.1016/j.aei.2011.01.003.

Brown, M., D. Windridge, and J.-Y. Guillemaut. 2015. "Globally optimal 2D-3D registration from points or lines without correspondences." In *Proc., IEEE Int. Conf. on Computer Vision*, 2111–2119. New York: IEEE.

Canny, J. 1986. "A computational approach to edge detection." *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-8 (6): 679–698. https://doi.org/10.1109/TPAMI.1986.4767851.

Chen, H. H. 1990. "Pose determination from line-to-plane correspondences: Existence condition and closed-form solutions." In *Proc., 3rd Int. Conf. on Computer Vision, 1990*, 374–378. New York: IEEE.

Concha, A., and Civera, J. 2015. "DPPTAM: Dense piecewise planar tracking and mapping from a monocular sequence." In *Proc., 2015 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 5686–5693. New York: IEEE.

Coughlan, J. M., and A. L. Yuille. 1999. "Manhattan world: Compass direction from a single image by Bayesian inference." In Vol. 2 of *Proc., 7th IEEE Int. Conf. on Computer Vision, 1999*, 941–947. New York: IEEE.

David, P., and D. DeMenthon. 2005. "Object recognition in high clutter images using line features." In Vol. 2 of *Proc., 10th IEEE Int. Conf. on Computer Vision, 2005 ICCV 2005*, 1581–1588. New York: IEEE.

David, P., D. DeMenthon, R. Duraiswami, and H. Samet. 2003. "Simultaneous pose and correspondence determination using line features." In Vol. 2 of *Proc., 2003 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 2003*. New York: IEEE.

David, P., D. DeMenthon, R. Duraiswami, and H. Samet. 2004. "Softposit: Simultaneous pose and correspondence determination." *Int. J. Comput. Vision* 59 (3): 259–284. https://doi.org/10.1023/B:VISI.0000025800.10423.1f.

Dementhon, D. F., and L. S. Davis. 1995. "Model-based object pose in 25 lines of code." *Int. J. Comput. Vision* 15 (1–2): 123–141. https://doi.org/10.1007/BF01450852.

Denis, P., J. H. Elder, and F. J. Estrada. 2008. "Efficient edge-based methods for estimating Manhattan frames in urban imagery." In *Proc., European Conf. on Computer Vision*, 197–210. Dordrecht, Netherlands: Springer.

Diaz, J., and M. Abderrahim. 2007. "Modified Softposit algorithm for 3D visual tracking." In *Proc., 2007 IEEE Int. Symp. on Intelligent Signal Processing*, 1–6. New York: IEEE.

Engel, J., T. Schöps, and D. Cremers. 2014. "LSD-SLAM: Large-scale direct monocular SLAM." In *Proc., European Conf. on Computer Vision*, 834–849. Dordrecht, Netherlands: Springer.

Fetić, A., D. Jurić, and D. Osmanković. 2012. "The procedure of a camera calibration using camera calibration toolbox for Matlab." In *Proc., 35th Int. Convention MIPRO*, 1752–1757. New York: IEEE.

Fischler, M. A., and R. C. Bolles. 1987. "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography." In *Readings in computer vision*, 726–740. Amsterdam, Netherlands: Elsevier.

Fuentes-Pacheco, J., J. Ruiz-Ascencio, and J. M. Rendón-Mancha. 2015. "Visual simultaneous localization and mapping: A survey." *Artif. Intell. Rev.* 43 (1): 55–81. https://doi.org/10.1007/s10462-012-9365-8.

Fuhrmann, S., F. Langguth, and M. Goesele. 2014. "MVE—A multi-view reconstruction environment." In *Proc., GCH*, 11–18. Geneva, Switzerland: Eurographics Association.

Furht, B., D. Grostick, D. Gluch, G. Rabbat, J. Parker, and M. McRoberts. 1991. *Introduction to real-time computing*, 1–35. Boston: Springer.

Gálvez-López, D., and J. D. Tardós. 2012. "Bags of binary words for fast place recognition in image sequences." *IEEE Trans. Rob.* 28 (5): 1188–1197. https://doi.org/10.1109/TRO.2012.2197158.

Gold, S., C.-P. Lu, A. Rangarajan, S. Pappu, and E. Mjolsness. 1995. "New algorithms for 2D and 3D point matching: Pose estimation and correspondence." In *Advances in neural information processing systems*, 957–964. Amsterdam, Netherlands: Elsevier.

Golparvar-Fard, M., J. Bohn, J. Teizer, S. Savarese, and F. Peña-Mora. 2011. "Evaluation of image-based modeling and laser scanning accuracy for emerging automated performance monitoring techniques." *Autom. Constr.* 20 (8): 1143–1155. https://doi.org/10.1016/j.autcon.2011.04.016.

Golparvar-Fard, M., F. Peña-Mora, and S. Savarese. 2009. "D4AR—A 4-dimensional augmented reality model for automating construction progress monitoring data collection, processing and communication." *J. Inf. Technol. Constr.* 14 (13): 129–153.

Golparvar-Fard, M., F. Peña-Mora, and S. Savarese. 2015. "Automated progress monitoring using unordered daily construction photographs and IFC-based building information models." *J. Comput. Civ. Eng.* 29 (1): 04014025. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000205.

Ham, Y., K. K. Han, J. J. Lin, and M. Golparvar-Fard. 2016. "Visual monitoring of civil infrastructure systems via camera-equipped unmanned aerial vehicles (UAVS): A review of related works." *Visualization Eng.* 4 (1): 1. https://doi.org/10.1186/s40327-015-0029-z.

Han, K., J. Degol, and M. Golparvar-Fard. 2018. "Geometry- and appearance-based reasoning of construction progress monitoring." *J. Constr. Eng. Manage.* 144 (2): 04017110. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001428.

Han, K., J. Lin, and M. Golparvar-Fard. 2015. "A formalism for utilization of autonomous vision-based systems and integrated project models for construction progress monitoring." In *Proc., 2015 Conf. on Autonomous and Robotic Construction of Infrastructure*, 118–131. Ames, IA: Iowa State Univ.

Han, K. K., and M. Golparvar-Fard. 2015a. "Appearance-based material classification for monitoring of operation-level construction progress using 4D BIM and site photologs." *Autom. Constr.* 53 (May): 44–57. https://doi.org/10.1016/j.autcon.2015.02.007.

Han, K. K., and M. Golparvar-Fard. 2015b. "BIM-assisted structure-from-motion for analyzing and visualizing construction progress deviations through daily site images and BIM." In *Proc., Int. Workshop on Computing in Civil Engineering*, 596–603. Reston, VA: ASCE.

Han, K. K., and M. Golparvar-Fard. 2017. "Potential of big visual data and building information modeling for construction performance analytics: An exploratory study." *Autom. Constr.* 73 (Jan): 184–198. https://doi.org/10.1016/j.autcon.2016.11.004.

Hedau, V., D. Hoiem, and D. Forsyth. 2009. "Recovering the spatial layout of cluttered rooms." In *Proc., 2009 IEEE 12th Int. Conf. on Computer vision*, 1849–1856. New York: IEEE.

Ibrahim, Y., T. C. Lukins, X. Zhang, E. Trucco, and A. Kaka. 2009. "Towards automated progress assessment of workpackage components in construction projects using computer vision." *Adv. Eng. Inf.* 23 (1): 93–103. https://doi.org/10.1016/j.aei.2008.07.002.

Karsch, K., M. Golparvar-Fard, and D. Forsyth. 2014. "Constructaide: Analyzing and visualizing construction sites through photographs and building models." *ACM Trans. Graphics* 33 (6): 176. https://doi.org/10.1145/2661229.2661256.

Kim, C., B. Kim, and H. Kim. 2013a. "4D {CAD} model updating using image processing-based construction progress monitoring." *Autom. Constr.* 35 (Nov): 44–52. https://doi.org/10.1016/j.autcon.2013.03.005.

Kim, C., H. Son, and C. Kim. 2013b. "Fully automated registration of 3D data to a 3D CAD model for project progress monitoring." *Autom. Constr.* 35 (Nov): 587–594. https://doi.org/10.1016/j.autcon.2013.01.005.

Košecká, J., and W. Zhang. 2002. "Video compass." In *Proc., European Conf. on Computer Vision*, 476–490. Dordrecht, Netherlands: Springer.

Krishna, C. M. 1999. *Real-time systems*. Atlanta: American Cancer Society.

Kropp, C., C. Koch, and M. König. 2018. "Interior construction state recognition with 4D BIM registered image sequences." *Autom. Constr.* 86 (Feb): 11–32. https://doi.org/10.1016/j.autcon.2017.10.027.

Lee, D. C., M. Hebert, and T. Kanade. 2009. "Geometric reasoning for single image structure recovery." In *Proc., IEEE Conf. on Computer Vision and Pattern Recognition, 2009 CVPR 2009*, 2136–2143. New York: IEEE.

Liu, Y., T. S. Huang, and O. D. Faugeras. 1990. "Determination of camera location from 2-D to 3-D line and point correspondences." *IEEE Trans. Pattern Anal. Mach. Intell.* 12 (1): 28–37. https://doi.org/10.1109/34.41381.

Lowe, D. 2004. "Distinctive image features from scale-invariant keypoints." *Int. J. Comput. Vision* 60 (2): 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94.

Lukins, T. C., and E. Trucco. 2007. "Towards automated visual assessment of progress in construction projects." In *Proc., BMVC*, 1–10. Coventry, UK: Univ. of Warwick.

Mapillary. 2018. "OpenSfM: Open source structure from motion pipeline." Accessed May 16, 2019. https://github.com/mapillary/OpenSfM.

Mirzaei, F. M., and S. I. Roumeliotis. 2011. "Globally optimal pose estimation from line correspondences." In *Proc., 2011 IEEE Int. Conf. on Robotics and Automation (ICRA)*, 5581–5588. New York: IEEE.

Moreno-Noguer, F., V. Lepetit, and P. Fua. 2008. "Pose priors for simultaneously solving alignment and correspondence." In *Proc., European Conf. on Computer Vision*, 405–418. Dordrecht, Netherlands: Springer.

Mur-Artal, R., J. M. M. Montiel, and J. D. Tardos. 2015. "ORB-SLAM: A versatile and accurate monocular slam system." *IEEE Trans. Rob.* 31 (5): 1147–1163. https://doi.org/10.1109/TRO.2015.2463671.

Mur-Artal, R., and J. D. Tardós. 2016. "ORB-SLAM2: An open-source slam system for monocular, stereo and RGB-D cameras." Preprint, submitted October 20, 2016. http://arxiv.org/abs/1610.06475.

Navon, R. 2007. "Research in automated measurement of project performance indicators." *Autom. Constr.* 16 (2): 176–188. https://doi.org/10.1016/j.autcon.2006.03.003.

NVIDIA. 2017. "Unleash your potential with the Jetson tx1 developer kit." Accessed November 29, 2017. https://developer.nvidia.com/embedded/buy/jetson-tx1-devkit.

Pătrăucean, V., I. Armeni, M. Nahangi, J. Yeung, I. Brilakis, and C. Haas. 2015. "State of research in automatic as-built modelling." *Adv. Eng. Inf.* 29 (2): 162–171. https://doi.org/10.1016/j.aei.2015.01.001.

Podbreznik, P., and D. Rebolj. 2005. "Automatic comparison of site images and the 4D model of the building." In *Proc., CIB W78 22nd Conf. on Information Technology in Construction*, 235–239. Dresden, Germany: Institute for Construction Informatics.

Přibyl, B., P. Zemčík, and M. Čadik. 2015. "Camera pose estimation from lines using Plücker coordinates." In *Proc., British Machine Vision Conf. (BMVC)*, edited by M. W. J. Xianghua Xie and G. K. L. Tam, 45.1–45.12. Delft, Netherlands: International Council for Research and Innovation in Building and Construction.

Pučko, Z., N. Šuman, and D. Rebolj. 2018. "Automated continuous construction progress monitoring using multiple workplace real time 3D scans." *Adv. Eng. Inf.* 38 (Oct): 27–40. https://doi.org/10.1016/j.aei.2018.06.001.

Qin, T., P. Li, and S. Shen. 2017. "VINS-mono: A robust and versatile monocular visual-inertial state estimator." Preprint, submitted August 13, 2017. http://arxiv.org/abs/1708.03852.

Ramey, R. 2004. "Serialization for persistence and marshalling." Accessed May 16, 2019. https://www.boost.org/doc/libs/.

Rebolj, D., N. Č. Babič, A. Magdič, P. Podbreznik, and M. Pšunder. 2008. "Automated construction activity monitoring system." *Adv. Eng. Inf.* 22 (4): 493–503. https://doi.org/10.1016/j.aei.2008.06.002.

Reconstruct. 2018. "Visual data analytics platform for construction." Accessed December 11, 2018. https://www.reconstructinc.com/.

Rossi, C., M. Abderrahim, and J. C. Diaz. 2005. "Evopose: A model-based pose estimation algorithm with correspondences determination." In Vol. 3 of *Proc., IEEE Int. Conf. Mechatronics and Automation, 2005*, 1551–1556. New York: IEEE.

Rother, C. 2002. "A new approach to vanishing point detection in architectural environments." *Image Vision Comput.* 20 (9–10): 647–655. https://doi.org/10.1016/S0262-8856(02)00054-9.

Shi, Y. 2004. "Particle swarm optimization." *IEEE Connections* 2 (1): 8–13.

Siebert, S., and J. Teizer. 2014. "Mobile 3D mapping for surveying earthwork projects using an unmanned aerial vehicle (UAV) system." *Autom. Constr.* 41 (May): 1–14. https://doi.org/10.1016/j.autcon.2014.01.004.

Snavely, N., S. M. Seitz, and R. Szeliski. 2006. "Photo tourism: Exploring photo collections in 3D." *ACM Trans. Graphics (TOG)* 25 (3): 835–846. https://doi.org/10.1145/1141911.1141964.

Snavely, N., S. M. Seitz, and R. Szeliski. 2008. "Modeling the world from internet photo collections." *Int. J. Comput. Vision* 80 (2): 189–210. https://doi.org/10.1007/s11263-007-0107-3.

Solihin, W., and C. Eastman. 2015. "Classification of rules for automated BIM rule checking development." *Autom. Constr.* 53 (May): 69–82. https://doi.org/10.1016/j.autcon.2015.03.003.

Stankovic, J. A. 1988. "Misconceptions about real-time computing: A serious problem for next-generation systems." *Computer* 21 (10): 10–19. https://doi.org/10.1109/2.7053.

Turkan, Y., F. Bosché, C. T. Haas, and R. Haas. 2013. "Tracking secondary and temporary concrete construction objects using 3D imaging technologies." In *Proc., Int. Workshop on Computing in Civil Engineering*, 749–756. Reston, VA: ASCE.

Tuytelaars, T., L. Van Gool, M. Proesmans, and T. Moons. 1998. "The cascaded Hough transform as an aid in aerial image interpretation." In *Proc., 6th Int. Conf. on Computer Vision, 1998*, 67–72. New York: IEEE.

Wu, C., et al. 2011a. "VisualSfM: A visual structure from motion system." Accessed May 16, 2019. http://ccwu.me/vsfm/doc.html.

Wu, C., S. Agarwal, B. Curless, and S. M. Seitz. 2011b. "Multicore bundle adjustment." In *Proc., CVPR 2011*, 3057–3064. New York: IEEE.

Xia, J., X. Xu, and J. Xiong. 2012. "Simultaneous pose and correspondence determination using differential evolution." In *Proc., 8th Int. Conf. on Natural Computation (ICNC), 2012*, 703–707. New York: IEEE.

Xu, C., L. Zhang, L. Cheng, and R. Koch. 2017. "Pose estimation from line correspondences: A complete analysis and a series of solutions." *IEEE Trans. Pattern Anal. Mach. Intell.* 39 (6): 1209–1222. https://doi.org/10.1109/TPAMI.2016.2582162.

Yang, J., M.-W. Park, P. A. Vela, and M. Golparvar-Fard. 2015. "Construction performance monitoring via still images, time-lapse photos, and video streams: Now, tomorrow, and the future." *Adv. Eng. Inf.* 29 (2): 211–224. https://doi.org/10.1016/j.aei.2015.01.011.

Yang, M. Y., Y. Cao, and J. McDonald. 2011. "Fusion of camera images and laser scans for wide baseline 3D scene alignment in urban environments." *ISPRS J. Photogramm. Remote Sens.* 66 (6): S52–S61. https://doi.org/10.1016/j.isprsjprs.2011.09.004.

Zhang, L., C. Xu, K. M. Lee, and R. Koch. 2012. "Robust and efficient pose estimation from line correspondences." In *Proc., Asian Conf. on Computer Vision*, 217–230. Dordrecht, Netherlands: Springer.

Zou, C., A. Colburn, Q. Shan, and D. Hoiem. 2018. "Layoutnet: Reconstructing the 3D room layout from a single RGB image." In *Proc., IEEE Conf. on Computer Vision and Pattern Recognition*, 2051–2059. New York: IEEE.