



TSLAM: a tag-based object-centered monocular navigation system for augmented manual woodworking

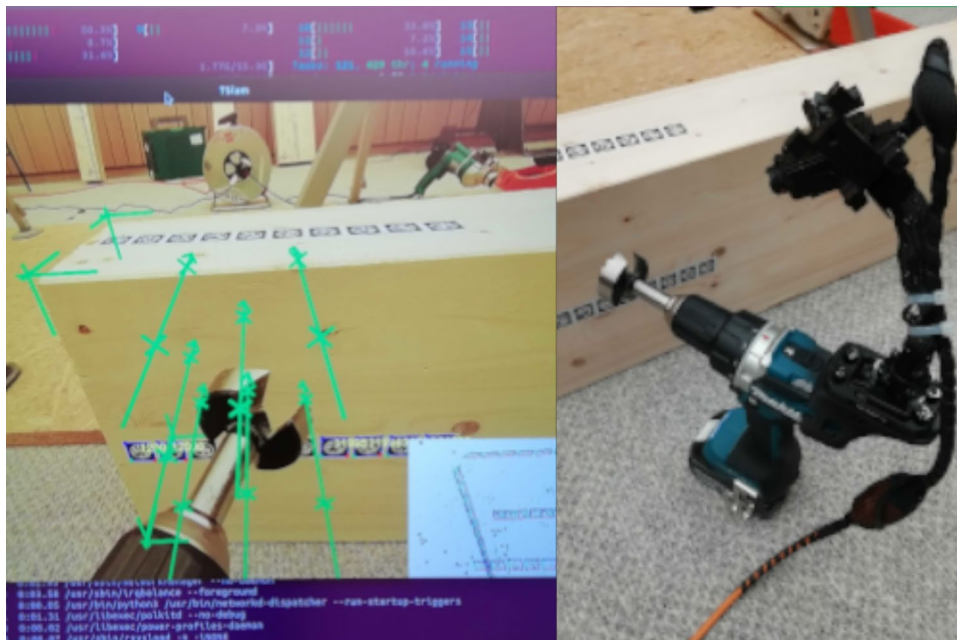
Andrea Settimi¹ · Hong-Bin Yang¹ · Julien Gamberro¹ · Yves Weinand¹

Received: 8 January 2024 / Accepted: 7 March 2024
© The Author(s) 2024

Abstract

TimberSLAM (TSLAM) is an object-centered, tag-based visual self-localization and mapping (SLAM) system for monocular RGB cameras. It was specifically developed to support a robust and augmented reality pipeline for close-range, noisy, and cluttered fabrication sequences that involve woodworking operations, such as cutting, drilling, sawing, and screwing with multiple tools and end-effectors. By leveraging and combining multiple open-source projects, we obtain a functional pipeline that can map, three-dimensionally reconstruct, and finally provide a robust camera pose stream during fabrication time to overlay an execution model with its digital-twin model, even under close-range views, dynamic environments, and heavy scene obstructions. To benchmark the proposed navigation system under real fabrication scenarios, we produce a data set of 1344 closeups of different woodworking operations with multiple tools, tool heads, and varying parameters (e.g., tag layout and density). The evaluation campaign indicates that TSLAM is satisfyingly capable of detecting the camera's millimeter position and subangular rotation during the majority of fabrication sequences. The reconstruction algorithm's accuracy is also gauged and yields results that demonstrate its capacity to acquire shapes of timber beams with up to two preexisting joints. We have made the entire source code, evaluation pipeline, and data set open to the public for reproducibility and the benefit of the community.

Graphic abstract



Extended author information available on the last page of the article

Published online: 16 April 2024

Keywords Self-localization and mapping (SLAM) · Computer vision · Digital fabrication · Augmented reality · Woodworking · Carpentry · Timber construction

Abbreviations

TSLAM	TimberSLAM
SLAM	Simultaneous localization and mapping
AR	Augmented reality
LiDaR	Light detection and ranging
IMU	Inertial measurement unit
VO	Visual odometry
VIO	Visual inertial odometry
ATE	Absolute trajectory error
RE	Relative (trajectory) error
GNSS	Global navigation satellite system
RTS	Robotic tracking station
MR	Mixed reality
6DoF	Six degrees of freedom
HMD	Head mounted display
AABB	Axis-aligned bounding box
FOV	Field of view
RE	Trajectory relative error metric
CFV	Coverage fabrication value

1 Introduction

1.1 Motivation

Research into the evolution of self-localization systems in subtractive fabrication operations in timber construction has received relatively limited attention compared with their additive counterpart. A deeper investigation into more accurate self-localization systems for subtractive fabrication workflows is thus warranted. Given the significance of prefabrication in timber construction through computer numerical control (CNC) machining technologies, precalibrated robotic setups Eversmann et al. (2017), Thoma et al. (2018), Thoma et al. (2019), Adel et al. (2018), Adel (2020), Adel (2023), and controlled shop environments, the need for sensor-based self-localization navigation systems is under-shadowed. By contrast, environment mapping, object and self-localization are inevitable core challenges to address early on in the design of any manufacturing processes that involve mobile robotics Dörfler et al. (2022) or augmented reality (AR). In additive augmented construction, this prerogative has propelled researchers into the development of reliable solutions for visual navigation that obtains millimeter precision Sandy et al. (2016), Sandy and Buchli (2018), Mitterberger et al. (2020). In our previous work Settimi et al. (2022), we demonstrated how subtractive AR fabrication can become possible once ordinary tools are equipped with visual sensors. We also outlined the potential of tool-aware

fabrication in the digital construction landscape. Yet, numerous technical challenges remain unsolved, among which the most critical is the identification of a reliable visual navigation methodology for subtractive tasks. Machining vibrations, a lack of geometric features, close-range sequences, visual noise (e.g., chips), a dynamic lighting environment, as well as the constant manipulation of the timber element contributed to poor self- and object-localization, which was majorly responsible for the low overhaul fabrication tolerance that we obtained with our first prototype (10 mm). To address all of the issues that we encountered, we propose a millimeter-accurate navigation system for augmented subtractive woodworking tasks named TimberSLAM (TSLAM), a monocular hybrid object-oriented self-localization and mapping framework. It is capable of reconstructing a 3D model of the piece to be manufactured and robustly overlaying it onto its physical twin via an AR interface during the woodworking phase. TSLAM's efficiency under real-life construction conditions is gauged through an experimental campaign specifically created for this study.

1.2 Outline of this paper

First, in the remainder of this section, we present a review of the state of the art to emphasize the key research contributions that form the foundation of TSLAM. Next, Sect. 2 presents the general function and implementation details of the proposed methodology. Next, Sect. 3 provides an overview of the evaluation campaign design before presenting and discussing its experimental results. Then, Sect. 4 highlights the current study's limitations and possible avenues for improvement. Finally, in Sect. 5, we draw conclusions and assess the impacts of the proposed software for research in subtractive augmented fabrication.

1.3 Relevant works

The following literature review is divided into two sections. The first exposes the current state-of-the-art of self-localization algorithms contextualized to digital fabrication relevance. The second offers an overview of the available data sets for benchmarking.

1.3.1 Self-localization in digital fabrication

Visual navigation systems are generally solved using natural landmarks or optical flow. Their particularity is that they can

perform mapping and self-localization of a sensor's pose simultaneously; these algorithms are known as simultaneous localization and mapping (SLAM). Hence, if an area that has already been mapped is tracked again, the algorithm can relocate the current position of the agent.

Feature-based SLAM: Among all the available approaches Khairuddin et al. (2015), Taheri and Xia (2021), Barros et al. (2022), indirect SLAM occupies a preeminent position. Feature-based or indirect SLAMs infer landmarks or features from sensor measurements to determine the trajectory and map of a mobile agent's environment. The series of ORB-SLAMs Mur-Artal et al. (2015), Mur-Artal and Tardos (2017), Campos et al. (2021) represents one of the most implemented open-source, feature-based monocular systems of this genre. Klein and Murray (2007) proposed an indirect SLAM version tailored to tracking a hand-held camera specifically for small-scale workplaces. Although this technique may acquire a good pose with little resources or be able to track the target, these methods often fail to do so due to texture-less areas, illumination changes, or motion blur. Close-up views might also result in a substantial lack of features. Therefore, a popular approach in mobile robotic applications is to complement visual SLAM algorithms with inertial sensing. OKVIS Leutenegger et al. (2014), ROVIO Bloesch et al. (2015), Bloesch et al. (2017), VINS-Mono Qin et al. (2018) and SVO Forster et al. (2017)–MAV Lynen et al. (2013) are state-of-the-art examples of this strategy. Recently, Johns et al. (2020), adapted a graph SLAM from Dube et al. (2017), and fused it with Global Navigation Satellite System (GNSS) and inertial measurements to localize an unmanned robotic excavator Jud et al. (2021) to a target wall.

Direct SLAM: In contrast to feature-based algorithms, direct SLAMs compute the agent's pose based on feature extraction from, for example, depth Shin et al. (2018), stereo Engel et al. (2015), or most frequently, RGB monocular Engel et al. (2014), Gao et al. (2018), Li et al. (2020) sensor data. Regardless of the captured raw data, they utilize pixel-level information directly. To estimate the camera's motion and reconstruct a dense 3D map, direct SLAMs often minimize the photometric error between consecutive or across multiple frames. In environments with few distinctive features, dynamic environments, or surfaces with low textures, direct SLAM methods are advantageous. Despite this, they often require substantial resources, and their performance may be adversely affected by changes in lighting and reflectance.

Deep-learning and semantic SLAM: In recent years, deep learning techniques have been integrated into SLAM algorithms with promising early results, especially for monocular applications Li et al. (2022), Mokssit et al. (2023). They either replace traditional geometric filters as in LIFT-SLAM Bruno and Colombari (2021), inform the algorithm with

semantic data McCormac et al. (2017), or compute the pose from on-the-fly inferences of depth Tateno et al. (2017), Li et al. (2021). Despite the precise results in some scenarios in the order of centimeters Li et al. (2021), such approaches are out of the millimetric-precision requirements typically necessary for woodworking operations.

Object-oriented SLAM: Additive in-situ manufacturing is a digital fabrication research sector where self-positioning is still a topic of interest Alatisse and Hancke (2020), Dörfler et al. (2022). From static environments and preregistered point cloud maps Dörfler et al. (2016), BIM models' geometric data are progressively levered to develop more object-oriented SLAMs Sandy et al. (2016). While edge-based object tracking Lowe (1991), Bouthemy (1989) in SLAMs Salas-Moreno et al. (2013) are a long-standing topic in computer vision, it is with Sandy and Buchli (2018), and the resulting demonstrator in AR bricklaying from Mitterberger et al. (2020) that monocular object-based localization systems have demonstrated their potential in digital additive construction. However, in subtractive fabrication scenarios, since the target shape dynamically changes throughout the fabrication sequence (e.g., top-end cutting or half-lap joinery), there are limitations to the reliability of such a typology of tracking.

Tag-based SLAM: Previously described methods often do not provide a scaled map of the environment. Since the scale is unknown, real measurements cannot be extrapolated from it. This can be a serious limitation, particularly for fabrication tasks. Conversely, tag-based SLAM systems, through the use of artificial landmarks, are capable of providing mapping with known scales; therefore, they can provide reliable measurement feedback. Not only are they more robust than other methods but they are also more resilient to the absence of textures, distinct geometric features, and dynamic scenes. However, tags need to be introduced in the environment beforehand, which generates an additional preparation phase before the fabrication itself. Davison et al. (2007) were the earliest to use fiducial markers for initialization in a monocular SLAM system. Early attempts at reconstructing wide-area fiducial marker maps Klopschitz and Schmalstieg (2007), and consequent relocation Shaya et al. (2012) exist. Nonetheless, SPM-SLAM Muñoz-Salinas et al. (2019b) represents the first fully functional tag-based system where markers from the ArUco library Garrido-Jurado et al. (2014, 2016) simultaneously help to localize the camera pose and build a map with a known scale. Similarly, TagSLAM Pfrommer and Daniilidis (2019) provides an approach with different square planar tags called AprilTags Wang and Olson (2016). One disadvantage of this technique is that it requires a large number of markers to be placed to construct the map. The reason for this is that for a connection between two markers to be established, at least two markers must be visible in an image. Muñoz-Salinas et al. developed UcoSLAM Muñoz-Salinas

and Medina-Carnicer (2020) which addresses the aforementioned problem by fusing feature points and square fiducial markers Garrido-Jurado et al. (2014, 2016) in a simultaneous mapping and tracking method. In the current work, we propose an object-centric version of UcoSLAM Muñoz-Salinas and Medina-Carnicer (2020) as our main navigation system, with ad-hoc modifications and new features specifically designed for manufacturing scenarios. Although ArUco Garrido-Jurado et al. (2014, 2016) are the de facto standards for simultaneous mapping and self-localization algorithms, the landscape of fiducial markers untested under SLAM scenarios is significantly wider. Among all possibilities, and given the highly cluttered, close-ranged, and noisy scenario of woodworking operations, we implemented STag Benligiray et al. (2019) in our proposed TSLAM due to its capacity to perform accurate pose detection even under harsh visual conditions. At the time of its publication, STag Benligiray et al. (2019) was benchmarked against the ArUco Garrido-Jurado et al. (2014, 2016), ARToolKitPlus Wagner and Schmalstieg (2007) and the RUNETag Bergamasco et al. (2011, 2016). Gains in robustness were reported against ArUco Garrido-Jurado et al. (2014) and STag is an order of magnitude faster than RUNE-Tag Bergamasco et al. (2011, 2016). In more recent experimental comparisons Kalaitzakis et al. (2021) with ARTag (Fiala 2005), and AprilTag Wang and Olson (2016), STag has achieved excellent detection rates and position measurements; however, it is sensitive to larger distances, which ultimately does not represent a limitation for our close-range scenario. Fiducial marker-based SLAMs are also present in commercial woodworking products. Although only limited to planar operations, Shaper Origin[®] (Shaper 2021) is a commercial portable router powered by a monocular tag-based two-axis self-localization system. It integrates a spindle capable of automatically compensating for all routing imprecisions generated by hand-holding maneuvers. The tags are first mapped, and then their location is associated with a two-dimensional execution layout by the user. The camera can perform local visual odometry (VO) at fabrication time and display the augmented guidance feedback accordingly. A similar two-dimensional tag-based system was already prototyped in previous research Rivers et al. (2012). We adopt a kindred fabrication sequencing and localization system but extend it to linear elements employed in carpentry operations, such as rectangular-section beams, either intact or with pre-existing joinery.

Object reconstruction and locking: Besides the camera's self-localization, a common practice in augmented fabrication is to employ tags as a robust reference system to link execution models to their physical equivalents. In a study by Hughes et al. (2021) ArUco markers' poses Garrido-Jurado et al. (2014, 2016), from the Fologram[®]'s augmented framework Jahn et al. (2019), were registered to ensure

interpolation between the physical space, headset, and building model. The same tracking toolkit was employed by Parry and Guy (2021) to digitize various lengths of timber offcuts and inform the design on the spot. Furthermore, Larsson et al. (2019) employed fiducial markers to detect the six degrees of freedom of nonstandard timber branches in a CNC operating space. In a previous experiment Settimi et al. (2022) with a Hololens2[®] head-mounted display (HMD), we employed QR codes and LiDAR scanning techniques to ensure object locking before fabrication. Analogous investigations on QR codes were conducted by Kyaw et al. (2023) on glulam beams. By leveraging the Fologram[®] plug-in, they reported alignment values between static physical beams and the associated execution model of under 2 mm when the markers had a maximum interval of 0.38 m. Robotic manipulation in construction has also leveraged tags' capacity for the referencing and tracking of execution models in both controlled environments for timber plate insertion Rogeau et al. (2020), and for larger on-site robotic molding Gifftthaler et al. (2017), Lussi et al. (2018), Dörfler et al. (2019), where studies have demonstrated how to calibrate a computer-aided design (CAD) model to the same tags employed for the localization of the robotic end-effector. In a similar fashion, we also use preregistered tags' map in a dual manner in TSLAM. The tag-based map is used not only for three-dimensionally reconstructing the piece and referencing it to its execution CAD model but also for self-localizing the camera sensor. The rendering of the overlapped model over the physical object is likely to suffer a smaller misalignment compared with the previous examples of disjointed AR tracking systems. In such scenarios, the HMD's SLAM is not referenced or reconstructed from the tags calibrated to the physical object. Hence, through similar means to the planar reconstruction method proposed in PolyFit Nan and Wonka (2017), we exploit the map's tags to generate and reference it to an accurate reconstruction of the three-dimensional shape of a piece of timber to be fabricated.

1.3.2 SLAM's benchmark

Following the common literature, self-localization systems are usually benchmarked on state-of-the-art data sets. Since most of them traditionally target autonomous vehicle applications Chen et al. (2022), data sets are generated to include scenes that span from aerial contexts, such as the Zurich Aerial Vehicle Data Set Majdik et al. (2013, 2014, 2015, 2017) visual inertial EuRoC-MAV Burri et al. (2016), UZH-FPV Delmerico et al. (2019), Cioffi et al. (2022), or drone-ranging LiDaR system-based data sets Nguyen et al. (2021), Zhu et al. (2023), Kim et al. (2020). Urban scenarios are popular subjects in Kitti Geiger et al. (2012) and the more recent KITTI-360 Liao et al. (2022), which have been famously referenced as benchmarks, together with other urban data sets

with real kinematic ground truth Maddern et al. (2016, 2020), such as the MVSEC Zhu et al. (2018a, b). Moreover, natural, less densely urbanized data sets are also available, such as the North Campus Long-Term (NCLT) Carlevaris-Bianco et al. (2015), FinnForest Ali et al. (2020), and Cambridge Kendall et al. (2015) data sets. Yet, these famous data sets do not present any visual clues, including fabrication sequences. To diversify the visual scenery in data sets, researchers have also leveraged graphic engines to produce diversified and particular synthetic outdoor Wang et al. (2020) and indoor Handa et al. (2014), McCormac et al. (2017) landscapes. Although woodworking operations and associated visual effects (e.g., chipping, dusting, and end-effector motions) would most probably represent a challenge to obtaining a realistic level of detail, both in terms of physics simulation and photorealism, indoor data sets, such as the RGB-D TUM Sturm et al. (2012), offer a highly generic set of scenarios that are suitable for the majority of domestic AR deployments Golodetz et al. (2018), Gao et al. (2022). The most referenced data set for AprilTag Olson (2011)-based SLAMs is also features recordings from interiors Muñoz-Salinas et al. (2019b). Terrestrial data sets produced from hand-held devices or robotic platforms present dynamic transitions between indoor and outdoor environments as well as unstable light conditions, which reinforce the similarity with unstructured construction surroundings Recchiuto et al. (2017). They also often present a wide plethora of visual interferences and highly changing architectural environments Schubert et al. (2018), Klenk et al. (2021), Engel et al. (2016); yet, the context remains broadly urban or dwelling-related.

Nevertheless, in recent years, the digital construction community has become increasingly interested in testing and developing SLAM software specifically designed to be deployed for building tasks, such as robotic in-situ operations or construction monitoring. The HILTI® data set Helmberger et al. (2022) is an important asset that contains multi-sensor recordings collected in industrial areas and at construction sites. The ConSLAM Trzeciak et al. (2023) contains time-synchronized ground truth data and spatially aligned images from a LiDaR sensor. However, to the best of our knowledge, there are no suitable data sets that involve SLAM or VO/VIO for close-range fabrication scenarios. Thus, we decided to create our data set of close-up woodworking operations.

2 Methodology

The current chapter illustrates each TSLAM's components' functioning and structure. TSLAM was developed on top of UcoSLAM Muñoz-Salinas and Medina-Carnicer (2020), and core features were adjusted. Additional functionalities and all modifications are discussed in this section. TSLAM was designed specifically to provide a robust and reliable

navigation system for AR-guided subtractive processes in carpentry. TSLAM is written in C++, leverages only CPU computing, currently targets UNIX platforms, and can be integrated into any project as a third-party API. We tested our system on a portable NUC station with an Intel 4-Core i7-1360P processor, 32 GB RAM, and a 2D monitor as designated AR support. We have made the source code publicly accessible for the benefit of the community Settimi and Yang (2023). The designated sensor for TSLAM is an RGB monocular camera RunCam (2023) (Fig. 1a) tethered to a computational unit. We target this category of sensing devices because of their availability, reduced size, and large field of view (FOV; in our case 142°), which could ultimately be adapted to the largest variety of AR devices and easily deployed on onboard sensing systems, such as the ones implemented for our manual electric tools, as shown in Fig. 1e, c.

The proposed SLAM is characterized by the following dual and sequential workflow (Fig. 2): (i) *timber mapping*, and (ii) *fabrication relocalization*. In the first phase (i), the timber (e.g., a beam) is mapped, and TSLAM performs in a hybrid format between tracking and mapping to obtain a SLAM map as well as a 3D reconstruction of the timber geometry. In the second phase (ii), at fabrication time, only the self-localization component is active, with an AR interface that provides a digital overlay on the physical piece. The following subsections describe the two phases in detail.

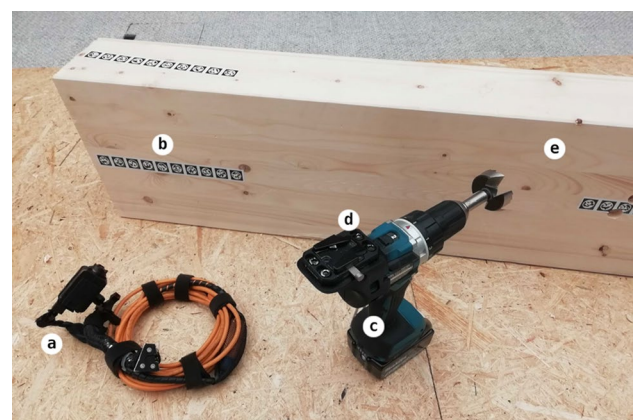


Fig. 1 Overview of TSLAM's hardware components: **a** tethered monocular RGB sensor; **b** sticker stripes of fiducial markers; **c** electric hand tool; **d** ad hoc 3D-printed mount to attach the sensor to the tool during fabrication; and **e** designated timber element to be fabricated. In addition, an external monitor is required for visualizing the augmented fabrication information

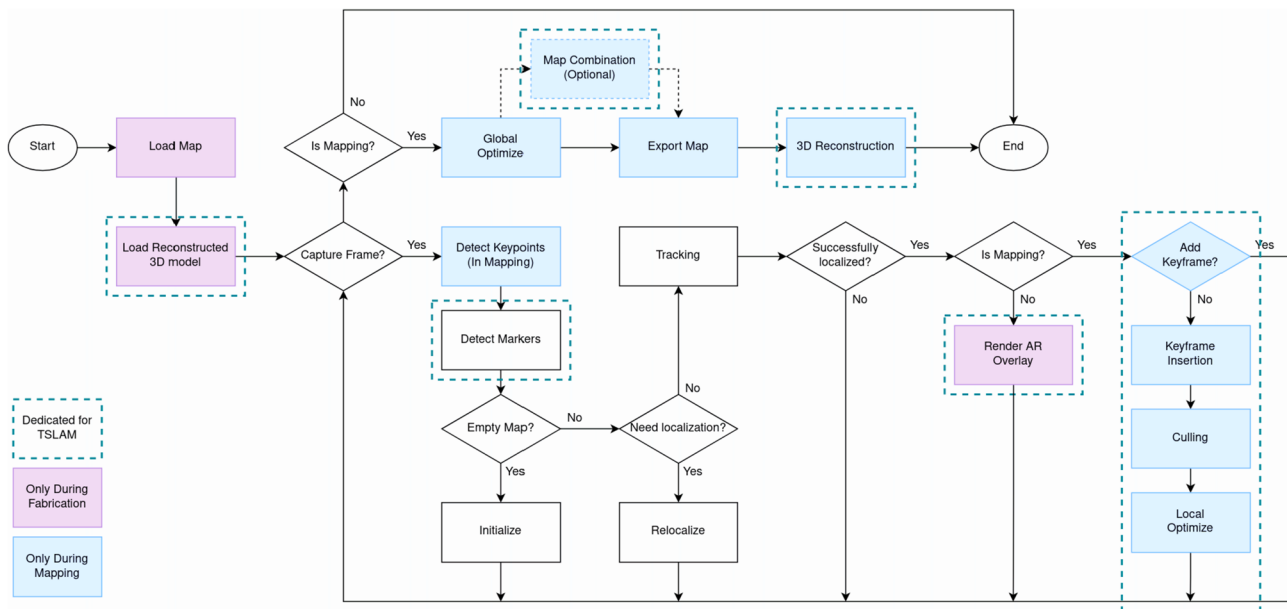


Fig. 2 TSLAM's workflow: in blue, the components that are uniquely active during the mapping phase; in pink, the ones in use during the fabrication phase. Highlighted in hidden blue lines are the components specifically developed for our fabrication scenario

2.1 Timber mapping

The first step before fabrication consists of applying tags to the timber element, mapping it, and obtaining a 3D reconstruction of its geometry.

2.1.1 Map recording

As introduced in the literature review (Sect. 1.3.1), in TSLAM, we opted for the STag Library HD11 Benligiray et al. (2019) as our designated artificial marker system. This library's package consists of 22,309 different tags and has the lowest false positive detection rate as per the original paper's test results. To simplify the tagging process on timber, we developed a generator to automatically output ready-to-print stripes of tags. Each stripe is approximately 1 m long and accommodates 47 tags, resulting in a total of 474 unique stripes. Considering the later fabrication phase, the camera is installed on the tool and is thus very close to the timber surface, to which the tags are attached. Consequently, the tags should be reduced in size to fit in a batch within the camera's view. By contrast, if the tags are too small, the system may struggle to recognize them or produce additional errors. In our empirical investigation, we determined that the optimal dimensions are 2 cm × 2 cm (Fig. 3a, b). This close-range detection can benefit the context of workshop woodworking environments. The advantage arises from the capacity to automatically filter out extraneous tags associated with other already mapped timber elements situated in the background (Fig. 3c), thus limiting false tag detection.

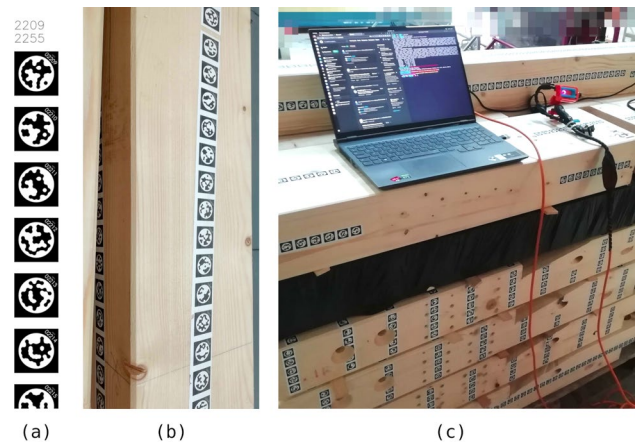
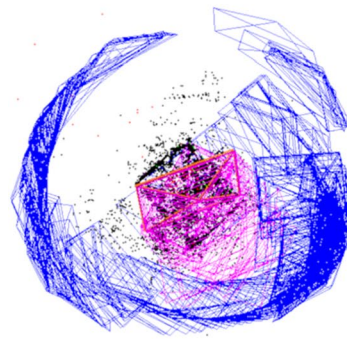


Fig. 3 TSLAM's STags system: **a** a zoom of a portion of the 1-m-long stripe composed by multiple markers; **b** a view of a tag stripe applied to a piece of timber's face, to note that the stripe does not need to be straight; and **c** a stack of mapped or already fabricated beams stored in the woodworking shop

Once the stripes are attached to each timber face, the recording for mapping can commence.

TSLAM's map is incrementally built as the monocular sensor navigates around the timber element (Fig. 4a). It is an object-centric collection of keyframes, feature points, and fiducial markers (Fig. 5). The system also maintains a connection graph \mathcal{G} , which is a hidden data structure representing the strength of each keyframe's connectivity, alongside a keyframe recognition database storing the Bag-of-Word (BoW) of extracted keyframe features as

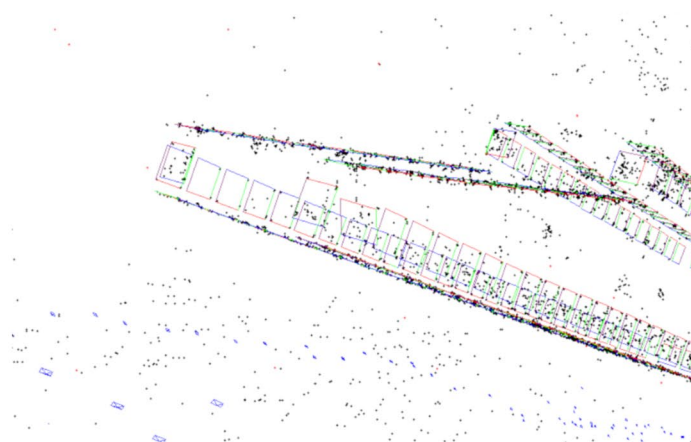
Fig. 4 **a** Frontal view of the registered map of a timber beam—note that the camera poses (in blue and magenta) are expressed in the timber’s coordinate system and **b** view of the mapping interface



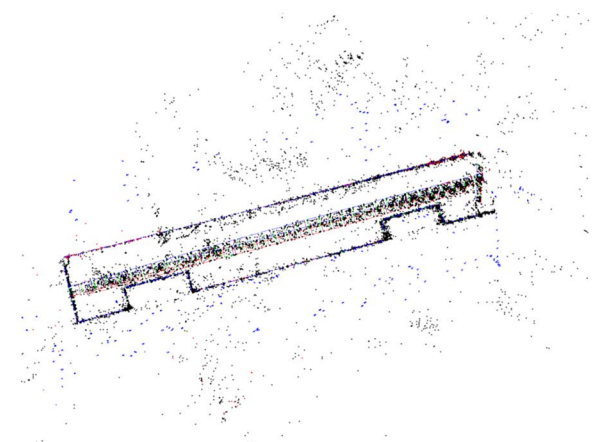
(a)



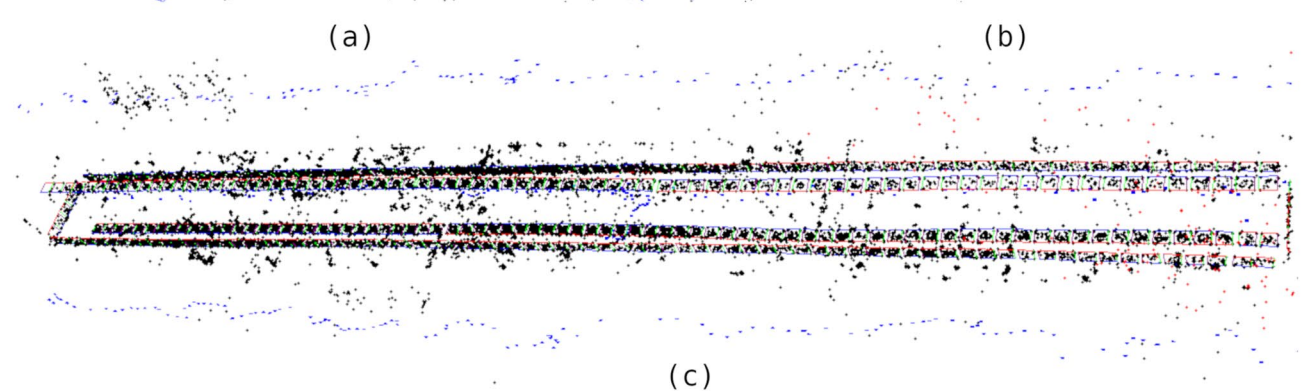
(b)



(a)



(b)



(c)

Fig. 5 Multiple views of different TSLAM maps: **a** detail of a map where a scarf joint can be guessed; **b** map of a moderately sized timber element with two half-lap joints; and **c** longer element 3 m in length

proposed in Galvez-López and Tardos (2012). Having the map referent to the object and not the landscape is crucial for solving the problem of the continuous manipulation of the timber without losing track of the camera or object detection during the fabrication process.

Theoretically, the system could be initialized with either key points or markers. However, in our scenario, initializing

it with a marker is preferred because tags are always referenced to the target beam, whereas key points might also appear in the background. The system adopted the marker initialization strategy proposed in SPM-SLAM Muñoz-Salinas et al. (2019a). The map will be successfully initialized if a marker is unambiguously detected by one frame or ambiguously detected by two frames, which makes the

initialization process fast and robust. Another advantage of initializing the map using markers with known dimensions is that the system will have a scaling reference to the real world. Therefore, the constructed map will have a 1:1 scale with respect to the real world.

Once the map has been successfully initialized, the system continuously estimates the camera pose and updates the map accordingly. TSLAM’s pose estimation mechanism is similar to UcoSLAM Muñoz-Salinas and Medina-Carnicer (2020), but the hyperparameters that control the weight of map points and markers are modified to fit our scenarios. Here in Equation (1), we follow the same notation in Muñoz-Salinas and Medina-Carnicer (2020) to simplify the paragraph. If the camera pose was estimated in the previous frame, the system attempts to estimate the current one by minimizing the projection error. During the mapping phase, the projection error is calculated in a hybrid fashion with key points and markers jointly as follows:

$$\mathbf{f}_T = \underset{T}{\operatorname{argmin}} \left(\mathbf{w}_p^f H(\Upsilon_p^f, T) + \mathbf{w}_m^f H(\Upsilon_m^f, T) \right) \quad (1a)$$

where \mathbf{H} is the function for determining the reprojection errors according to the observation of key points (Υ_p^f) and markers (Υ_m^f) in frame \mathbf{f} with respect to the transformation \mathbf{T} . The weight \mathbf{w}_p^f and \mathbf{w}_m^f control the importance of points and markers. The key points and markers’ weights can be expressed, respectively, as follows:

$$\mathbf{w}_p^f = 1 - \mathbf{w}_m^f \quad (2a)$$

$$\mathbf{w}_m^f = \min \left(1, \frac{\mathbf{n}_f}{\tau_m} \right) \quad (2b)$$

where \mathbf{n}_f is the number of valid markers in frame \mathbf{f} , and τ_m is a threshold. That is, the system relies on markers but neglects the error generated by the point if there are more than τ_m markers present in a scene. The reason for this is that the timber is constantly manipulated during both mapping and fabrication, which results in an inconsistent reprojection. Since in our scenario the camera’s relative pose to the timber is more relevant than the background, diminishing the effectiveness of all points ensures that the noise created by detected background points does not affect the final pose estimation in both phases of TSLAM. Following the evaluation campaign (see Sect. 3.4), we set $\tau_m = 3$ based on the statistical result of \mathbf{n}_f across our own data set. However, pose estimation through key points allows TSLAM to bridge tags’ gaps and map longer elements that do not necessarily have a continuous portion of wood populated by tags (e.g., glulam beams with major spans, as in Fig. 6h or 7).

If the system fails to estimate the pose, it will wait until a known marker is present in the scene again, and only within

this instance would it perform relocalization. With an unambiguously detected marker, the pose can be estimated. Relocalization through key points is disabled in TSLAM because of its low performance as well as its low importance to our scenario. To scan all six faces of a timber beam, one may need to flip the beam during scanning. Although TSLAM is highly resilient to manipulation, this might occasionally lead to inconsistencies in the relative positions of objects within the scene, resulting in calculation errors and the subsequent creation of a defective map. TSLAM provides dedicated functionality for combining separately scanned maps. To achieve such a result, commonly shared markers are instrumental. Here, we calculate the affine transformation matrix $\mathbf{M}_A \in \mathbb{R}^{4 \times 4}$ by OpenCV (v4.5.0)’s *estimateAffine3D* function (Itseez 2015) using the four corner points of the common markers as references. Say that one has one source map and a target map to combine. One can apply \mathbf{M}_A across all of the elements in the source map to transform it to align with the

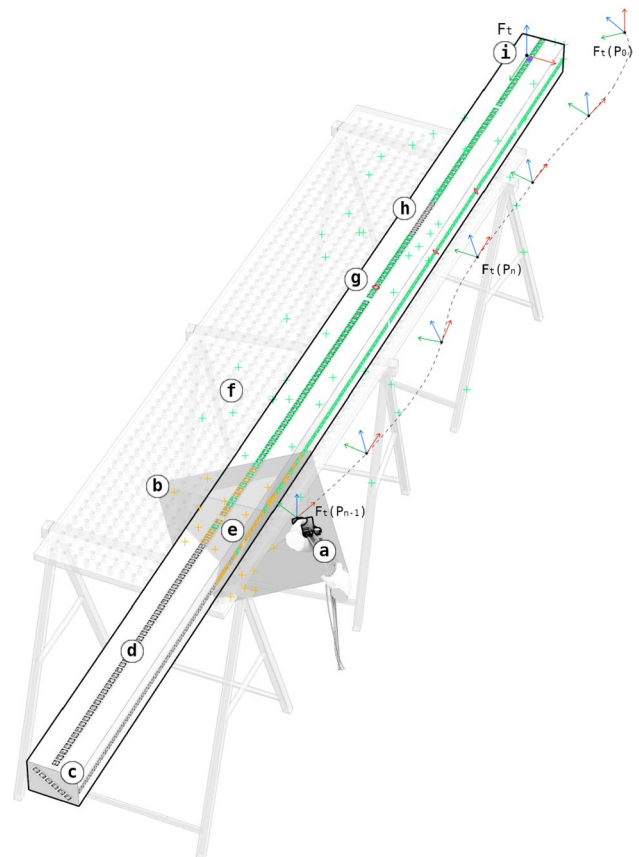


Fig. 6 Schematization of the mapping process: **a** hand-held monocular sensor; **b** representation of the sensor’s field of view; **c** piece of timber; **d** stick tags that have not yet been mapped; **e** ongoing processed tags and features points in yellow and green for the tags registered in the map; **f** during the mapping, feature points are also detected in the scene background; **g** incorrect or corrupted tag poses, which are corrected in the latest global optimization stage; **h** possible portion of tags not registered due to human error; and **i** first detected tag’s frame becomes the global reference for the rest of the map

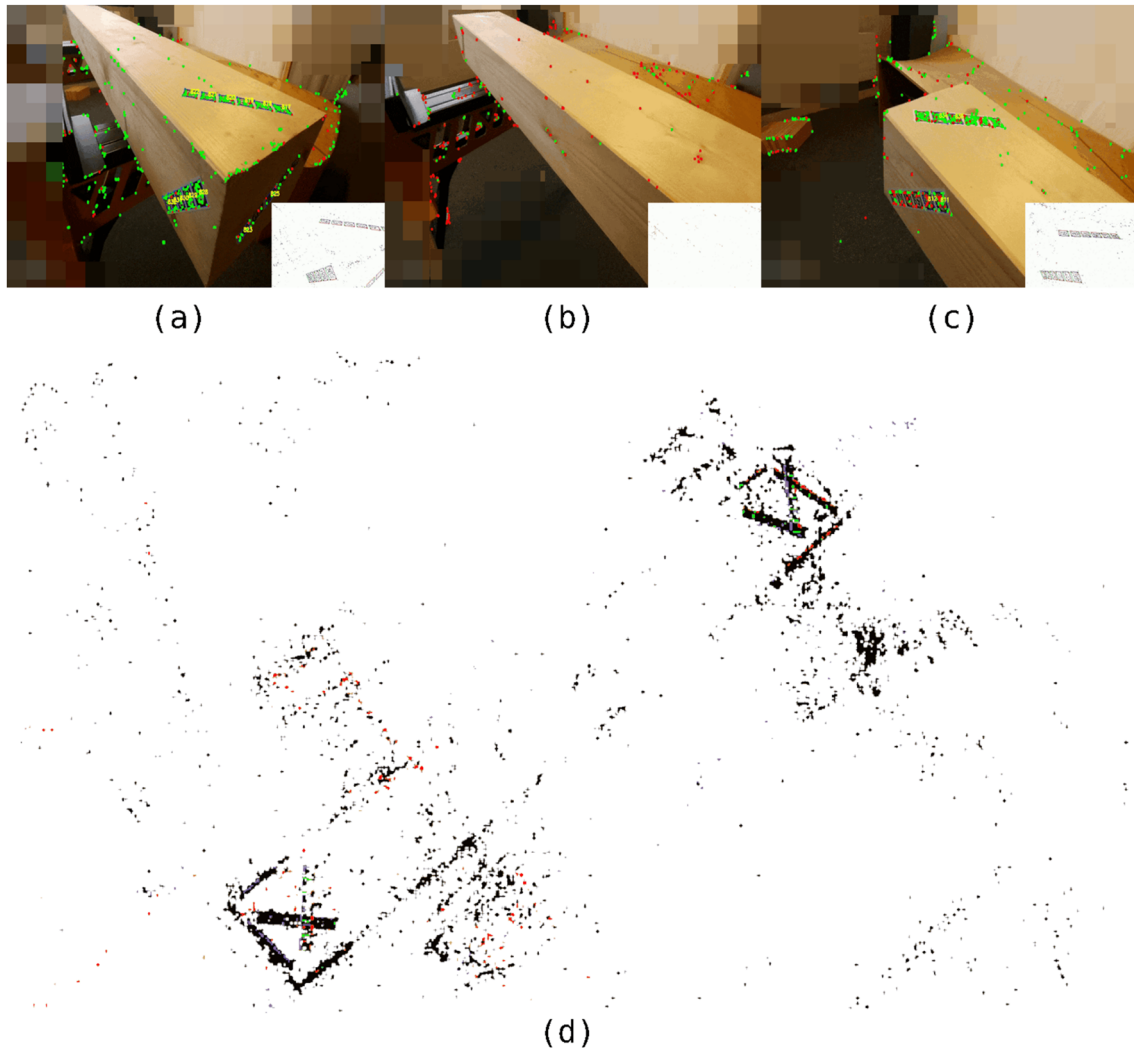


Fig. 7 Dual functioning of TSLAM allows for continued mapping even if tags are not continuous along the timber element. It might often occur that only the extremities need to be processed; in this

case, tags could indeed be limited uniquely to those areas: **a–c** mapping sequencing of the start, middle, end, and **d** the output map

target map. Since the timber is a rigid body of unchanged shape during the mapping procedure, one can assume that the same M_A is used to define the transformation for each pair of marker corner points and can be found by the least squares solution, such that

$$\sum_{m \in \mathbf{m}_c} \left\| \begin{bmatrix} \mathbf{c}_m^t \\ 1 \end{bmatrix} - M_A \cdot \begin{bmatrix} \mathbf{c}_m^s \\ 1 \end{bmatrix} \right\| \tag{3a}$$

is minimized, where \mathbf{m}_c is the set of common markers, \mathbf{c}_m^t is the position of the corner of the common markers in the target map, and \mathbf{c}_m^s is their corresponding position in the source map. Later, the transformation is applied to reposition the rest of the key points, markers, and keyframes, effectively merging them into a unified representation. Throughout the mapping recording, an inevitable drift of

the camera occurs and must be detected and corrected. In UcoSLAM Muñoz-Salinas and Medina-Carnicer (2020), a global optimization and a simplified local optimization were used to correct such errors. The global optimization, which affects the entire map, runs when a loop closure is detected. On the other hand, local optimization is active when a new keyframe is added to the map and only affects its neighbors within the connection graph \mathcal{G} . However, global optimization is a slow process that creates undesirable latency and thus harms the woodworking operation with unwanted latencies. In TSLAM, loop closure detection is disabled, and only when the mapping phase is complete does the recorded map undergo a global optimization process to refine and correct inconsistencies (Fig. 6g). The result is saved as a serialized file to be loaded later at fabrication time. In addition, the system exports supplementary metadata that contain IDs and

corner positions of markers, which are used as input for the reconstruction of a 3D model.

2.1.2 Model reconstruction

As carpentry interventions often involve both off-the-shelf squared sections and irregular elements that incorporate pre-existing joints, TSLAM was also required to acquire these irregular and more complex geometries. Hence, similar to Nan and Wonka (2017), we designed a reconstruction

algorithm that first obtains all possible planes from the geometry's faces before selecting the best candidate polygons to compose the mesh's faces. Therefore, by detecting the tags' positions during the mapping phase, the proposed pipeline obtains an accurate 3D reconstructed mesh of an irregular physical object with a simple monocular camera. The current multi-step algorithm is uniquely based on the tags' detected poses registered during the mapping. The geometric solver can be resumed as illustrated in the following stack flow in Algorithm 1.

Algorithm 1 The geometric solver algorithm

Input: $T \leftarrow$ Input Tags
Input: $\theta \leftarrow$ Angle tolerance to be considered as co-plane
Input: $d \leftarrow$ Distance tolerance to be considered as co-plane

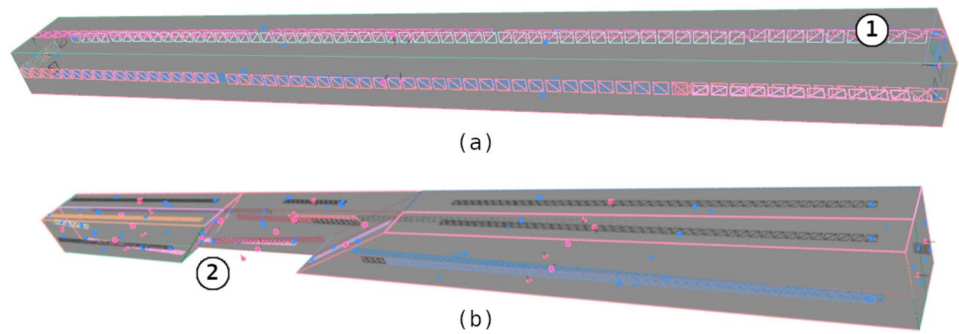
$S \leftarrow \text{NormalBasedSegment}(T)$
 $P_S \leftarrow \{\phi\}$ \triangleright Set of planes formed by each segment
for all $S_i \in S$ **do** \triangleright Retrieve the average plane for each segment
 $P_{S_i} \leftarrow \{\phi\}$
 for all $T_k \in S_i$ **do**
 $P_{S_i} \leftarrow P_{S_i} \cup \text{Plane}(T_k)$
 end for
 $P_S \leftarrow P_S \cup \overline{P_{S_i}}$
end for

$P \leftarrow \{\phi\}$ \triangleright All detected planes
for all $P_{S_i} \in P_S$ **do** \triangleright Merge planes within the tolerance
 for all $P_j \in P$ **do**
 if $\angle(P_{S_i}, P_j) \leq \theta \wedge D(P_{S_i}, P_j) \leq d$ **then**
 $P_j \leftarrow \text{merge}(P_j, P_{S_i})$
 end if
 end for
 if not merged then
 $P \cup P_{S_i}$
 end if
end for

$AABB \leftarrow$ Axis-aligned bounding box of all tags' corners
 $C \leftarrow \text{Intersect}(P, AABB)$ \triangleright Polygons of plane–plane intersection
 $F \leftarrow \{\phi\}$ \triangleright Set of selected face candidates
for all $C_i \in C$ **do** \triangleright Select face candidates based on inline tags
 for all $T_i \in T$ **do**
 if $\angle(\text{Plane}(T_i), \text{Plane}(C_i)) \leq \theta$ **and** $D(p_{T_i}, \text{Plane}(C_i)) \leq d$ **then**
 $F_f \cup \text{BoundarySurface}(C_i)$
 end if
 end for
end for

$M \leftarrow \text{Assemble}(F)$ \triangleright The reconstructed mesh

Fig. 8 Examples of reconstructed mesh: **a** six-face beam; **a1** if they belong to the same face, the tag stripes are clustered; **b** timber stud that presents a successfully modeled half-lap joint



The first step consists of parsing the tags \mathbf{T} by detecting the timber’s geometry. The tags’ corners are first bundled into stripes \mathbf{S} that belong to the same piece of timber’s face by applying a normal-based segmentation from the Cilantro library library Zampogiannis et al. (2018) (Fig. 8a1). For each cluster, we obtain a single plane \mathbf{P}_S , defined by the average of all the cluster’s tags’ planes. Duplicates of the resulting planes are subsequently merged averagely by checking their coplanarity within a given angular θ and distance threshold \mathbf{d} . Ultimately, we obtain \mathbf{P} , which is a set of unique planes.

Now that we have acquired the entire planar pool, we intersect them with the axis-aligned bounding box (AABB) of the tags’ corners. The obtained polygons are later intersected among them, which results in a series of segments that link the two-point intersections for each couple of polygons. Next, a tessellation is performed between the segments and the polygons with the *arrangements2d* function from the CGAL package Wein et al. (2023). The outcome \mathbf{C} represents, at this stage, the pool of all possible polygons describing the subject’s faces. In the last step, the face candidates \mathbf{F} are selected by testing whether, considering all tags and a given threshold distance, at least one projection of a tag’s center \mathbf{p}_T onto the polygon’s associated planes falls inside its perimeter (Fig. 8b2). As an additional refinement, we also verify that the potential candidate tag and polygon’s normals have the same direction. It is important to note that, for the success of the last step, all physical faces of the object must be occupied by at least one fiducial marker.

Finally, the selected faces are assembled to create a watertight triangular mesh \mathbf{M} . The mesh is registered in TSLAM’s reference system and can be exported and processed in any CAD software. Later, the mesh model, which is enriched with fabrication data (e.g., holes, cuts, lines, and labels), can be reintroduced in the same TSLAM world coordinates to be visualized as an augmented overlay. Once the mapping phase is terminated, TSLAM can be initialized for the woodworking phase.

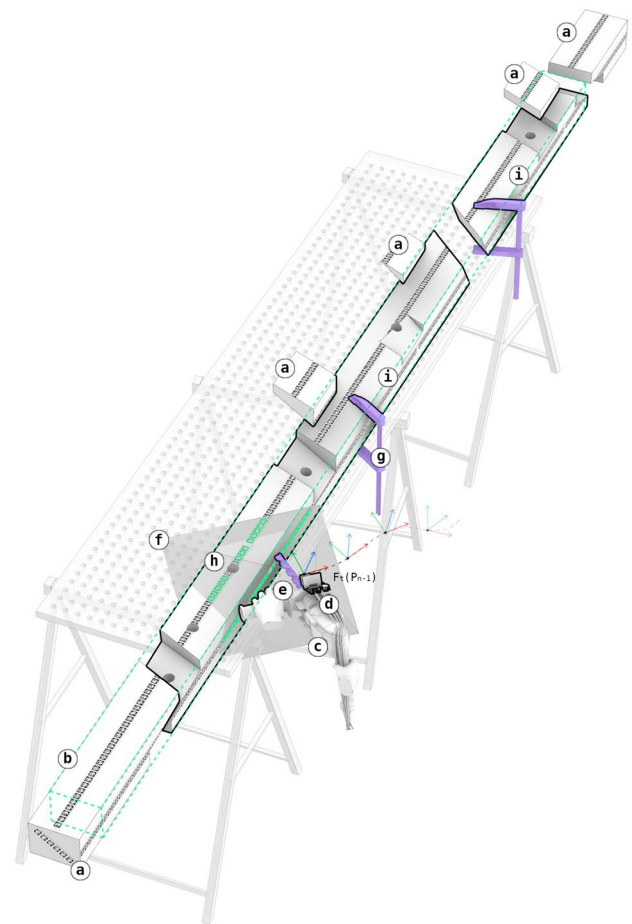


Fig. 9 Illustration of TSLAM’s self-localization during fabrication: **a** produced scraps; **b** AR model overlapped-object-locking is ensured in any portion of the beam even if severed; **c** electric manual tool embedded in **d** the monocular camera; **e**, **g** the tool’s end-effectors, clamps, and timber chips are the main source of occlusion; **f**, **h** the represented camera’s field of view and the detected tags employed to estimate the current camera pose with respect to the previously mapped beam. TSLAM can accommodate multiple output pieces within the same mapped element **i**

2.2 Fabrication relocation

At the fabrication stage, we designed TSLAM to perform with the same camera employed for mapping but now mounted on the tool (Fig. 9d). Although this condition is challenging for a SLAM system, it broadens possible avenues for the development of optimal AR applications in sub-tasks where the tool head is also monitored from the camera's view. To start the fabrication, the previously captured map and reconstructed model are loaded. At this juncture, TSLAM can relocalize the current camera position and orientation for the stored map (Fig. 9h). This implies that the timber piece can undergo unrestricted movement during fabrication, as our reference system is centered on the object itself rather than on the background.

The abundant tags within a single stripe also enhance the system's resilience, with particular significance for the camera pose relocalization's resistance to noise and shape alterations. Only markers are employed during the fabrication, meaning $w_p^f = 0$ and $w_m^f = 1$. Employing feature points would indeed be counterproductive in this application. Here, the visual feed is occluded by multiple noises generated by, for example, timber chipping (Fig. 10a) and sawdust (Fig. 10b). In addition, drill bits, blades, and metal guides

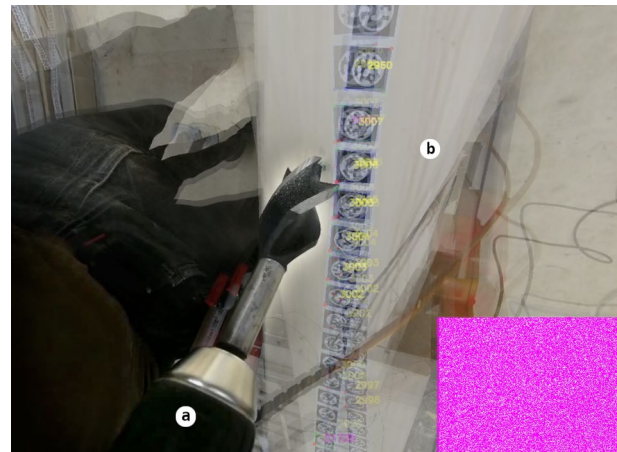


Fig. 11 Merged frames from 10 s of a video sequence: **a** tool heads always remain stationary in the camera buffer since the sensor rigidly follows the tool's movement, whereas **b** background is dynamic

(Fig. 10c) remain constantly stationary in the camera frame while the background is dynamic (Fig. 11).

In addition, the redundancy present in tags, combined with object-centric relocalization, facilitates effective camera pose estimation even in scenarios where the object



Fig. 10 Views of fabrication sequences during tracking by TSLAM. Due to the tags' redundancy, TSLAM is capable of performing self-localization against **a** chips and generated blur; **b** accumulation of

sawdust; and **c** tool attachments, such as the circular saw's guide cluttering the scene and obstructing tags



Fig. 12 Fabrication segments of a 500-mm-long auger drill bit, and b sawing with a saber saw

undergoes cutting, the loss of components, drilling, or alterations in its original form during the fabrication process. The only discriminating factor is the presence of mapped tags in the detection FOV (Fig. 12).

While close-range does not present a problem due to the large view provided by the chosen fish-eye lens, vibrations do. In opposition to the auger drill bit in Fig. 12b, which produces significant vibration, rotary motored tools placed at a long distance, as depicted in Fig. 12a provide the optimal conditions for TSLAM’s relocalization.

TSLAM can tightly overlay the scanned and processed detection mesh onto the mapped object (Fig. 9b). We integrated a basic AR model visualizer to showcase TSLAM’s capacity as a suitable and applied navigation system for subtractive woodworking scenarios. Given the ability to accurately position the camera to the map, the object retains its positional stability even when subjected to movement or shape modifications. Furthermore, the model remains consistently visualized in its correct location as long as a few key tags remain intact. On the other hand, we emphasize the fact that TSLAM’s scope of detection is tailored to close-range self-localizations. Hence, when a local detection occurs, one might notice that the farthest portion of the overlaid model could suffer minor offsets compared with the area closest to the sensor (Fig. 13a).

Given that the reconstruction process preserves the coordinate system, additional alignment between the reconstructed mesh and the scanned map is not required. By employing a perspective projection based on the camera matrix, points from 3D space can be mapped onto 2D screen coordinates. The perspective projection matrix M_{pp} concerning the camera intrinsic parameters is a 4×4 matrix (Eq. 4).

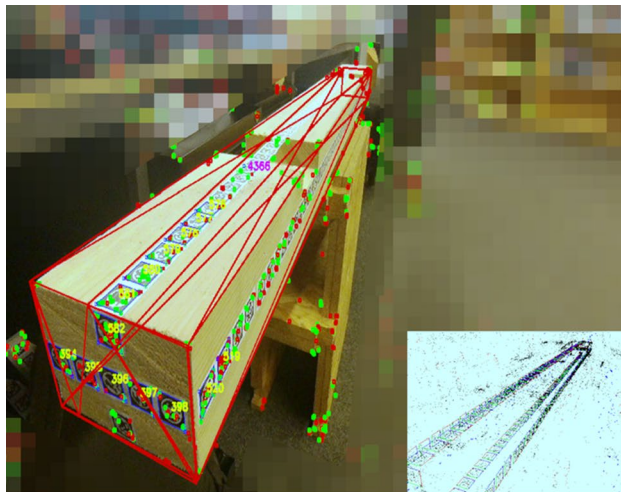
$$M_{pp} = \begin{bmatrix} \frac{2f_x}{w} & 0 & \frac{w-2c_x}{h-2c_y} & 0 \\ 0 & \frac{2f_y}{h} & -\frac{Z_f \times Z_n}{Z_f - Z_n} & -\frac{2 \times Z_f \times Z_n}{Z_f - Z_n} \\ 0 & 0 & -1 & 0 \end{bmatrix} \quad (4a)$$

where f_x and f_y is the camera’s focal length, w and h is the frame width and height, c_x and c_y is the camera’s optical center, and Z_n and Z_f is the near and far clipping plane. By applying the perspective project matrix M_{pp} to the point x_w in the 3D world coordinate, one can obtain the 2D axis in the screen coordinate $x_s \in \mathbb{R}^2$ by dividing w to convert it from homogeneous vector to position vector. The equation can be expressed as follows:

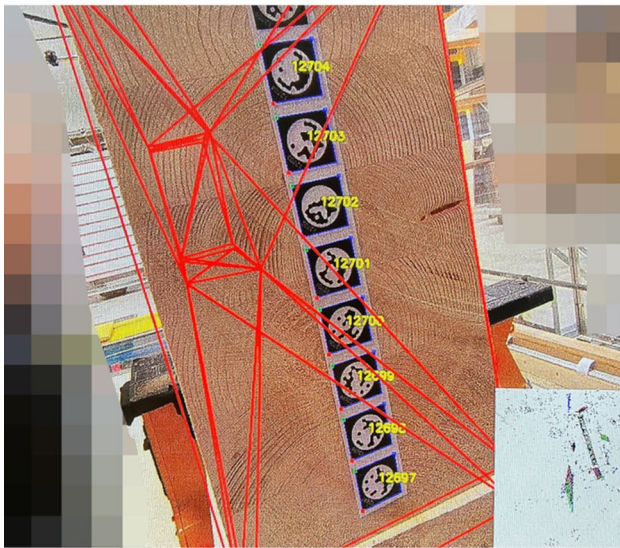
$$\begin{bmatrix} x_{pp} \\ y_{pp} \\ z_{pp} \\ w \end{bmatrix} = M_{pp} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (5a)$$

$$x_s = \left(\frac{x_{pp}}{w}, \frac{y_{pp}}{w} \right) \quad (5b)$$

After one obtains the 2D axis, lines connecting the corresponding points are drawn on the frame buffer to visualize the edges. Any other widget required for guiding the user can be represented similarly (Fig. 14a2, a3). We rendered the mesh outer contours uniquely to avoid encumbering the augmented view with unnecessary visual elements (Fig. 13b). Moreover, because rendering basic lines requires less computational power and is easier to implement, we did not rely on rendering engines, such as OpenGL, which keeps the system slim and flexible for future third-party integration. The



(a)



(b)

Fig. 13 Preliminary developments of model alignment: **a** early object-locking tests on 2-m-long studs; and **b** mesh overlay onto a larger glulam beam

subsequent chapter introduces the experimental campaign designed for the presented methodology.

3 Experimental campaign

In the experimental campaign presented in this section, we evaluated TSLAM based on contextually relevant woodworking scenarios. We aimed to assess the performance of TSLAM when applied to subtractive timber-related fabrication processes. As part of the evaluation of TSLAM, we analyzed the following aspects: (i) its capability to accurately determine the position and orientation of a set

of sensor-equipped tools in a dynamic fabrication environment; (ii) the algorithm's capacity to concurrently build a coherent 3D reconstruction of the timber piece; and (iii) the required preparation time compared with traditional marking techniques. Simultaneously, we sought to identify which tags' distribution results in the best tradeoff of the metrics described in Sect. 3.4.

3.1 Setup and parameters

To achieve the specified objectives, we designed a unique evaluation protocol. It resulted in an evaluation campaign that incorporated the following set of fixed and varying distinct variables:

1. *Timber's dimensions*: For the experimental campaign, we employed DUO laminated timber with an off-shelf square section of 14×14 cm. The beam's length was set to 2 m. This is a limitation imposed by the cubic tracking volume of 3 m, as described in Sect. 3.2. The chosen beam was able to allocate all necessary variants that impact the volumetry onto the same piece, as illustrated in Fig. 15.
2. *Joinery and fasteners*: We selected the following four joints among the most commonly employed in timber truss fabrication: scarf, half-lap, cross-lap, and notched (e.g., between the main rafter and tie-beam) joints (Fig. 15a1). To test the capacity of TSLAM to model pre-existing joints, we introduced a variation on the number of joints already present on the beam before the fabrication began (Fig. 15b1). For drilling operations, we selected a set of multiple piercing trajectories ranging from 30° to 90° with different drill bits (auger, self-eating and diagonal drill bits). In addition, we used four different screws with varying lengths—namely, 120, 100, 80, and 45 mm.
3. *Woodworking tools*: Based on a seemingly common carpentry practice, we limited the employed electric woodworking tools to the following: a drill with $\varnothing 18/25$ mm augers, $\varnothing 50$ mm self-feeding, and $\varnothing 35$ mm twist drill bits, a saber saw with a timber blade insert of length 300 mm and finally a circular saw with a blade of $\varnothing 190$ mm.
4. *Tags' distributions*: We selected four possible combinations among the multitude of possible combinations of tag distribution and density schemes. The empirical focus of this study mainly consisted of two different density levels and two discrete layout arrangements (Fig. 16). The layouts were limited to a configuration of tags, either longitudinal (*stripe*) or perpendicular (*rings*).

Fig. 14 Preliminary implementation of an AR overlay based on TSLAM’s object locking: **a1**, **b1** the detected tags used to estimate the camera pose; **a2** visual widgets that indicate the model overlay’s fitness to the physical twin; **a3** drilling indications; **a4** overview of the prerecorded map from the current camera view; and **b5**, **b6** the monocular sensor attached to the tool and tethered to the computing station

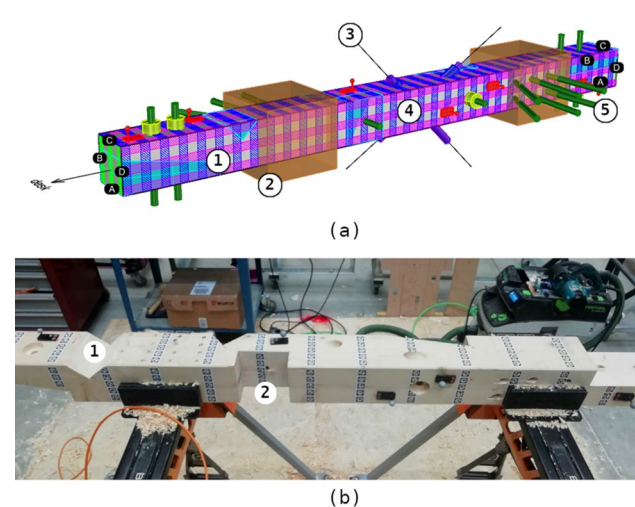
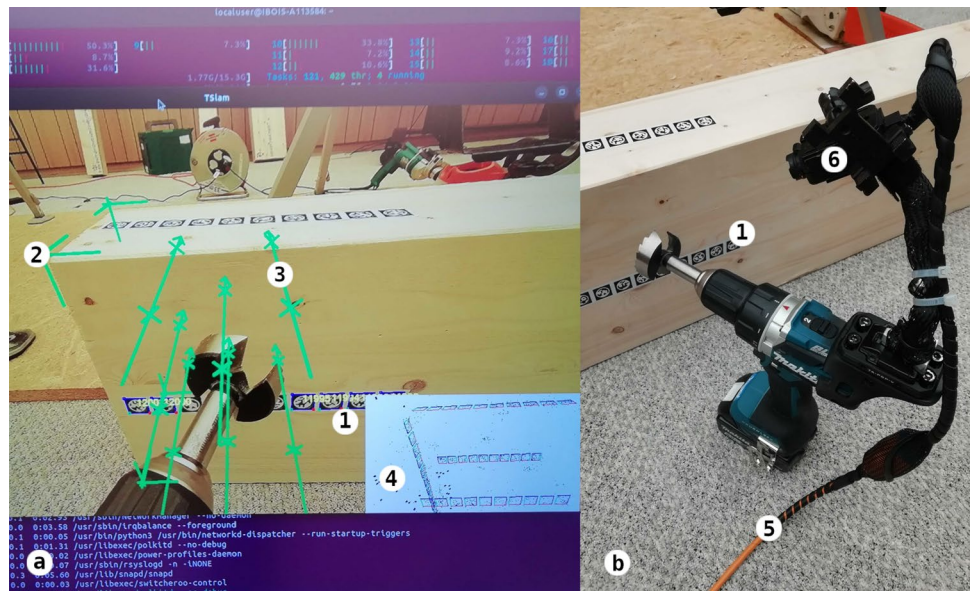


Fig. 15 Overview of the designed specimen timber beam that allocated all variants for the evaluation campaign and a photo of an evaluated specimen: **a1** in cyan joints; **a2** the clamps’ encumbrance; **a3**, **a5** drilling emplacements; **a4** the tags’ selected configurations; **b1** a joint fabricated during the fabrication session; and **b2** a pre-existing half-lap joint to test the TSLAM geometric solver

5. *Preparation time:* For each fabricated beam, we timed the TSLAM preparation procedure, including tag application and mapping.

The computed variation matrix for the evaluation resulted in a total of 20 beams to be fabricated, for a total of approximately 1344 sawing, drilling, and screwing operations.

3.2 Evaluation methodology for self-localization

As presented in Sect. 1.3.2, the creation of an ad-hoc fabrication-flavored data set was necessary for testing the accuracy of the proposed SLAM under construction conditions by evaluating the computed trajectories to its ground-truth counterpart. An operator is carrying out all the woodworking operations designed in Sect. 3.1 without following any augmented interface but rather traditional markings. This allowed us to restrain the scope of the current evaluation uniquely to the camera’s self-localization without considering any AR interface or computed guidance influence on the operator’s behavior. The data set is publicly available Settimi et al. (2023) and contains the recorded videos at 30 fps for each frame, its associated camera’s pose ground-truth data, and additional labels for all the carried fabrication sessions.

The Optitrack® system, composed of six cameras (Fig. 17) of type Flex13 (Optitrack 2023) is instrumental in obtaining the ground-truth trajectory during fabrication. The chosen outside—in tracking system allows submillimetric positional tracking of any reflective beacons in the scene with a refresh rate close to 160 fps.

If preregistered in groups, the markers permit the tracking of the pose of any given rigid body in the capture volume, which in our case is the one from the monocular RGB sensor mounted on each tool (Fig. 18b). The on-board sensor is equipped with six reflective beacons to ensure the continuity of the ground-truth signal even in the event of visual occlusions (e.g., the operator encumbering the infrared camera’s

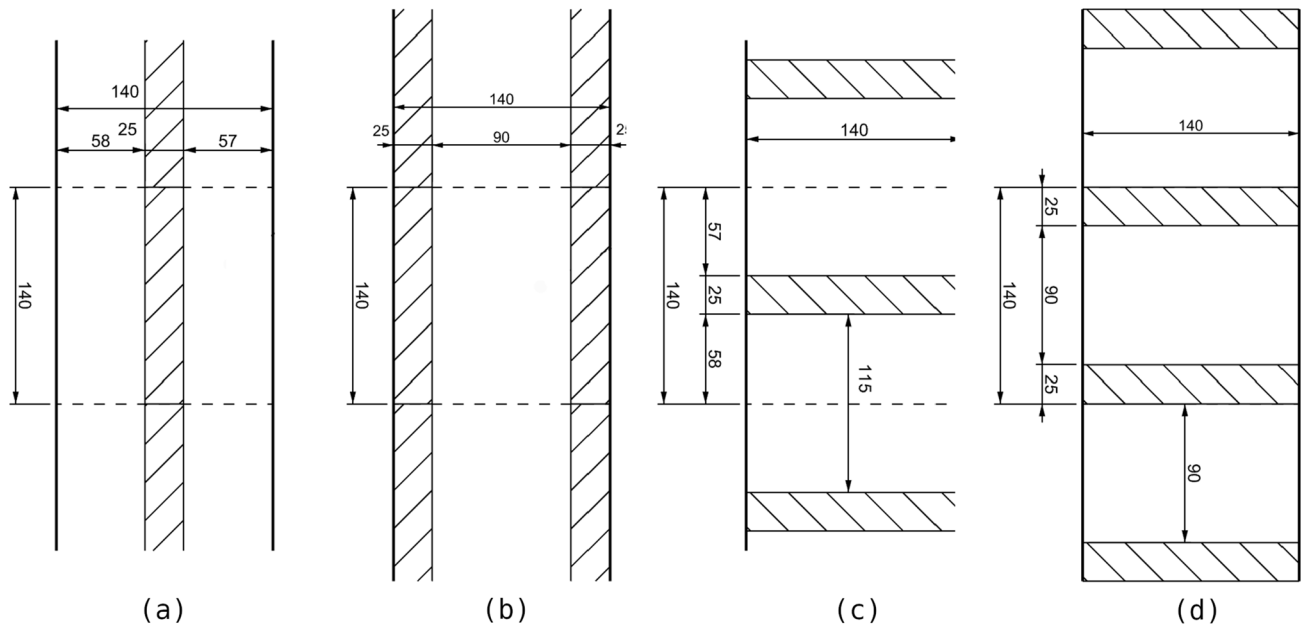


Fig. 16 Top view of the four selected tag configurations indicated by the hatched surface: **a** low density and stripe layout; **b** high density and stripe layout; **c** low density and ring layout; and **d** high density

and ring layout. In the low densities, the ratio of occupied timber by the stickers was 17%, whereas the higher density version presented a 35% coverage

view, timber chips, and false metal detection) and vibrations from the tool.

Following the mapping of the timber element, the TSLAM data were stored and employed later at inference time during the monitored fabrication sequence. At this stage, the monitoring system's signal was captured for each of the 30 frames per second provided by the camera's feed. The frame's timestamps, together with the RGB image, were stored as a new entry in the data set. TSLAM was benchmarked on the entire data set on a headless server running on Ubuntu 20.04.5 LTS with an AMD Ryzen 9 5950X as the main processor unit. The computation of the estimated pose based on the total 12.3 h of video footage took approximately 72 h to complete. The developed pipeline can be reproduced and found in the public repository of the project Settini and Yang (2023).

Once all of TSLAM's estimated poses (P_{est}) were obtained, the next step was to regulate both trajectories. Nonetheless, due to the object-centric nature of TSLAM, P_{est} were expressed in the timber's frame of reference (Fig. 19a), whereas ground-truth poses (P_{gt}) were referenced in the global world frame system (Fig. 19b). Given the constant manipulation of the timber piece during the recording, it was impossible to compute a global registration of the two trajectories as they were. Despite a possible solution to record the rigid body poses of the timber in the ground-truth data and later express P_{gt} in its reference frame, early tests indicated that this procedure causes additional errors.

Hence, we proposed evaluating each subtrajectory framed in the timeframe of a single operation (Fig. 19c). To accomplish this, we conducted a comprehensive assessment of each operation within the entire data set by manually annotating the commencement and conclusion points for every working fabrication. All of the couples of trajectories were individually registered by leveraging the Umeyama (1991) transformation in its rigid variant but with a custom filter to limit the candidate points as follows:

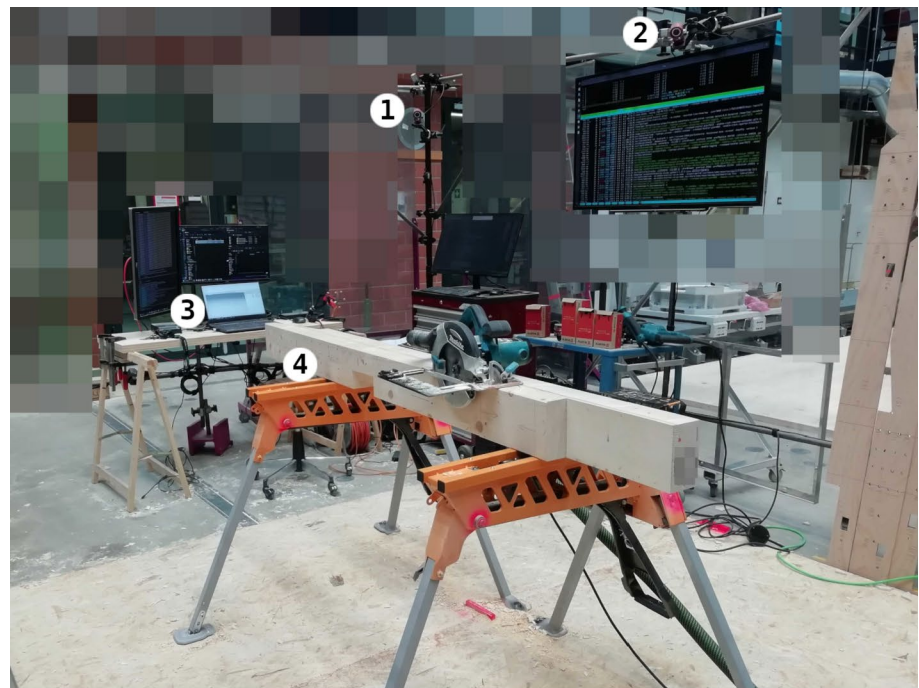
$$\begin{bmatrix} R & t \\ 0 & 1S \end{bmatrix} = \arg \min_R \sum_i \delta_{n_i > 3} \| \mathbf{p}_i - (R\mathbf{p}'_i + \mathbf{t}) \|^2 \quad (6)$$

$$\delta_i = \begin{cases} 1 & \text{if } n_i > 3 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

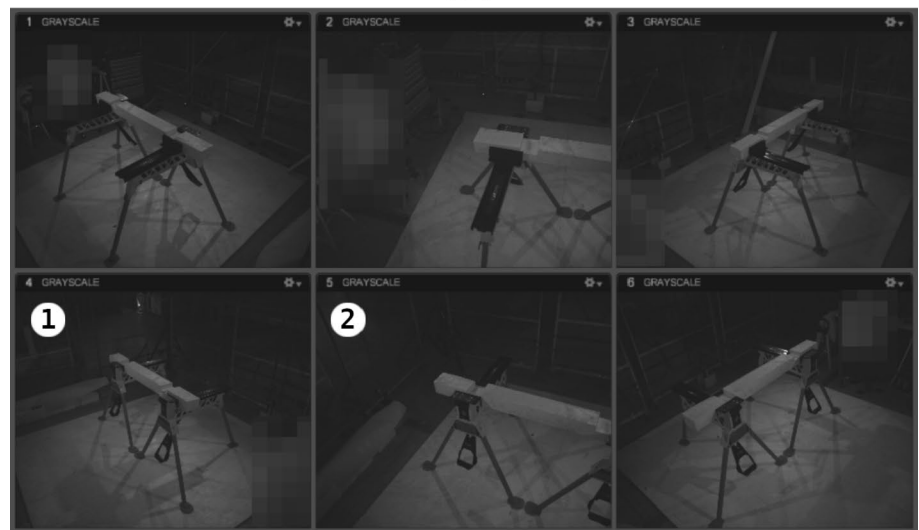
where $\delta_{n_i > 3}$ is the filtering of the employed points detected with at least three tags. This modification resulted in a more robust registration since the employed reference points were culled from potential false or inaccurate detections, such as one fiducial marker only. As a result, for each set of operations (Fig. 20), the subtrajectories were now aligned.

The computed metrics are relative for each subsequence, computed per frame, and applied only to the previously filtered points (see Eq. 6 and Fig. 21c). Besides the relative state-of-the-art trajectory error metrics (RE) as described by Zhang and Scaramuzza (2018) (Fig. 21), we also introduced an index that indicated the frequency of tag detection as

Fig. 17 **a** Capture of the evaluation setup: **a1**, **a2** two of the six outside—in tracking sensors; **a3** the computing station to which both the monocular sensor and the Optitrack[®]'s sensors are connected; and **a4** the $3 \times 3 \times 3$ m tracking volume. **b** Infrared views from each of the six cameras pointing to the fabrication area; the infrared sensors need to be calibrated and are sensitive to reflective metal parts that are preemptively painted or masked to avoid any false detection



(a)



(b)

well as a novel metric called the coverage fabrication value (CFV). The CFV gauges the signal coverage during different phases of a woodworking operation, specifically the beginning, middle, and end. It is expressed as a percentage of the outputted TSLAM's valid camera poses per frame over the quintiles of the timelapse of the woodworking operation. In essence, it provides a measure of how well the system captures and represents relevant signals across different stages of the woodworking operation. Through introducing this last parameter, we were interested not only in benchmarking the accuracy of individual components (position, rotation, and tag detection) but also in the temporal dynamics and

coverage of the system's output throughout the entire duration of the augmented fabrication operation.

3.3 Evaluation methodology for 3D reconstruction

We estimated the accuracy of each of TSLAM's reconstructed models, which also present different pre-existing joints according to the variation matrix (see Sect. 3.2), by comparing it with the ground-truth point cloud of the physical piece illustrated in Fig. 22. These referential data were obtained using a high-resolution, industrial-grade, hand-held 3D scanner (FARO Freestyle2[®]).

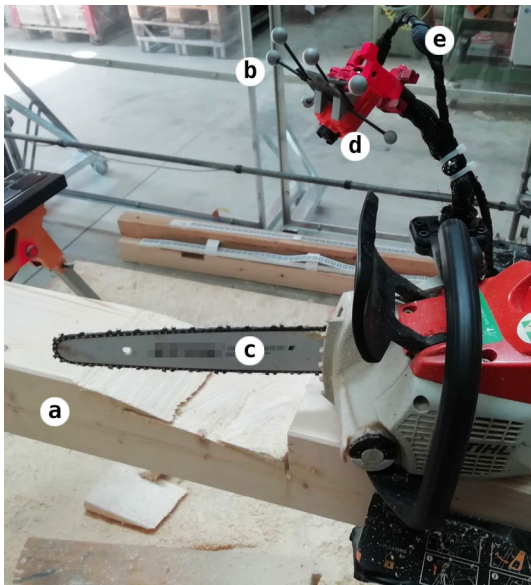


Fig. 18 Six reflective beacons **b** installed on the camera **d** allow the tracking of a rigid body in scenarios where at most four of the markers are occluded; **e** the camera tethered to the computing unit also collects the ground-truth data from the outside—in monitoring system; **c** current tool head; and **a** fabricated timber element

Initially, the reconstructed mesh was uniformly subsampled into a point cloud of 100,000 points. Next, a manual alignment was performed to coarsely align the two point clouds before running an iterative closest point (ICP) registration Rusinkiewicz and Levoy (2001). This method automatically applied a rigid transformation to the source point cloud to minimize the error w.r.t. the target point cloud. Once aligned, the point cloud underwent a 5 mm voxel downsampling by aggregating the constituent points into a singular point for every occupied voxel. For each point in the reconstructed point cloud, denoted as p_r , its nearest counterpart in the ground-truth point cloud, p_{gt} , was identified. The 3D Euclidean distance between these points was then calculated (Fig. 23), and this distance was averaged across all points to derive the error. Thus, the reconstruction error, represented by E_r , can be expressed as follows:

$$E_r = \frac{1}{N} \sum_{i=1}^N \text{Dist}(p_r, p_{gt}), \tag{8}$$

where N is the number of points in the reconstructed point cloud after voxel downsampling.

3.4 Results

To identify the optimal tag layout and density to adopt for TSLAM, we regrouped the computed results into four

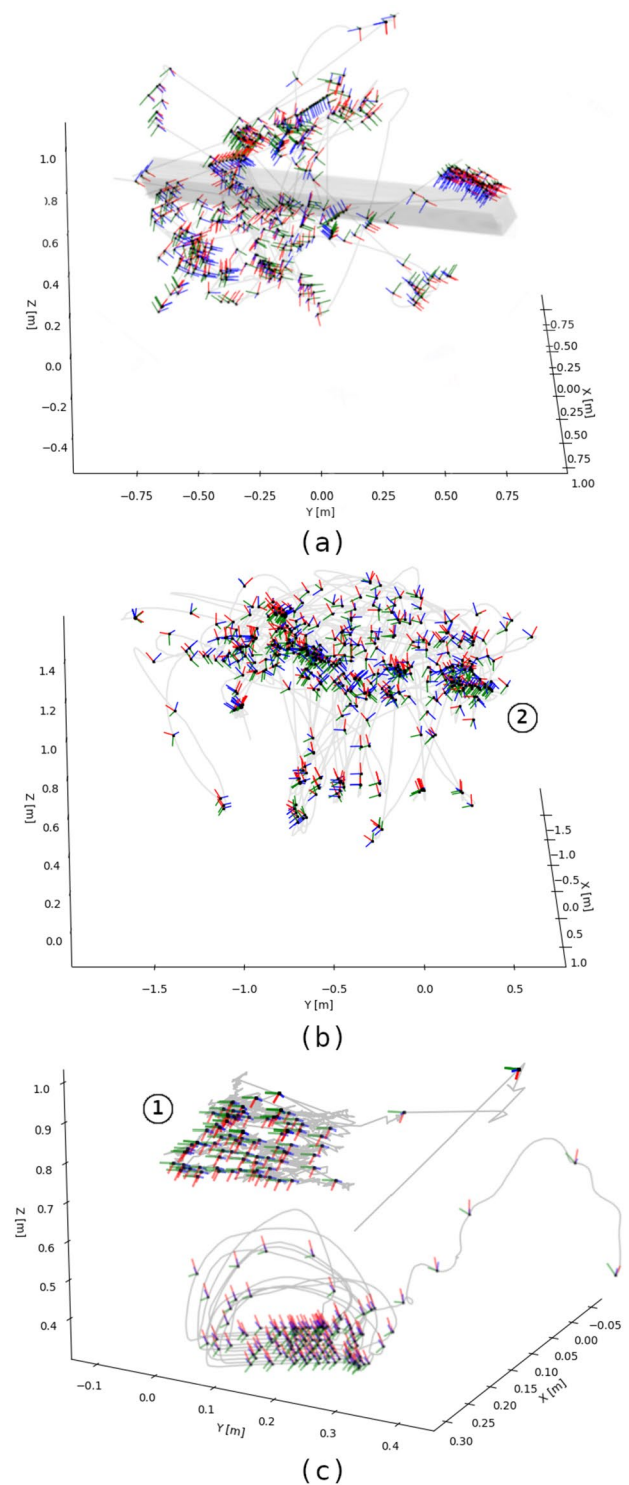


Fig. 19 **a** This graph represents the estimated TSLAM trajectory. To note that the output is object-centric and expressed to the tag map (the timber piece is represented in grey here), **b** reports the ground-truth trajectory referenced in world coordinates. **c** This graph shows how single operations' trajectories (e.g., circular sawing of a half-lap joint) can be extracted and eventually compared

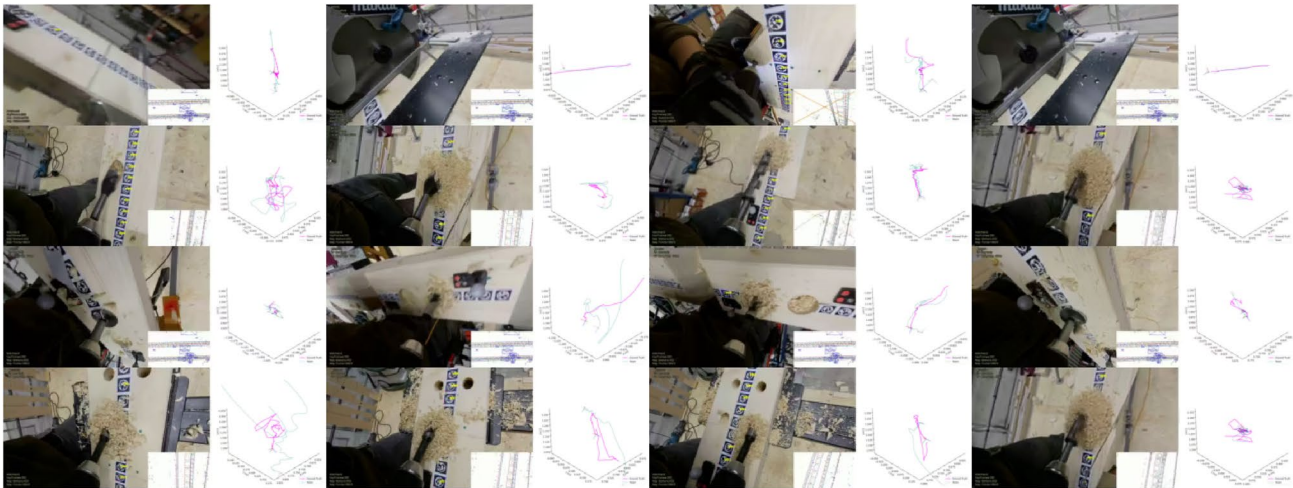


Fig. 20 Overview of some of the evaluated TSLAM trajectories of the total 1344 woodworking operations

categories following the evaluated combinations of tag layouts and densities (see Fig. 16). First, the overview, taken together, highlights positional (Fig. 24a) and rotational (Fig. 24b) error means of 1 mm and 0.1° (see Table 1) with minor differences between tags' distribution categories. For the average tag detection score (Fig. 24c) only the high-density stripe distribution performed substantially better, especially compared with the ring alternative. In addition, the recorded data concerning the timing of TSLAM's preparation prior to fabrication indicated that the single low-density stripe could be set up approximately twice as fast as manual marking and equally as fast as using templates (Fig. 25).

TSLAM's capability to mesh the object is dependent on the number of joints and the correct positioning of tags. Despite higher densities for the stripe layout (Fig. 26b), the same cannot be said for the ring (Fig. 26d). The most relevant factor for the success of the reconstruction algorithm is the tags' coverage of timber portions rather than their densities. The least performance was the single-stripe layout (Fig. 26a), which is a configuration that can perform a reconstruction on intact beams as well as ones with pre-existing joints; however, it failed at meshing in the other scenarios.

Hence, we observed that the low-density stripe is the configuration that strikes the optimal tradeoff between self-localization accuracy and logistical performance. Nevertheless, its 3D-reconstruction capabilities, given the current state of the software, are limited to off-shelf beams but without predictable limitations in dimensions.

When we limited our analysis to results that correspond to the candidate tags' configuration, we obtained excellent and consistent submillimetric camera pose estimation for the majority of circumstances (see Table 2).

We could also confirm that, besides woodworking operations involving the saber saw, TSLAM is capable of performing self-localization on a monocular sensor in all the fabrication scenarios contained in the data set with a positional and angular mean error of 1 mm and 0.3° for all tools (Fig. 27a, b). In addition, the graph 27c demonstrates how the mean number of detected tags for all tools is approximately 6. This affirms that adopting tag redundancy serves as an effective strategy when dealing with noise, such as instances where chipping and sawdust partially obscure the tag stripes but rarely entirely. TSLAM's tracking for the saber saw scored a mean value of 1.58 mm with consistently higher variance. This can be explained by the high vibrations of the motor generated by the oscillating movement of the blade, which are absent in rotary sawing devices like circular saws. This was confirmed by the novel metric that we introduced for the coverage value, namely, the CFV (Fig. 28). In addition, we noted that drilling operations with self-feeding bits experienced a signal of lesser quality than longer drill bits, which is because of the significantly close distance to the timber surface at the end of the operations.

In the following chapter, we expose the current limitations and foreseeable improvements for TSLAM.

4 Current limitations and improvements

The efficacy of the system is contingent on precise camera calibration, particularly concerning the undistortion matrix. Failure to achieve accuracy in this calibration may yield unsatisfactory outcomes, characterized by a warped

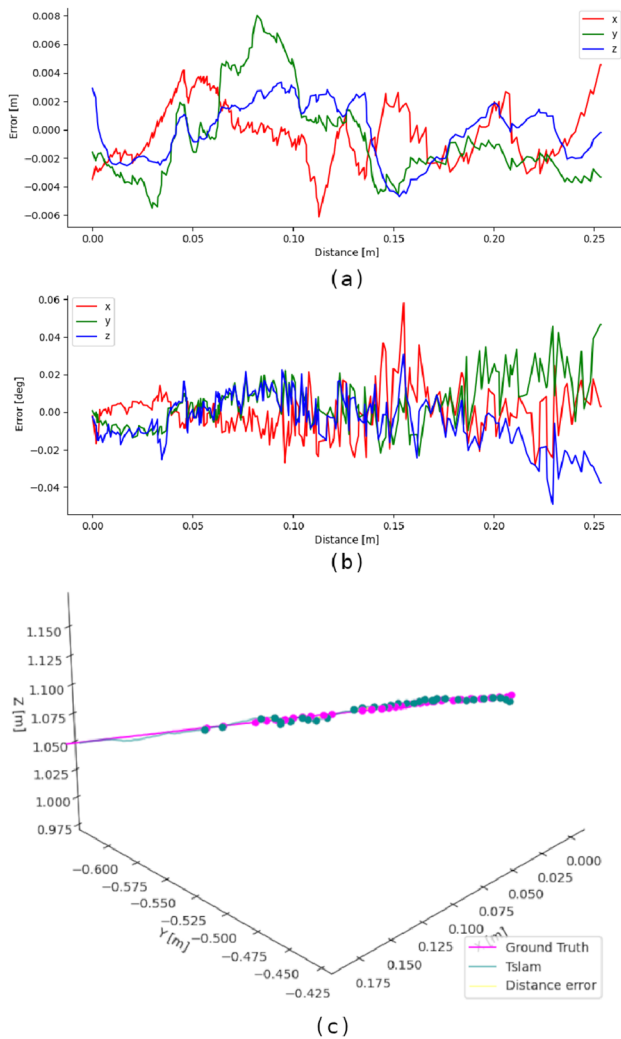


Fig. 21 **a** Positional and **b** rotational drift were the two major components evaluated for each set of subsequences. They represent the error distance from the ground-truth trajectory of TSLAM. In **c**, we depict the candidate points used in the transformation and evaluation



Fig. 22 Raw point cloud of the scanned preprocessed timber piece to compare with the mesh reconstructed by TSLAM. The object was scanned twice on both sides and then aligned and combined

appearance in the scanned output. This constraint imposes limitations on the selection of suitable cameras and may necessitate additional efforts to acquire the optimal camera matrix. Furthermore, in the use of a fisheye camera, the accurate undistortion of objects is compromised if they are situated near the camera or at its periphery. Consequently, in the context of timber mapping with a fisheye camera, adherence to a consistent distance and perpendicular orientation is imperative. This practice ensures that the target object remains as centrally positioned as possible, thereby guaranteeing optimal results.

When we tested the timber mapping, there were instances where the map became distorted, which required us to terminate the process and restart it. We believe that the primary cause of this problem was the ambiguity in the marker’s pose. While the system is designed to address ambiguous marker poses by minimizing the reprojection error across all frames, it sometimes fails to achieve the expected outcome. Regrettably, no known algorithm can perfectly resolve this challenge. Fortunately, however, this problem does not always arise, and the outcome can be easily detected.

Finally, in augmented fabrications, it is not uncommon for the user to want to inspect the overhaul model overlay from a further distance for a broader overview. This functionality is currently possible in TSLAM to the extent that the sensor can detect tags. Since TSLAM is voluntarily calibrated and dedicated to close-range camera self-localizations, it allows the user to zoom out only to approximately 60 cm. The integration of dual functioning with feature-point detection during fabrication could enhance the visualization distance from the beam; however, it would require the masking of tool heads that obstruct the scene constantly.

5 Conclusions

In conclusion, we have presented a solution to the challenges inherent in AR self-localization for augmented subtractive woodworking tasks through the introduction of TSLAM. TSLAM stands as a submillimetrically accurate navigation system that employs a monocular hybrid object-oriented SLAM C++ framework on top of ad-hoc features tailored to fabrication scenarios. This innovative system demonstrated its capabilities by reconstructing a precise 3D model of common timber beams’ sections and seamlessly overlaying it onto its physical counterpart through an AR interface,

Fig. 23 Overview of a selection of the reconstructed and evaluated models: **a–d, h** represent the successfully reconstructed model with limited or anodyne distance error to the ground truth, while **e, g, i** illustrate models where our method failed due to the absence of tags

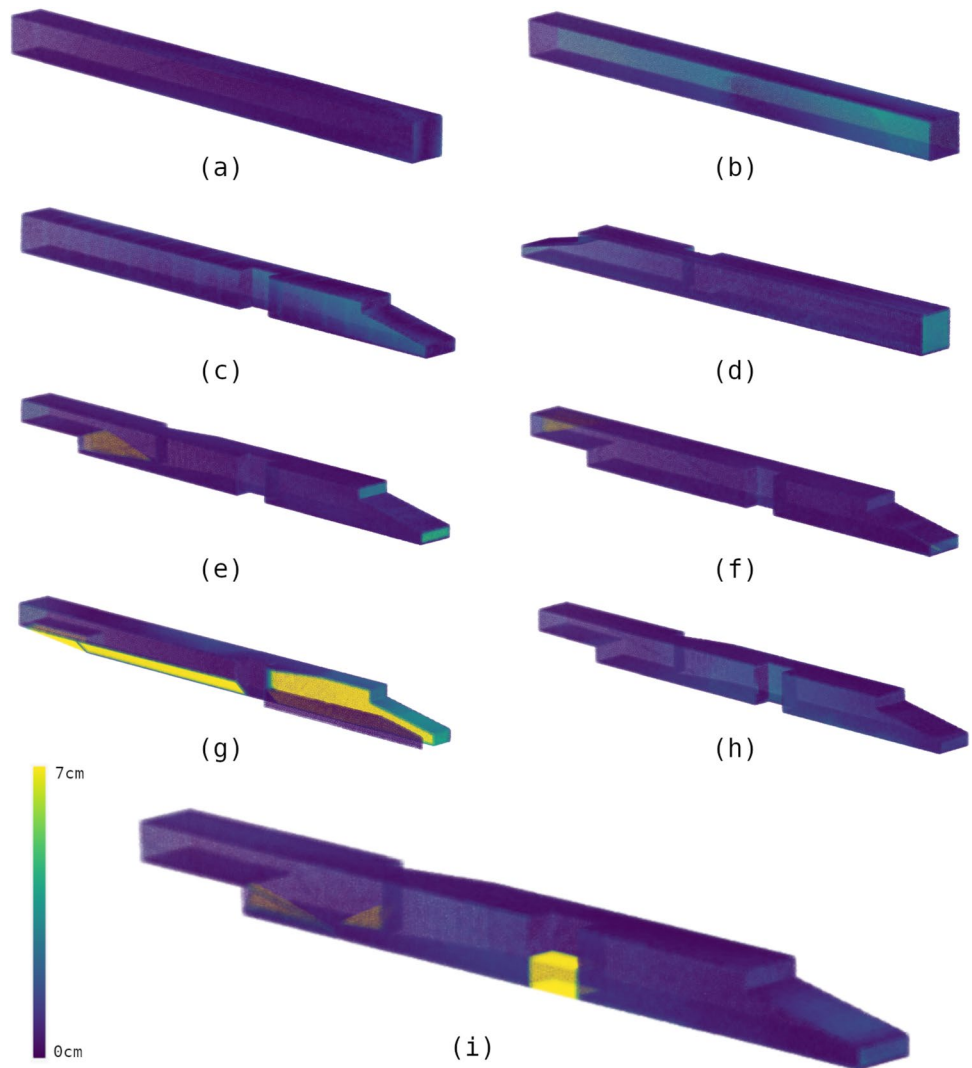


Table 1 Table resuming the error metrics for all 1344 operations of the data sets grouped by category of tags’ layers and densities where *P* is the positional error, *R* is the angular error, and *T* is the average detection of tags

	Low density (-)	High density (-)
Stripe layout	<i>P</i> : 0.11 ± 0.73 (mm)	<i>P</i> : 0.03 ± 0.64 (mm)
	<i>R</i> : 0.01 ± 0.01 (°)	<i>R</i> : 0.01 ± 0.01 (°)
	<i>T</i> : 7 ± 2 (-)	<i>T</i> : 7 ± 2 (-)
Ring layout	<i>P</i> : 0.03 ± 0.52 (mm)	<i>P</i> : 0.33 ± 0.97 (mm)
	<i>R</i> : 0.01 ± 0.01 (°)	<i>R</i> : 0.01 ± 0.01 (°)
	<i>T</i> : 9 ± 5 (-)	<i>R</i> : 15 ± 2 (-)

*The errors are represented in mean ± standard deviation

reaching millimetric accuracy. The robust performance of TSLAM was validated under real-life construction conditions through a dedicated experimental campaign that we tailored for this study. Said campaign showcased TSLAM’s effectiveness in diverse woodworking scenarios involving various tools commonly used in carpentry. We highlighted the correct use and distribution of tags for striking the optimal tradeoff between fabrication and pose estimation accuracy.

An adapted SLAM represents the very first foundational requirement for any AR application. TSLAM represents a reliable, robust, and precise approach to effectively performing AR-guided operations with manual tools in timber

Fig. 24 Violin plot of the summary metrics grouped by tag density and distribution. The values were computed across all tools and beams, and the green segment indicates the mean: **a** positional error distribution, **b** rotational error distribution, and **c** average of tag detection. The elongated flaps might be explained by the presence of noise in the form of pose detection outliers for graphs **a, b**

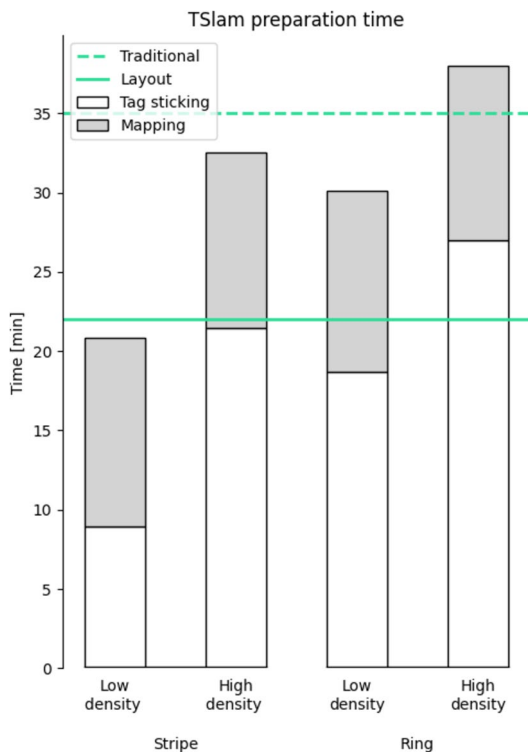
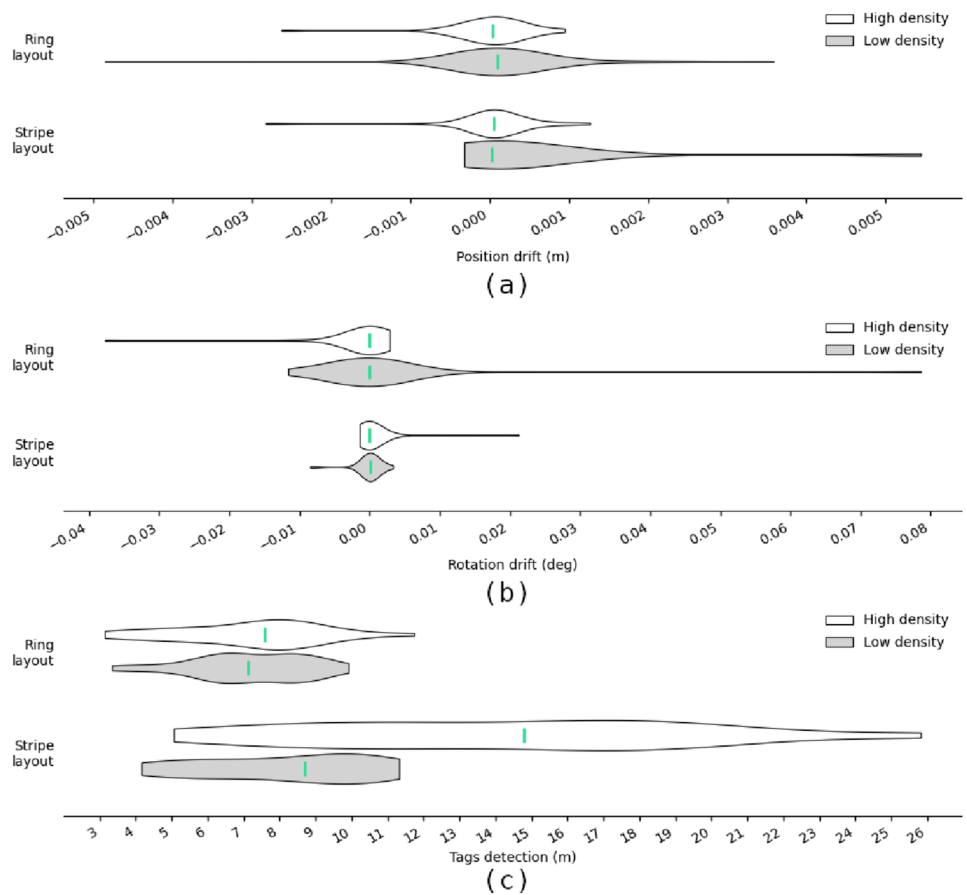


Fig. 25 Beam’s preparation and mapping for the stripe with low-density tag distribution is indicated to require half the time of hand marking and an equivalent time to a pre-fab preparation with templates

carpentry. With the proposed system, the piece can be manipulated at any time, under all light conditions, and with noise; moreover, its shape can be modified throughout the process, but the camera is constantly localized. We also demonstrated how the preparatory phase consumes less time than traditional hand-marking and tracing on the timber piece.

Nevertheless, TSLAM does not offer other equally important features like object detection, which would eventually complete a fully operational and state-of-the-art AR fabrication system for subtractive tasks. This is the subject of TSLAM’s ongoing development. Coupled with the system’s robust camera localization and timber reconstruction, a fully operational and reliable AR integration of tool head detection and localization will be proposed. In addition, we intend to expand the variety of geometries and lengths of sections that TSLAM can effectively reconstruct, such as round wood, curved elements, and longer glulam beams that are widely diffused in timber construction.

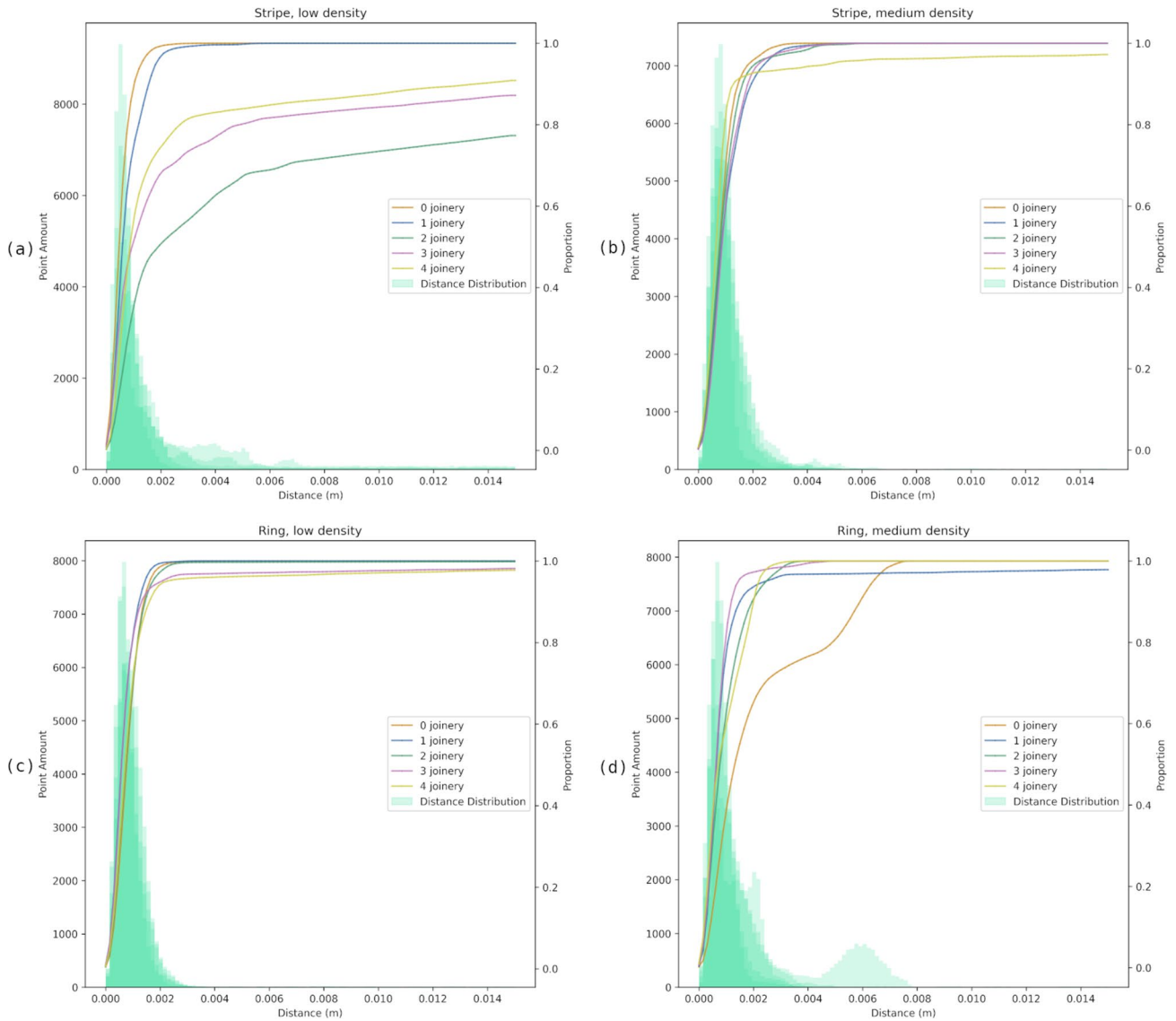


Fig. 26 Graphs that present the reconstruction accuracy for the four categories of tag distribution: **a** low-density stripe, **b** high-density stripe, **c** low-density ring, and **d** high-density layout

Table 2 Table resuming the error metrics for the one fabrication session on a beam with low density and single-stripe layout as tags' distribution

Tool name (-)	Number of operations (-)	Mean position error* (mm)	Mean orientation error* (°)	Mean tags detection* (-)	Mean coverage index (%)
Circular sawblade saber	32	0.15 ± 0.09	0.01 ± 0.01	3 ± 2	72.30
Sawblade drill	3	1.58 ± 8.54	0.01 ± 0.07	7 ± 7	21.57
Hing (ø50)	7	0.18 ± 4.32	0.01 ± 0.04	6 ± 5	68.75
Auger (ø20) drill	19	0.15 ± 3.77	0.07 ± 0.03	6 ± 4	93.12
Auger (ø25) drill	2	0.37 ± 0.61	0.04 ± 0.02	8 ± 6	98.51
Oblique (ø40) screw	2	0.22 ± 0.47	0.21 ± 0.11	8 ± 3	99.28
(120) screw	8	0.16 ± 1.87	0.18 ± 0.09	6 ± 4	95.23
(100) screw	8	0.29 ± 2.04	0.12 ± 0.06	7 ± 4	97.99
(80) screw	8	0.01 ± 3.16	0.21 ± 0.10	8 ± 3	97.77
(45)	8	0.01 ± 1.19	0.04 ± 0.02	6 ± 3	97.50

* The errors are represented in mean ± interquartile range (IQR = Q3 - Q1)

Fig. 27 Overview of the positional (a) and rotational (b) errors, as well as the tags' average detection (c) per tool type and for a regular beam with a low-density tag distribution. The high number of outliers in the representation can be interpreted as possible noise and minor drifting events that occurred during tracking. Means are indicated in green

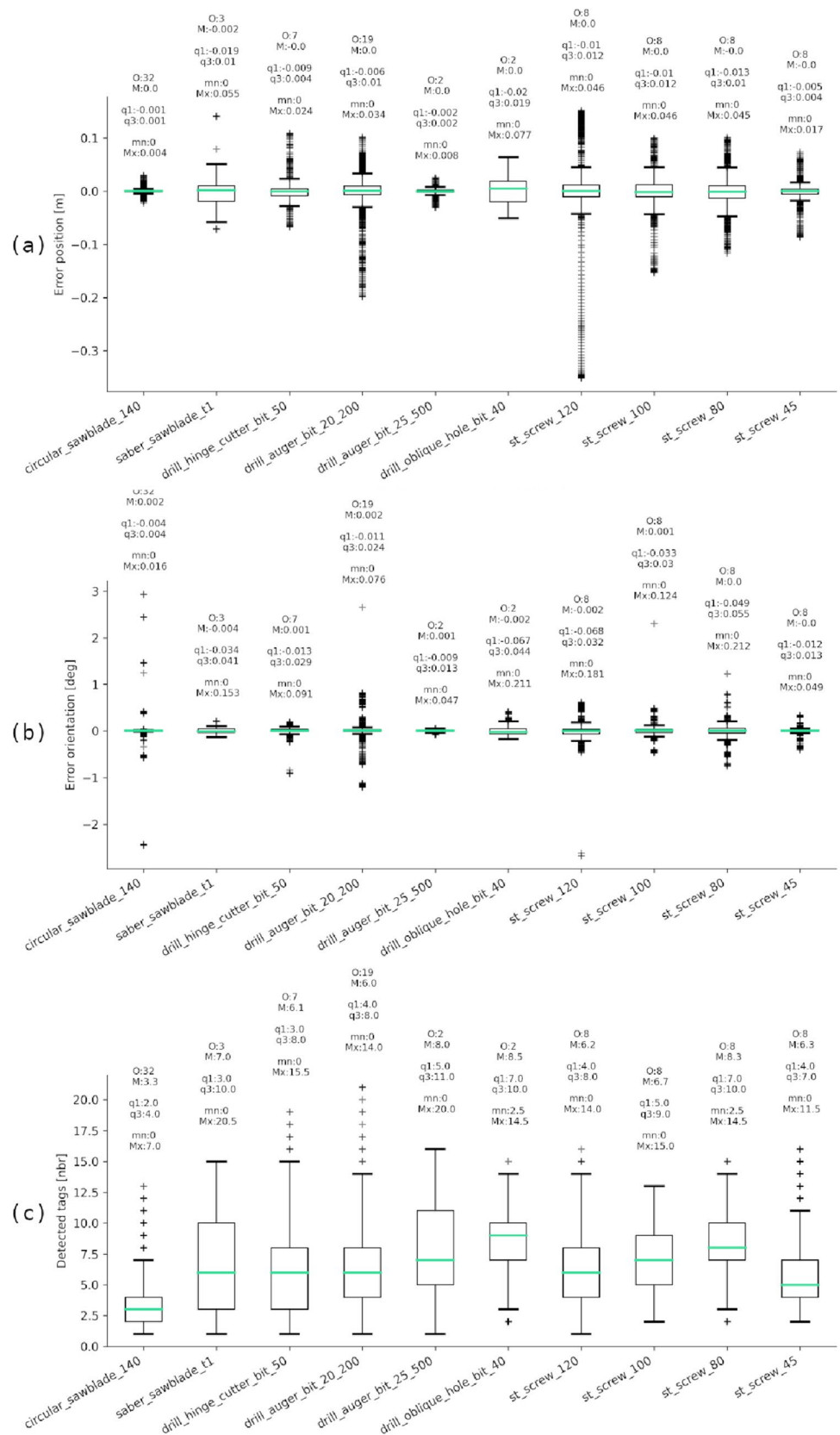
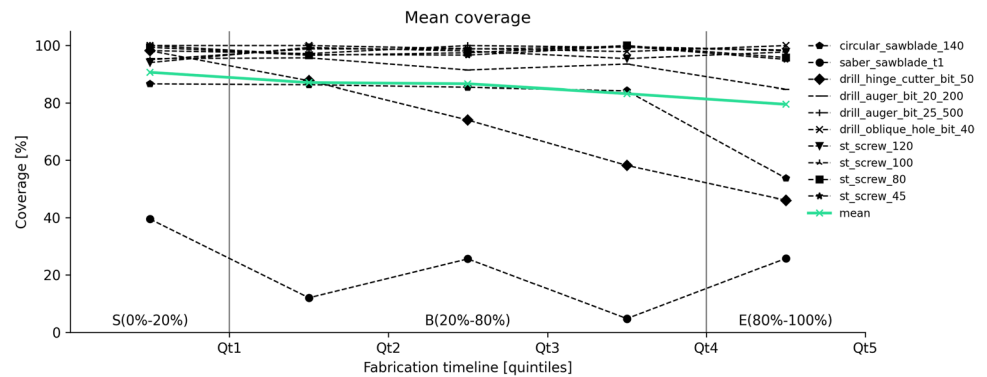


Fig. 28 Visualization of the coverage value. The coverage value is crucial for understanding TSLAM's capacity to provide a camera's pose feed at each phase of the fabrication sequence. The x axis indicates the timeframe for any given operation divided into quintiles, highlighting the start-, mid-, and end-phases



Funding Open access funding provided by EPFL Lausanne.

Data availability The source code of the software and its accompanying dataset used in this publication are publicly available. They can be accessed via <https://zenodo.org/records/10093230> for the source code and <https://zenodo.org/records/8377793> for the dataset. We believe in promoting transparency and reproducibility in research and encourage the community to utilize and build upon our work.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Adel AA (2020) Computational design for cooperative robotic assembly of nonstandard timber frame buildings. PhD thesis. <https://doi.org/10.3929/ETHZ-B-000439443>. <http://hdl.handle.net/20.500.11850/439443>
- Adel A (2023) Co-robotic assembly of nonstandard timber structures. <https://doi.org/10.7302/8675>. <http://deepblue.lib.umich.edu/handle/2027.42/178286>
- Adel A, Thoma A, Helmreich M, Gramazio F, Kohler M (2018) Design of robotically fabricated timber frame structures. In: Proceedings of the 38th annual conference of the association for computer aided design in architecture (ACADIA). ACADIA. <https://doi.org/10.52842/conf.acadia.2018.394>
- Alatise MB, Hancke GP (2020) A review on challenges of autonomous mobile robot and sensor fusion methods. *IEEE Access* 8:39830–39846. <https://doi.org/10.1109/access.2020.2975643>
- Ali I, Durmush A, Suominen O, Yli-Hietanen J, Peltonen S, Collin J, Gotchev A (2020) FinnForest dataset: a forest landscape for visual SLAM. *Robot Auton Syst* 132:103610. <https://doi.org/10.1016/j.robot.2020.103610>
- Barros AM, Michel M, Moline Y, Corre G, Carrel F (2022) A comprehensive survey of visual SLAM algorithms. *Robotics* 11(1):24. <https://doi.org/10.3390/robotics11010024>
- Benligiray B, Topal C, Akinlar C (2019) STag: a stable fiducial marker system. *Image Vis Comput* 89:158–169. <https://doi.org/10.1016/j.imavis.2019.06.007>
- Bergamasco F, Albarelli A, Rodola E, Torsello A (2011) RUNE-tag: a high accuracy fiducial marker with strong occlusion resilience. In: *CVPR 2011*. IEEE. <https://doi.org/10.1109/cvpr.2011.5995544>
- Bergamasco F, Albarelli A, Cosmo L, Rodola E, Torsello A (2016) An accurate and robust artificial marker based on cyclic codes. *IEEE Trans Pattern Anal Mach Intell* 38(12):2359–2373. <https://doi.org/10.1109/tpami.2016.2519024>
- Bloesch M, Omari S, Hutter M, Siegwart R (2015) Robust visual inertial odometry using a direct EKF-based approach. <https://doi.org/10.3929/ETHZ-A-010566547>. <http://hdl.handle.net/20.500.11850/155340>
- Bloesch M, Burri M, Omari S, Hutter M, Siegwart R (2017) Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback. *Int J Robot Res* 36(10):1053–1072. <https://doi.org/10.1177/0278364917728574>
- Bouthermy P (1989) A maximum likelihood framework for determining moving edges. *IEEE Trans Pattern Anal Mach Intell* 11(5):499–511. <https://doi.org/10.1109/34.24782>
- Bruno HMS, Colombini EL (2021) LIFT-SLAM: a deep-learning feature-based monocular visual SLAM method. *Neurocomputing* 455:97–110. <https://doi.org/10.1016/j.neucom.2021.05.027>
- Burri M, Nikolic J, Gohl P, Schneider T, Rehder J, Omari S, Achtelik MW, Siegwart R (2016) The EuRoC micro aerial vehicle datasets. *Int J Robot Res* 35(10):1157–1163. <https://doi.org/10.1177/0278364915620033>
- Campos C, Elvira R, Rodriguez JGG, Montiel JMM, Tardos JD (2021) ORB-SLAM3: an accurate open-source library for visual, visual inertial, and multimap SLAM. *IEEE Trans Robot* 37(6):1874–1890. <https://doi.org/10.1109/tro.2021.3075644>
- Carlevaris-Bianco N, Ushani AK, Eustice RM (2015) University of Michigan north campus long-term vision and lidar dataset. *Int J Robot Res* 35(9):1023–1035. <https://doi.org/10.1177/0278364915614638>
- Chen W, Shang G, Ji A, Zhou C, Wang X, Xu C, Li Z, Hu K (2022) An overview on visual SLAM: from tradition to semantic. *Remote Sens* 14(13):3010. <https://doi.org/10.3390/rs14133010>
- Cioffi G, Cieslewski T, Scaramuzza D (2022) Continuous-time vs. discrete-time vision-based slam: a comparative study. <https://doi.org/10.48550/ARXIV.2202.08894>. <https://arxiv.org/abs/2202.08894>
- 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. <https://doi.org/10.1109/tpami.2007.1049>
- Delmerico J, Cieslewski T, Rebecq H, Faessler M, Scaramuzza D (2019) Are we ready for autonomous drone racing? The UZH-FPV drone racing dataset. In: 2019 international conference on robotics and automation (ICRA). IEEE. <https://doi.org/10.1109/icra.2019.8793887>
- Dörfler K, Sandy T, Giftthaler M, Gramazio F, Kohler M, Buchli J (2016) Mobile robotic brickwork. In: *Robotic fabrication in architecture*,


- art and design 2016. Springer, London, pp 204–217. https://doi.org/10.1007/978-3-319-26378-6_15
- Dörfler K, Hack N, Sandy T, Gifthalder M, Lussi M, Walzer AN, Buchli J, Gramazio F, Kohler M (2019) Mobile robotic fabrication beyond factory conditions: case study Mesh Mould wall of the DFAB HOUSE. *Constr Robot* 3(1–4):53–67. <https://doi.org/10.1007/s41693-019-00020-w>
- Dörfler K, Dielemans G, Lachmayer L, Recker T, Raatz A, Lowke D, Gerke M (2022) Additive manufacturing using mobile robots: opportunities and challenges for building construction. *Cem Concr Res* 158:106772. <https://doi.org/10.1016/j.cemconres.2022.106772>
- Dube R, Gawel A, Sommer H, Nieto J, Siegwart R, Cadena C (2017) An online multi-robot SLAM system for 3D LiDARs. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE. <https://doi.org/10.1109/iros.2017.8202268>
- Engel J, Schöps T, Cremers D (2014) LSD-SLAM: large-scale direct monocular SLAM. In: *Computer vision—ECCV 2014*. Springer, London, pp 834–849. https://doi.org/10.1007/978-3-319-10605-2_54
- Engel J, Stuckler J, Cremers D (2015) Large-scale direct SLAM with stereo cameras. In: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE. <https://doi.org/10.1109/iros.2015.7353631>
- Engel J, Usenko V, Cremers D (2016) A photometrically calibrated benchmark for monocular visual odometry. <https://doi.org/10.48550/ARXIV.1607.02555>. <https://arxiv.org/abs/1607.02555>
- Eversmann P, Gramazio F, Kohler M (2017) Robotic prefabrication of timber structures: towards automated large-scale spatial assembly. *Constr Robot* 1(1–4):49–60. <https://doi.org/10.1007/s41693-017-0006-2>
- Fiala M (2005) ARTag, a fiducial marker system using digital techniques. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). IEEE. <https://doi.org/10.1109/cvpr.2005.74>
- Forster C, Zhang Z, Gassner M, Werlberger M, Scaramuzza D (2017) SVO: semidirect visual odometry for monocular and multicamera systems. *IEEE Trans Robot* 33(2):249–265. <https://doi.org/10.1109/tro.2016.2623335>
- Galvez-López D, Tardos JD (2012) Bags of binary words for fast place recognition in image sequences. *IEEE Trans Robot* 28(5):1188–1197. <https://doi.org/10.1109/tro.2012.2197158>
- Gao X, Wang R, Demmel N, Cremers D (2018) LDSO: direct sparse odometry with loop closure. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE. <https://doi.org/10.1109/iros.2018.8593376>
- Gao L, Liang Y, Yang J, Wu S, Wang C, Chen J, Kneip L (2022) VECtor: a versatile event-centric benchmark for multi-sensor SLAM. *IEEE Robot Autom Lett* 7(3):8217–8224. <https://doi.org/10.1109/lra.2022.3186770>
- Garrido-Jurado S, Muñoz-Salinas R, Madrid-Cuevas F, Marín-Jiménez M (2014) Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recogn* 47(6):2280–2292. <https://doi.org/10.1016/j.patcog.2014.01.005>
- Garrido-Jurado S, Muñoz-Salinas R, Madrid-Cuevas F, Medina-Carnicer R (2016) Generation of fiducial marker dictionaries using mixed integer linear programming. *Pattern Recogn* 51:481–491. <https://doi.org/10.1016/j.patcog.2015.09.023>
- Geiger A, Lenz P, Urtasun R (2012) Are we ready for autonomous driving? The kitti vision benchmark suite. In: 2012 IEEE conference on computer vision and pattern recognition, pp 3354–3361. <https://doi.org/10.1109/CVPR.2012.6248074>
- Gifthalder M, Sandy T, Dörfler K, Brooks I, Buckingham M, Rey G, Kohler M, Gramazio F, Buchli J (2017) Mobile robotic fabrication at 1:1 scale: the in situ fabricator. *Constr Robot* 1(1–4):3–14. <https://doi.org/10.1007/s41693-017-0003-5>
- Golodetz S, Cavallari T, Lord NA, Prisacariu VA, Murray DW, Torr PHS (2018) Collaborative large-scale dense 3D reconstruction with online inter-agent pose optimisation. <https://doi.org/10.48550/ARXIV.1801.08361>. <https://arxiv.org/abs/1801.08361>
- Handa A, Whelan T, McDonald J, Davison AJ (2014) A benchmark for RGB-d visual odometry, 3D reconstruction and SLAM. In: 2014 IEEE international conference on robotics and automation (ICRA). IEEE. <https://doi.org/10.1109/icra.2014.6907054>
- Helmberger M, Morin K, Berner B, Kumar N, Cioffi G, Scaramuzza D (2022) The hilti SLAM challenge dataset. *IEEE Robot Autom Lett* 7(3):7518–7525. <https://doi.org/10.1109/lra.2022.3183759>
- Hughes R, Osterlund T, Larsen NM (2021) Integrated design-for-manufacturing and AR-aided-assembly workflows for lightweight reciprocal frame timber structures. *Constr Robot* 5(2):147–157. <https://doi.org/10.1007/s41693-020-00048-3>
- Itseez (2015) Open source computer vision library. https://github.com/egonSchiele/OpenCV/blob/master/tests/cv/src/affine3d_estimator.cpp
- Jahn G, Newnham C, van den Berg N, Iraheta M, Wells J (2019) Holographic construction. In: *Impact: design with all senses*. Springer, London, pp 314–324. https://doi.org/10.1007/978-3-030-29829-6_25
- Johns RL, Wermelinger M, Mascaro R, Jud D, Gramazio F, Kohler M, Chli M, Hutter M (2020) Autonomous dry stone. *Constr Robot* 4(3–4):127–140. <https://doi.org/10.1007/s41693-020-00037-6>
- Jud D, Kerschler S, Wermelinger M, Jelavic E, Egli P, Leemann P, Hottiger G, Hutter M (2021) HEAP—the autonomous walking excavator. *Autom Constr* 129:103783. <https://doi.org/10.1016/j.autcon.2021.103783>
- Kalaitzakis M, Cain B, Carroll S, Ambrosi A, Whitehead C, Vitzilaios N (2021) Fiducial markers for pose estimation. *J Intell Robot Syst*. <https://doi.org/10.1007/s10846-020-01307-9>
- Kendall A, Grimes M, Cipolla R (2015) Posenet: a convolutional network for real-time 6-DOF camera relocalization. <https://doi.org/10.48550/ARXIV.1505.07427>. <https://arxiv.org/abs/1505.07427>
- Khairuddin AR, Talib MS, Haron H (2015) Review on simultaneous localization and mapping (SLAM). In: 2015 IEEE international conference on control system, computing and engineering (ICCSCE). IEEE. <https://doi.org/10.1109/iccsce.2015.7482163>
- Kim G, Park YS, Cho Y, Jeong J, Kim A (2020) MulRan: multimodal range dataset for urban place recognition. In: 2020 IEEE international conference on robotics and automation (ICRA). IEEE. <https://doi.org/10.1109/icra40945.2020.9197298>
- Klein G, Murray D (2007) Parallel tracking and mapping for small AR workspaces. In: 2007 6th IEEE and ACM international symposium on mixed and augmented reality. IEEE. <https://doi.org/10.1109/ismar.2007.4538852>
- Klenk S, Chui J, Demmel N, Cremers D (2021) Tum-vie: the tum stereo visual-inertial event dataset. <https://doi.org/10.48550/ARXIV.2108.07329>. <https://arxiv.org/abs/2108.07329>
- Klopschitz M, Schmalstieg D (2007) Automatic reconstruction of wide-area fiducial marker models. In: 2007 6th IEEE and ACM international symposium on mixed and augmented reality. IEEE. <https://doi.org/10.1109/ismar.2007.4538828>
- Kyaw AH, Xu AH, Jahn G, van den Berg N, Newnham C, Zivkovic S (2023) Augmented reality for high precision fabrication of glued laminated timber beams. *Autom Constr* 152:104912. <https://doi.org/10.1016/j.autcon.2023.104912>
- Larsson M, Yoshida H, Igarashi T (2019) Human-in-the-loop fabrication of 3D surfaces with natural tree branches. In: *Proceedings of the ACM symposium on computational fabrication*. ACM. <https://doi.org/10.1145/3328939.3329000>
- Leutenegger S, Lynen S, Bosse M, Siegwart R, Furgale P (2014) Keyframe-based visual-inertial odometry using nonlinear optimization. *Int J Robot Res* 34(3):314–334. <https://doi.org/10.1177/0278364914554813>

- Li Y, Brasch N, Wang Y, Navab N, Tombari F (2020) Structure-SLAM: low-drift monocular SLAM in indoor environments. *IEEE Robot Autom Lett* 5(4):6583–6590. <https://doi.org/10.1109/lra.2020.3015456>
- Li R, Wang S, Gu D (2021) DeepSLAM: a robust monocular SLAM system with unsupervised deep learning. *IEEE Trans Ind Electron* 68(4):3577–3587. <https://doi.org/10.1109/tie.2020.2982096>
- Li S, Zhang D, Xian Y, Li B, Zhang T, Zhong C (2022) Overview of deep learning application on visual SLAM. *Displays* 74:102298. <https://doi.org/10.1016/j.displa.2022.102298>
- Liao Y, Xie J, Geiger A (2022) KITTI-360: a novel dataset and benchmarks for urban scene understanding in 2D and 3D. *IEEE Trans Pattern Anal Mach Intell*. <https://doi.org/10.1109/tpami.2022.3179507>
- Lowe D (1991) Fitting parameterized three-dimensional models to images. *IEEE Trans Pattern Anal Mach Intell* 13(5):441–450. <https://doi.org/10.1109/34.134043>
- Lussi M, Sandy T, Dorfler K, Hack N, Gramazio F, Kohler M, Buchli J (2018) Accurate and adaptive in situ fabrication of an undulated wall using an on-board visual sensing system. In: 2018 IEEE international conference on robotics and automation (ICRA). IEEE. <https://doi.org/10.1109/icra.2018.8460480>
- Lynen S, Achtelik MW, Weiss S, Chli M, Siegwart R (2013) A robust and modular multi-sensor fusion approach applied to MAV navigation. In: 2013 IEEE/RSJ international conference on intelligent robots and systems. IEEE. <https://doi.org/10.1109/iros.2013.6696917>
- Maddern W, Pascoe G, Linegar C, Newman P (2016) 1 year, 1000 km: the Oxford RobotCar dataset. *Int J Robot Res* 36(1):3–15. <https://doi.org/10.1177/0278364916679498>
- Maddern W, Pascoe G, Gadd M, Barnes D, Yeomans B, Newman P (2020) Real-time kinematic ground truth for the oxford robotcar dataset. <https://doi.org/10.48550/ARXIV.2002.10152>. <https://arxiv.org/abs/2002.10152>
- Majdik AL, Albers-Schoenberg Y, Scaramuzza D (2013) MAV urban localization from google street view data. In: 2013 IEEE/RSJ international conference on intelligent robots and systems. IEEE. <https://doi.org/10.1109/iros.2013.6696925>
- Majdik AL, Verda D, Albers-Schoenberg Y, Scaramuzza D (2014) Micro air vehicle localization and position tracking from textured 3D cadastral models. In: 2014 IEEE international conference on robotics and automation (ICRA). IEEE. <https://doi.org/10.1109/icra.2014.6906964>
- Majdik AL, Verda D, Albers-Schoenberg Y, Scaramuzza D (2015) Air-ground matching: appearance-based GPS-denied urban localization of micro aerial vehicles. *J Field Robot* 32(7):1015–1039. <https://doi.org/10.1002/rob.21585>
- Majdik AL, Till C, Scaramuzza D (2017) The Zurich urban micro aerial vehicle dataset. *Int J Robot Res* 36(3):269–273. <https://doi.org/10.1177/0278364917702237>
- McCormac J, Handa A, Leutenegger S, Davison AJ (2017) SceneNet RGB-d: Can 5 m synthetic images beat generic ImageNet pre-training on indoor segmentation? In: 2017 IEEE international conference on computer vision (ICCV). IEEE. <https://doi.org/10.1109/iccv.2017.292>
- Mitterberger D, Dörfler K, Sandy T, Salveridou F, Hutter M, Gramazio F, Kohler M (2020) Augmented bricklaying. *Constr Robot* 4(3–4):151–161. <https://doi.org/10.1007/s41693-020-00035-8>
- Mokssit S, Licea DB, Guermah B, Ghogho M (2023) Deep learning techniques for visual SLAM: a survey. *IEEE Access* 11:20026–20050. <https://doi.org/10.1109/access.2023.3249661>
- Muñoz-Salinas R, Medina-Carnicer R (2020) UcoSLAM: simultaneous localization and mapping by fusion of keypoints and squared planar markers. *Pattern Recogn* 101:107193. <https://doi.org/10.1016/j.patcog.2019.107193>
- Muñoz-Salinas R, Marin-Jimenez MJ, Medina-Carnicer R (2019a) SPM-SLAM: simultaneous localization and mapping with squared planar markers. *Pattern Recogn* 86:156–171. <https://doi.org/10.1016/j.patcog.2018.09.003>
- Muñoz-Salinas R, Marín-Jimenez MJ, Medina-Carnicer R (2019b) SPM-SLAM: simultaneous localization and mapping with squared planar markers. *Pattern Recogn* 86:156–171. <https://doi.org/10.1016/j.patcog.2018.09.003>
- Mur-Artal R, Montiel JMM, Tardos JD (2015) ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Trans Robot* 31(5):1147–1163. <https://doi.org/10.1109/tro.2015.2463671>
- Mur-Artal R, Tardos JD (2017) ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-d cameras. *IEEE Trans Robot* 33(5):1255–1262. <https://doi.org/10.1109/tro.2017.2705103>
- Nan L, Wonka P (2017) PolyFit: polygonal surface reconstruction from point clouds. In: 2017 IEEE international conference on computer vision (ICCV). IEEE. <https://doi.org/10.1109/iccv.2017.258>
- Nguyen T-M, Yuan S, Cao M, Lyu Y, Nguyen TH, Xie L (2021) NTU VIRAL: a visual-inertial-ranging-lidar dataset, from an aerial vehicle viewpoint. *Int J Robot Res* 41(3):270–280. <https://doi.org/10.1177/02783649211052312>
- Olson E (2011) AprilTag: a robust and flexible visual fiducial system. In: 2011 IEEE international conference on robotics and automation. IEEE. <https://doi.org/10.1109/icra.2011.5979561>
- Optitrack (2023) Optitrack website. <https://optitrack.com/cameras/flex-13/>. Last accessed on 2023-11-05
- Parry C, Guy S (2021) Recycling construction waste material with the use of AR. In: Proceedings of the 2020 DigitalFUTURES. Springer, Singapore, pp 57–67. https://doi.org/10.1007/978-981-33-4400-6_6
- Pfrommer B, Daniilidis K (2019) Tagslam: robust slam with fiducial markers. <https://doi.org/10.48550/ARXIV.1910.00679>. <https://arxiv.org/abs/1910.00679>
- Qin T, Li P, Shen S (2018) VINS-mono: a robust and versatile monocular visual-inertial state estimator. *IEEE Trans Robot* 34(4):1004–1020. <https://doi.org/10.1109/tro.2018.2853729>
- Recchiuto CT, Scalmato A, Sgorbissa A (2017) A dataset for human localization and mapping with wearable sensors. *Robot Auton Syst* 97:136–143. <https://doi.org/10.1016/j.robot.2017.08.004>
- Rivers A, Moyer IE, Durand F (2012) Position-correcting tools for 2D digital fabrication. *ACM Trans Graph* 31(4):1–7. <https://doi.org/10.1145/2185520.2185584>
- Rogeanu N, Tiberghien V, Latteur P, Weinand Y (2020) Robotic insertion of timber joints using visual detection of fiducial markers. In: Proceedings of the 37th international symposium on automation and robotics in construction (ISARC). International Association for Automation and Robotics in Construction (IAARC). <https://doi.org/10.22260/isarc2020/0068>
- RunCam (2023) RunCam2 4K version user manual. https://store-m8o52p.mybigcommerce.com/product_images/img_hybrid/PDF_icon.jpg. Last accessed on 2023-11-05
- Rusinkiewicz S, Levoy M (2001) Efficient variants of the ICP algorithm. In: Proceedings 3rd international conference on 3-D digital imaging and modeling. IEEE Comput. Soc. <https://doi.org/10.1109/im.2001.924423>
- 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. IEEE. <https://doi.org/10.1109/cvpr.2013.178>
- Sandy T, Buchli J (2018) Object-based visual-inertial tracking for additive fabrication. *IEEE Robot Autom Lett* 3(3):1370–1377. <https://doi.org/10.1109/lra.2018.2798700>

- Sandy T, Gifftthaler M, Dorfler K, Kohler M, Buchli J (2016) Autonomous repositioning and localization of an in situ fabricator. In: 2016 IEEE international conference on robotics and automation (ICRA). IEEE. <https://doi.org/10.1109/icra.2016.7487449>
- Schubert D, Goll T, Demmel N, Usenko V, Stuckler J, Cremers D (2018) The TUM VI benchmark for evaluating visual-inertial odometry. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE. <https://doi.org/10.1109/iros.2018.8593419>
- Settimi A, Yang H-B (2023). Software of TSlam: a hybrid tag-based object-centered monocular navigation system for augmented manual wood-working fabrication. <https://doi.org/10.5281/zenodo.10093230>. <https://github.com/ibois-epfl/TSlam>
- Settimi A, Gamarro J, Weinand Y (2022) Augmented-reality-assisted timber drilling with smart retrofitted tools. *Autom Constr* 139:104272. <https://doi.org/10.1016/j.autcon.2022.104272>
- Settimi A, Yang H-B, Gamarro J, Weinand Y (2023). Tslam-dataset. <https://doi.org/10.5281/ZENODO.8377793>. <https://zenodo.org/record/8377793>
- Shaper (2021) Shaper origin tool website. <https://www.shapertools.com/>. Last accessed on 2021-10-24
- Shaya K, Mavrinac A, Herrera JLA, Chen X (2012) A self-localization system with global error reduction and online map-building capabilities. In: *Intelligent robotics and applications*. Springer, Berlin, pp 13–22. https://doi.org/10.1007/978-3-642-33503-7_2
- Shin Y-S, Park YS, Kim A (2018) Direct visual SLAM using sparse depth for camera-LiDAR system. In: 2018 IEEE international conference on robotics and automation (ICRA). IEEE. <https://doi.org/10.1109/icra.2018.8461102>
- Sturm J, Engelhard N, Endres F, Burgard W, Cremers D (2012) A benchmark for the evaluation of RGB-d SLAM systems. In: 2012 IEEE/RSJ international conference on intelligent robots and systems. IEEE. <https://doi.org/10.1109/iros.2012.6385773>
- Taheri H, Xia ZC (2021) SLAM: definition and evolution. *Eng Appl Artif Intell* 97:104032. <https://doi.org/10.1016/j.engappai.2020.104032>
- Tateno K, Tombari F, Laina I, Navab N (2017) CNN-SLAM: real-time dense monocular SLAM with learned depth prediction. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE. <https://doi.org/10.1109/cvpr.2017.695>
- Thoma A, Adel A, Helmreich M, Wehrle T, Gramazio F, Kohler M (2018) Robotic fabrication of bespoke timber frame modules. In: *Robotic fabrication in architecture, art and design*. Springer, London, pp 447–458. https://doi.org/10.1007/978-3-319-92294-2_34
- Thoma A, Jenny D, Helmreich M, Gandia A, Gramazio F, Kohler M (2019) Cooperative robotic fabrication of timber dowel assemblies. In: *Research culture in architecture*. De Gruyter, pp 77–88. <https://doi.org/10.1515/9783035620238-008>
- Trzeciak M, Pluta K, Fathy Y, Alcalde L, Chee S, Bromley A, Brilakis I, Alliez P (2023) ConSLAM: periodically collected real-world construction dataset for SLAM and progress monitoring. In: *Lecture notes in computer science*. Springer, London, pp 317–331. https://doi.org/10.1007/978-3-031-25082-8_21
- Umeyama S (1991) Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans Pattern Anal Mach Intell* 13(4):376–380. <https://doi.org/10.1109/34.88573>
- Wagner D, Schmalstieg D (2007) Artoolkitplus for pose tracking on mobile devices. In: Grabner M, Grabner H (eds) *Computer Vision Winter Workshop*, St. Lambrecht, Austria. <https://api.semanticscholar.org/CorpusID:14966142>
- Wang J, Olson E (2016) AprilTag2: efficient and robust fiducial detection. In: 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE. <https://doi.org/10.1109/iros.2016.7759617>
- Wang W, Zhu D, Wang X, Hu Y, Qiu Y, Wang C, Hu Y, Kapoor A, Scherer S (2020) Tartanair: a dataset to push the limits of visual slam. <https://doi.org/10.48550/ARXIV.2003.14338>. <https://arxiv.org/abs/2003.14338>
- Wein R, Berberich E, Fogel E, Halperin D, Hemmer M, Salzman O, Zukerman B (2023) 2D arrangements. In: *CGAL user and reference manual*, 5.6 Edition. CGAL Editorial Board. <https://doc.cgal.org/5.6/Manual/packages.html#PkgArrangementOnSurface2>
- Zampogiannis K, Fermuller C, Aloimonos Y (2018) Cilantro. In: *Proceedings of the 26th ACM international conference on multimedia*. ACM. <https://doi.org/10.1145/3240508.3243655>
- Zhang Z, Scaramuzza D (2018) A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 7244–7251. <https://doi.org/10.1109/IROS.2018.8593941>. https://github.com/uzh-rpg/rpg_trajectory_evaluation
- Zhu AZ, Yuan L, Chaney K, Daniilidis K (2018a) Ev-flownet: self-supervised optical flow estimation for event-based cameras. <https://doi.org/10.48550/ARXIV.1802.06898>. <https://arxiv.org/abs/1802.06898>
- Zhu AZ, Thakur D, Ozaşlan T, Pfrommer B, Kumar V, Daniilidis K (2018b) The multivehicle stereo event camera dataset: an event camera dataset for 3D perception. *IEEE Robot Autom Lett* 3(3):2032–2039. <https://doi.org/10.1109/lra.2018.2800793>
- Zhu Y, Kong Y, Jie Y, Xu S, Cheng H (2023) Graco: a multimodal dataset for ground and aerial cooperative localization and mapping. *IEEE Robot Autom Lett* 8(2):966–973. <https://doi.org/10.1109/LRA.2023.3234802>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Andrea Settimi¹  · Hong-Bin Yang¹ · Julien Gamarro¹ · Yves Weinand¹

✉ Andrea Settimi
andrea.settimi@epfl.ch

¹ IBOIS, EPFL, GC H2 711, 1015 Lausanne, Vaud, Switzerland