STeAM: Sensor Tracking and Mapping

Carlos Sánchez* Simone Ceriani* Pierluigi Taddei*
Erik Wolfart * Antonio L. Rodríguez* Vítor Sequeira*

ABSTRACT

We present a portable system for tracking and mapping using a real time 3D sensor. In particular our current implementation is composed by a spinning laser sensor mounted on a portable backpack that sends positioning and acquisition data to an hand-held tablet.

The system can be directly employed as a real time odometer system if no information on the environment are available or may take advantage of the known map. In the latter case an initial automatic off-line stage builds a 3D map of an unknown environment (either using high definition laser scanner acquisitions or our system alone). A second off-line stage extracts useful information from the generated map that are used in the subsequent on-line tracking stage. Finally, real-time pose tracking is performed using a robust ICP implementation that efficiently selects potentially descriptive points, removes outliers and that fuses a local odometer to allow the user to navigate through non-mapped areas.

Our system provides accurate real time positioning information in large indoor environments and, optionally, overlays the current real time 3D acquisition on the original map assisting the user in accurately identifying regions of the environment that undergone changes.

Categories and Subject Descriptors

I.4.8 [**Image Processing And Computer Vision**]: Scene Analysis

General Terms

Measurement, Security, Verification

Keywords

Laser Scanner, Localization, Mapping

*All authors are from European Commission, Joint Research Centre (JRC), Institute for Transuranium Elements (ITU), Via Enrico Fermi 2749, Ispra (VA), Italy; email: {name.surname}@jrc.ec.europa.eu

Indoor Localization Competition 2015 - Seattle, WA, USA

1. ENVIRONMENT MAPPING

To localize the sensor we need to perform a one-time activity to build a map of the environment. Our system employs either a high definition point cloud generated by pair wise registration of laser scanner acquisitions, yielding accurate maps consisting of 3D points, normals and (optionally) colors or incrementally build a map using the system itself. In this latter approach, once the sensor acquires a batch of frames, it performs a distributed ICP (Iterative Closest Point) using the map built so far to recover the trajectory followed by the user. In particular our system employs a specific optimization technique for spinning lasers that properly unwarps each sensor acquisition considering a continuous time trajectory yielding robustness even under abrupt movements (typical of portable systems). Once optimized, the points of the scans are added to a voxelized 3D map. The resulting map acts as a reference for the next scans. This technique can be thought as a SLAM (Self Localization And Mapping) system. By employing a hierarchical map representation we are able to update and display very large point clouds.

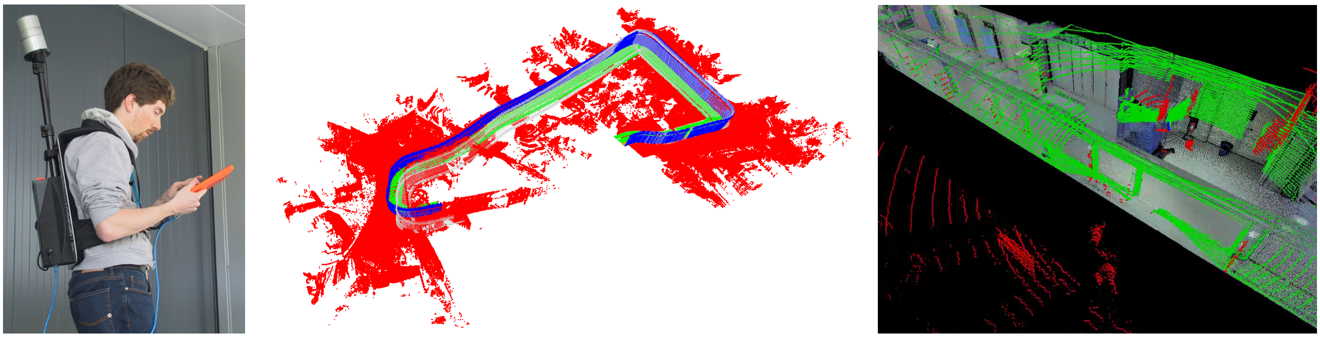
Our current implementation of the environment mapping stage runs with no user intervention in almost real time on a standard computer and, in the future, we expect to generate maps at frame rate by performing additional optimizations. Notice however that the current system is not able to compensate drift that accumulates when long tracks are recorded (i.e., it currently does not include loop detection and closure strategies).

2. PRE-PROCESSING

Taking as input data the 3D map generated in the previous stage and (optionally) the optimized trajectory, the pre-processing step extracts useful information for accelerating the real-time tracking. This stage is performed off-line and with no user intervention.

Specifically in this step the system (1) extracts floor and navigable areas information to define the map boundaries and to facilitate user interaction with the client application, (2) computes a classifier that allows solving the place recognition problem when track is lost, (3) generates a voxelized characterization of the environment to accelerate processing and memory accesses during on-line tracking.

Floor extraction is performed over a sparse voxel representation of the map. In order to label floor cells, it is assumed that the vertical component in their associated normals is dominant. The algorithm is initialized assuming that cells satisfying this condition, and located under the provided tra-



(a) Scanning system (b) Environment mapping stage (c) Localization stage

Figure 1: (a) The scanning system carried by a person, composed by a backpack to support the spinning laser scanner and a tablet computer for user interaction. (b) An example of the map building process used to generate environment maps (c) An example of the output of the system: the current scanned location overlayed on the reference map.

jectory, are part of the floor. Then, iteratively, a flooding step is executed where contiguous cells are labelled as floor if they are reachable (i.e. the estimated volume of the user fits in the map without penetrating full cells). This process is repeated until all reachable cells have been visited. We consider navigable areas all empty cells located within a given region over floor cells (e.g from 1.5 meters to 2.5 meters above the floor).

Using this characterization of the navigable area, we generate a training set of poses for the place recognition solver. By randomly placing a virtual sensor in navigable locations, we generate synthetic acquisitions which are then converted into compact descriptors. These descriptors and the relative poses that generated them are finally used to train a pose classifier. By analysing similarities between pose descriptors, in terms of number of neighbours in a fixed search radius, we effectively identify areas that present high ambiguity values (like corridors) and singular areas where the localization of the sensor can be performed robustly.

Finally, in order to accelerate the registration process performed during the on-line tracking, we compute a voxel representation of the environment where three types of cells are defined: (1) full cells, storing the centroid of all samples contained and an estimated normal, (2) empty cells, with no value associated and (3) empty cells that are closer than a given value (typically between 1 and 2 meters) to a full cell. This last kind of cells are employed to obtain a constant-time access to recover point correspondences during ICP iterations. The proposed voxel structure is stored in a hybrid dense/sparse manner (so access times are kept constant without the burden of high memory requirements) and can be streamed around the sensor pose in case of very large environments.

Our current system employs a k-nearest neighbours based classifier. We are currently working on an alternative solution that also takes into account temporal constraints between subsequent acquisitions and that allows compressing the search space without reducing classification performance.

3. ON-LINE TRACKING

The proposed real-time tracking strategy provides to the user its estimated pose according to the known map and the latest acquisitions of the sensor. The system considers three different states: (1) pose unknown, (2) tracking and (3) re-localization.

Initially, the system has no information on its current position in the map (1) and must identify the most likely position in the known environment. In this situation, the system solves the place recognition problem exploiting the classifier generated in the pre-processing stage. To do so, a compact descriptor of the sensor acquisition is estimated and the classifier is employed to recover the best candidate positions. If no ambiguity in the results is detected, the associated pose to the nearest neighbour will be used to start tracking. Otherwise, the system will repeat this process with the subsequent acquisitions until ambiguity disappears. The user can also provide a hint of its current position by simply pointing a specific location on the hand-held tablet.

When a valid pose is detected, the system enters into the tracking state (2) and it iteratively registers new sensor readings with respect to the known map. To efficiently perform this task, the proposed voxel characterization of the map is exploited (constant time nearest neighbour searches) and a robust and fast ICP implementation is used.

To add robustness to the system and to allow the user to explore non-mapped areas, a local map that considers the last observations is built in parallel and fused into the registration algorithm. This way, tracking continues even when the user leaves the map, but also elements that are not present in the original map can be exploited for registration.

Finally, if tracking is lost, the system enters into the re-localization state (3) and tries to locally recover the sensor pose by solving a local place recognition around the last known pose. If the result is consistent with the map, then the system returns to the tracking state, otherwise, it enters into the more global pose unknown state and starts the place recognition from scratch.