

Easing the Transition Between Multiple Trackers

Martin Wagner, Sven Hennauer and Gudrun Klinker
Technische Universität München, Fakultät für Informatik
Boltzmannstraße 3, 85748 Garching bei München, Germany
(martin.wagner, hennauer, klinker)@in.tum.de

Abstract

Augmented Reality research currently aims at extending the working range of applications by combining multiple trackers in adjacent areas. The transition between two such devices leads to discontinuities in the trajectory of a tracked object. The result are “jumps” in visual augmentations shown to the user.

We present a three step unsupervised learning algorithm that determines the working areas of involved trackers and the area they overlap, and permanently observes the tracked object’s position with regard to these areas in order to enable a smooth interpolation within the overlapping area from one tracker’s readings to another’s. We have tested the algorithm’s performance in an experimental setup. The results show that the method is feasible and only adds a negligible overhead to AR systems.

1. Introduction

A major trend in current Augmented Reality (AR) research is to extend the working area of setups that traditionally have been restricted to the volume covered by a single tracker (e.g. [4]). To accomplish this goal, multiple trackers have to be combined. In this paper, we treat the neglected problem of providing a smooth transition between two tracking devices with partially overlapping working volume.

2. Problem Statement and Approach

As depicted in figure 1, a tracked object is moving from left to right, thus starting in the area of Tracker 1, going through the overlapping area and finally being tracked solely by Tracker 2. As long as the object is in the area observed exclusively by Tracker 1, we just use its measurements. Once the object reaches the overlapping area, we start to combine both sensors’ readings by means of a weighted interpolation that culminates in solely using the measurements of

Tracker 2 once the object leaves the overlapping area (see figure 1(c)). In consequence, we have flattened out the discontinuity and minimized the visual distraction of the user.

To make this idea work, we propose the following algorithm.

Determine Working Volumes. We use fully unsupervised machine learning techniques to determine the tracker’s working volumes automatically without any user or developer intervention.

Determine “Distance” to Tracker Boundary. Depending on the parameterization of the working volume description, we have to choose a suitable method to compute the distance to the tracker whose area we are about to leave.

Interpolate Tracker Measurements. We use the chosen metric to adjust the interpolation scheme between both trackers.

3. Estimating Working Volumes

We implemented the first step of our algorithm in two fully unsupervised ways, i.e. there was no need to perform specific calibration steps, just using the system for a certain time (usually less than 5 minutes) sufficed to initialize the system.

Convex Hull Approach Based on the assumption that the working volume of a tracker is convex, we represent the working volume as the convex hull of all points which the tracker can retrieve. This information can be collected continuously while moving around a tracked object. During this learning phase, the convex hull of the tracker is updated incrementally, which means that each position data received is immediately inserted into the convex hull. To be able to perform this computation efficiently, we use an incremental algorithm [1].

Neural Network-Based Classifier This approach separates the sets of points of both trackers by means of a decision boundary. The location of it indicates the center of

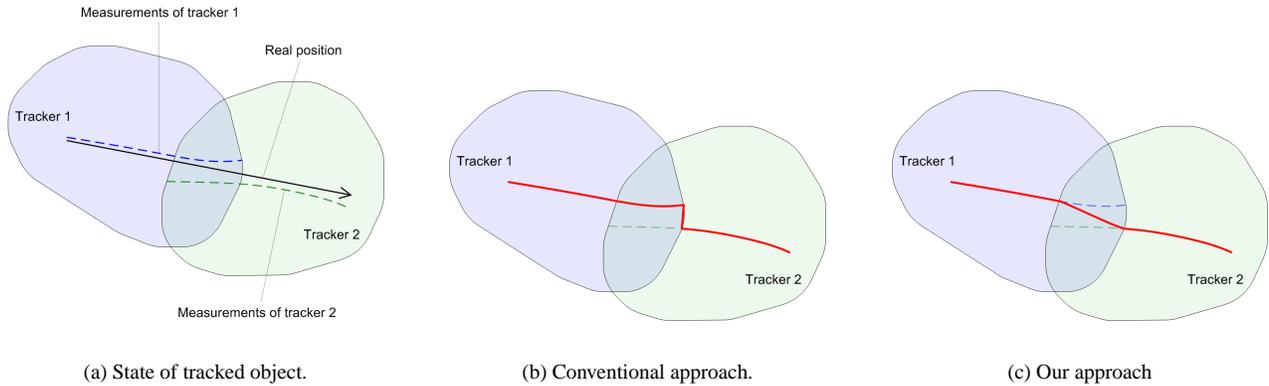


Figure 1. Tracker Setup: Two trackers measure the position and orientation of the same object simultaneously, both yielding values that differ from the object's real pose.

the overlapping area and therefore allows us to draw conclusions concerning the placement of the working volumes.

Since we do not have any information about the shape of this boundary, we use a multi-layer feed forward neural network with three inputs, a layer with three hidden neurons and one output as a universal classifier. This network is trained using the standard backpropagation algorithm [3] with position data of both trackers. After training, the network will produce positive or negative outputs depending on which side of the boundary the input-coordinates are located.

4. Results and Future Work

Our evaluation setup consists of an optical *ARTrack1 / DTrack* system from A.R.T. GmbH and an *IS-600* system from Intersense Inc. Each system roughly observes half of our lab, and their working volumes overlap in an area of about two meters. We use a camera tracked by both systems to produce video see-through AR on either a head mounted display or a standard laptop computer for debugging purposes.

The system has been implemented using the DWARF AR framework [2]. We implemented a `MultiTracker` service that takes pose data from two services abstracting our two trackers and yields the pose data combined according to the working volume algorithms shown above and a simple linear interpolation scheme for position and the SLERP algorithm [5] for orientation. The DWARF tracker abstraction allows to use the functionality of the `MultiTracker` service without changing any other components involved.

Runtime measurements on a Athlon-XP 2500 standard PC showed that the convex hull algorithm could be trained on-line with only a marginal overhead (approx. $70\mu s$),

while the neural net took considerable more time to train (max. $10s$). After the training phase, both algorithms were sufficiently fast, including our simple linear interpolation of position and rotation it usually took less than $0.1ms$ for all three steps to complete.

Next steps will be a more thorough interpolation scheme and user studies to get solid results on how the visual distraction can be minimized with our approach.

Acknowledgments

This work was supported by the Deutsche Forschungsgemeinschaft, project DySenNetz (KL1460/1-1). The Intersense tracker was on loan from Volkswagen AG.

References

- [1] The incremental algorithm.
http://www-student.cs.uni-bonn.de/~wichmann/writings/convex_hull/node3.html.
- [2] M. Bauer, B. Bruegge, G. Klinker, A. MacWilliams, T. Reichler, S. Riss, C. Sandor, and M. Wagner. Design of a component-based augmented reality framework. In *Proceedings of the 2nd International Symposium on Augmented Reality (ISAR 2001)*, New York, USA, 2001.
- [3] C. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.
- [4] T. Caudell and D. Mizell. Augmented reality: An application of heads-up display technology to manual manufacturing processes. In *Proc. HICCS '92*, 1992.
- [5] E. B. Dam, M. Koch, and M. Lillholm. Quaternions, interpolation and animation. Technical Report DIKU-TR-98/5, University of Copenhagen, July 1998.