

# Autocalibration in Ubiquitous Tracking Environments

Daniel Pustka

January 15, 2004

## Ubiquitous Tracking and Spatial Relationship Graphs

Virtually all current AR applications are built for a particular tracking technology. Also transforms between different coordinate frames that are used in the application are usually hard-coded. That makes it difficult to move such an application to a different tracking setup.

The Ubiquitous Tracking approach aims to solve this problem. All positional measurements form a spacial relationship graph. Nodes in the graph represent identifiable objects or coordinate frame. An edge in the graph stands for a known spacial relation between the two nodes, such as a tracker measurement or a previously known static relation. By finding paths in the graph, the UbiTrack framework can automatically infer transformations between different coordinate systems and a single tracking API gives the application access to all available tracking hardware.

## Motivation for Autocalibration

Most current AR applications require a separate calibration step, in which the user walks around and aligns objects to known locations. This information is used to establish calibration values that describe the relation between some real-world coordinate system and that of a tracker or an output device. In an scenario with multiple trackers such a relation can often be inferred automatically by using overlapping tracking areas. What makes calibration different from the normal accumulation of edges in the UbiTrack environment is the fact that calibration values are considered to be constant. Therefore measurements used to perform calibration and those used for realtime tracking need not (but can) occur in the same time frame. Also many measurements at different positions and times can be combined to give a higher calibration accuracy.

An interesting example is a setup where the working areas of two trackers overlap. Any object that is detected by the two trackers at the same time will automatically calibrate the two trackers with respect to each other. Therefore the transition between those areas will be smoother.

When data fusion is used to combine the measurements of multiple sensors, a good calibration also is necessary – otherwise the two trackers deliver too contradicting data.

## UbiTrack implementation

Autocalibration can be implemented by inserting additional edges into the static relationship graph. These edges describe the transformation from the tracker coordinate system to some reference system. This transformation is assumed to be constant. A calibration service then tries to find other measurements that can be used to infer the calibration coordinate transformation and combines measurements from different times and positions in order to give a stable estimation.

The calibration transformation can be represented in different ways. If only translation and rotation need to be considered, a 3D vector and a quaternion suffices, having the advantage of being compatible with the measurements of ordinary trackers. More sophisticated representations are homogenous matrices (allows scaling and shearing) and nonlinear models.

## Thesis contents

The thesis will cover the following topics:

- Overview of existing autocalibration and manual calibration techniques.
- Explanation of mathematical methods for autocalibration.
- Integration of autocalibration into the DWARF and UbiTrack frameworks.
- Implementation of autocalibration services.
- Evaluation of experiments with available tracking equipment.

## Challenges

**Dealing with uncertainty:** All measurements in the ubiquitous tracking setup are associated with a covariance matrix which gives an estimation of the tracking error. This error also affects the estimation of the calibration parameters, as precise measurements should be weighted differently than

rough position estimations. Therefore a simple least squares estimation is probably not sufficient.

Also, the resulting calibration values should also include error estimations in order to give the framework the ability to decide how to weight calibrated values with respect to other measurements in subsequent client requests.

**Persistence of measurements:** In current DWARF setups, tracking values are short-lived information. Spatial information is passed as a message to subscribers and discarded afterwards. In order to perform autocalibration, spatial information needs to be collected for later fusing with other measurements. As it is not feasible to store all acquired measurements, it will be necessary to select samples. Selection criteria could include precision and the density of measurements in the working area of a sensor.

Also, it is not always immediately clear if a measurement will be relevant for a later autocalibration step as other measurements that can serve as a link in the SR graph may become available at a later time.

**Calibration beyond linear coordinate transforms:** It should be investigated how the autocalibration in ubiquitous tracking approach can be applied to other calibration problems such as estimation of internal camera parameters or drift in velocity and acceleration sensors.

**Integrating manual calibration:** In many tracking setup no automatic calibration can be performed as there are no reference measurements available (e.g. the projector in the SHEEP project). In this case normal calibration has to be integrated into the UbiTrack framework. This can be done by adding the manually specified positions as measurements to the SR graph.

## Schedule

|                  |  |
|------------------|--|
| until workshop   | Literature scanning for other manual calibration and autocalibration problems and solutions. Collecting more UbiTrack ideas. |
| workshop         | Specification of api, interfaces   |
| workshop - march | Theoretical work and first implementation of algorithms  |
| april            | Integration with other parts of the project and evaluation of results  |
| may              | Final demo   |
| june - july      | Writing thesis   |