# An Alternative Way to Calibrate Ubisense Real-Time Location System via Multi-Camera Calibration Methods

**Rok Mandeljc, Janez Perš, Matej Kristan, Stanislav Kovačič**

*Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, 1000 Ljubljana, Slovenia*
*E-mail: rok.mandeljc@gmail.com*

## Abstract

*In the paper, an alternative approach to calibration of Ubisense Real-Time Location System is considered. The approach is based on capturing the raw angles of arrival and projecting them into virtual image plane, as if sensors were perspective cameras. The extrinsic parameters (position and orientation) of sensors are then obtained by calibration of virtual perspective cameras using multi-camera calibration methods.*

*An application considered in the paper is rapid deployment of Ubisense system for tracking in sports. Survey points can be easily determined from the standard markings on the court floor, which makes calibration from survey points coordinates more convenient than measuring sensor positions, which is prerequisite for standard Ubisense system calibration.*

## 1 Introduction

The Ubisense Real-Time Location System for tracking people and assets has seen adoption in a wide range of applications, such as manufacturing, military, transport, childcare, livestock tracking and tracking in sports.

The main characteristic of the usual Ubisense system deployments is that the system is fixed; the sensors are permanently mounted to the locations that were determined as optimal during the site survey and planning. After physical deployment and sensor calibration the system is ready for use and its configuration (with respect to sensor positions) usually does not change.

In this paper, however, we consider the possibility of a mobile deployment for tracking in sports; the sensors are mounted on camera stands — possibly along with the cameras in case of a combined radio-video tracking system — and brought to the sports hall, stadium or the court where they are deployed for each game. The obvious advantage of such deployment is that the Ubisense system can be used in places where it would otherwise not be available or in places that lack the proper infrastructure (outside, improvised courts, etc.). The disadvantage, however, is that during each deployment the system needs to be calibrated.

Furthermore, we assume that a laser survey instrument (theodolite or tachymeter) for measuring sensors' positions is not necessarily available. Therefore measuring the sensors' positions might not be feasible and it might be more convenient to measure a set of survey points in the working volume (i.e. court). This is especially the case with the sports, because a set of points with known coordinates can be obtained with the help of standard markings on the court floor. The Ubisense system, however, requires the sensor positions to be known in order to perform the rest of calibration.

Therefore in this paper, alternative ways to calibrate the system are explored, based on the multi-camera calibration methods developed in the field of computer vision.

The rest of the paper is organized as follows: in Section 2, a brief overview of Ubisense RTLS and its main characteristics is given, along with the standard way of calibrating such a system. In Section 3, two multi-camera calibration methods are briefly presented. Section 4 contains description of a preliminary experiment and its results and in Section 5 the conclusions drawn from the experiment are given.

## 2 The Ubisense Real-Time Location System

The Ubisense Real-Time Location System consists of sensors, tags and Ubisense software platform running on a PC.

The sensor is a precision ultra-wideband (UWB) measurement device that contains an array of antennas and UWB radio receivers. It detects UWB pulses from the tags, allowing the Ubisense location system to find the tags' positions. The sensors are connected to a PC via Ethernet cable. In addition, sensors are connected among themselves with Ethernet cables that serve as timing cables.

The tag transmits UWB radio pulses which are detected by sensors; each sensor measures angle of arrival (AoA) and time difference of arrival (TDoA) of the incoming signal and this information is used to determine tag's location.

The system operates on $6$ – $8$ $GHz$ radio frequency range. In addition, $2.4$ $GHz$ channel is used for sending telemetry commands to the tags (such as when to emit a pulse). The advertised operating range (in open conditions) is up to $160$ $m$ with achievable accuracy better than $30$ $cm$. The angles of a sensor coverage are $120°$ horizontally and $100°$ vertically.

Figure 1: Ubisense Series 7000 Hardware: a Slim Tag (top left), a Compact Tag (top right) and a Sensor (bottom)

### 2.1 Typical system calibration

The typical deployment and calibration routine for the system as outlined in the Ubisense *Location Engine Configuration Manual* [1] consists of following steps:

1. Install a sensor cell
2. Measure the sensor positions
3. Start the location engine software
4. Add or import sensors
5. Configure the cell plan
6. Configure tag range
7. Boot sensors
8. Calibrate the sensors thresholds
9. Wake up tags
10. Calibrate orientation and cable offsets
11. Check the operation

The recommended practice for measuring sensor positions is to use a laser surveying instrument and fiducial marks that can be found on sensors' front sides if the cover is removed.

Known sensor positions are prerequisite for the orientation and cable offset calibration using the methods built in the Ubisense software. There are three ways of performing orientation and cable offset calibration.

For *full calibration*, multiple measurements from five or more survey points with known and fixed Z coordinate are used to determine both orientation and cable offsets for all sensors.

In *dual calibration*, multiple measurements from a single survey point with known X, Y and Z coordinates are used to determine orientation and cable offsets for a pair of sensors.

The third option is to use *orientation calibration*, which determines orientation (pitch and yaw; roll is assumed to be zero) and *cable calibration*, which determines cable offset. Both methods need to be performed on each sensor, and they both require multiple measurements from a single survey point with known X, Y and Z coordinates.

### 2.2 Location engine data

Once the system is calibrated, the data from Location engine can be obtained either via Ubisense's .NET API or by specifying a custom sink and implementing a UDP client that receives Ubisense On-The-Wire protocol packets [2]. These contain the processed information, such as tag ID, its location, location error, etc.

Raw data, such as the angle of arrival for each sensor, can be obtained by capturing the events using the Ubisense Location Engine Config program. Up to 10000 events can be captured and exported to an XML file [3]. It should be noted here that raw data can be obtained even from an uncalibrated system, which is the basis for our attempt of an alternative calibration.

## 3 Multi-camera calibration methods

Contrary to the Ubisense system calibration method, where positions of sensors need to be known, the multi-camera calibration methods developed and used in the field of computer vision do not require the positions of cameras.

Both extrinsic camera parameters (position and orientation) and intrinsic camera parameters (focal length, principal point, skew) are estimated from multiple camera views of corresponding points and either from known locations (coordinates) of those points or from additional constraints such as the ones imposed by the rigidity of the scene.

Here, application of two such methods is considered; first one was developed by Zhengyou Zhang [4] and the second by Tomáš Svoboda [7].

### 3.1 Zhang's method

Zhang's method [4], which has been implemented in the MATLAB Camera Calibration Toolbox by Jean - Yves Bouguet [5], is similar to the Direct Linear Transformation (DLT) method [6], which is a textbook example of camera calibration method.

The method estimates camera's extrinsic and intrinsic parameters along with lens distortion coefficients — although estimation of intrinsic parameters and lens distortion coefficients can be disabled.

The method requires the world coordinates of survey points along with the coordinates of their projections in the image plane for each camera to be calibrated.

### 3.2 Svoboda's method

The method developed by Tomaš Svoboda [7], for which a MATLAB implementation also exists [8], requires only that a calibration object is moved around throughout the working volume; world coordinates of the survey points are not needed.

Using rank-4 factorization, both projective motion (i.e. cameras' projection matrices) and projective shape (i.e. points' 3-D world coordinates) are estimated. However, in the end, the resulting coordinate system still needs to be aligned with the real coordinate system.

### 3.3 The idea

The main idea behind our work is that raw angle of arrival measurements (azimuth and elevation) as received by Ubisense sensors are equivalent to coordinates in the image plane if perspective cameras were used instead of the sensors.

The two are linked by an arbitrary perspective transformation; therefore, each sensor can be thought of as a virtual camera with arbitrarily chosen intrinsic parameters. The captured azimuth and elevation can be projected onto virtual image plane and the resulting virtual images can be used as an input to multi-camera calibration methods.

The resulting sensors' positions and orientations can then be entered into Ubisense Location Engine Config and the only part of calibration left to perform is the cable offset calibration, which can be done using a single survey point.

The way Ubisense system operates allows the use of multiple tags simultaneously; since only one tag is active during a given timeslot and since tag IDs are recorded, determining correspondences across virtual images is not an issue.

On the other hand, azimuth and elevation data obtained by Ubisense sensors (and consequently the virtual image coordinates) tend to be much noisier compared to the real cameras. Furthermore, in case of multipath occurrences, significant outliers occur in the measured data. Both of these can result in degradation of the camera calibration.

## 4 Experimental results

### 4.1 The setup

A simple preliminary experiment was carried out in order to verify the plausibility of the proposed application of camera calibration methods.

Four sensors were placed in the corners of a $6 \times 4\ m$ grid with resolution $1\ m$; at each point on the grid, a tag was placed on the ground and $1\ m$ above the ground. For each such ground truth position, 100 events (containing, among other data, azimuth and elevation readings from all sensors) were captured. Out of these 100 measurements, depending on the method used, 1 or 4 measurements were obtained by calculating median of the corresponding (sub)intervals with the aim of robust handling of outliers caused by multi-path and reflections.

For virtual cameras' intrinsic parameters, a $120°$ field of view and $600 \times 600$ pixel image plane were chosen, with principal point in the middle. Using this virtual camera model, azimuth and elevation angles were projected into an image plane; the resulting virtual images served as an input to the camera calibration methods. The choice of image plane resolution is arbitrary and does not affect the results of calibration, whereas choosing a narrower field of view causes measurements that "fall out of view" to be discarded.

Before the experiment, the system was calibrated via standard Ubisense calibration method; while this is not necessary for obtaining the raw angle of arrival measurements, it does allow for comparison and evaluation of parameters obtained from camera calibration methods.

For Zhang's method, a single azimuth and elevation measurement at each tag position was used. Because we are interested only in extrinsic parameters, the optimization of intrinsic parameters and calculation of lens distortion (which is not applicable) has been disabled. This also prevents the method from making changes to intrinsic parameters in order to compensate for the noisy input data.

For Svoboda's method, four azimuth and elevation measurements at each tag position were used. Bundle adjustment was enabled due to noisy input data and only a single iteration was used in order to prevent the lens distortion correction which would have been triggered due to relatively high reprojection error, but is otherwise inapplicable to our virtual camera model.

### 4.2 Reconstruction error

In the process of determining the camera parameters, both methods attempt to minimize the reprojection error, which is the deviation of reprojected point coordinates in the 2-D image plane from the measured ones.

The other indicator of calibration accuracy, which is perhaps even more relevant in our application, is the reconstruction error — the deviation of the reconstructed 3-D coordinates from the measured ones.

In case of Svoboda's method, the reconstructed coordinates are obtained together with the virtual camera parameters and can be compared directly to the known control coordinates by means of Euclidean distance. When calculating reconstruction error, only points recognized as inliers were taken into account.

For Zhang's method, reconstructed coordinates can be obtained by calculating DLT parameters for all virtual cameras and reconstructing a 3-D point from two or more 2-D image projections using least-squares method [6].

In addition, two more reconstruction errors are considered and presented in Table 1. First is the error of Ubisense Location Engine; because the experiment was performed with fully calibrated Ubisense system, the accuracy of its tag localization can be assessed. Since angle of arrival and time difference of arrival information are used in process, the localization should be quite accurate (i.e. within the bounds of advertised 30 cm).

Furthermore, 3-D point reconstruction can also be performed by combining known position and orientation of already calibrated sensors with the 2-D image projections corresponding to raw angle of arrival measurements.

Table 1: Reconstruction error (in meters); mean and standard deviation

| | **Ubisense LE** | **AoA** | **Zhang** | **Svoboda** |
|---|---|---|---|---|
| $\mu$ **[m]** | 0.1699 | 0.2090 | 0.1128 | 0.1635 |
| $\sigma$ **[m]** | 0.0987 | 0.1037 | 0.0510 | 0.1074 |

### 4.3 Sensor position and orientation

The goal of our alternative calibration is to obtain sensor parameters that can be entered into Ubisense Location Engine — sensors' position (X, Y and Z coordinate) and orientation (roll, pitch and yaw angle).

These parameters can be extracted from the virtual cameras' projection matrices that are the output of multi-camera calibration methods.

In Table 2, the sensor parameters as estimated by both methods are compared to the parameters that were determined using standard Ubisense calibration method.

Table 2: Sensors' position and orientation

| | | Ubisense | Zhang | Svoboda |
|---|---|---|---|---|
| Sensor #1 | **X [m]** | 6.00 | 5.97 | 6.31 |
| | **Y [m]** | 4.10 | 4.24 | 4.48 |
| | **Z [m]** | 2.20 | 2.21 | 1.99 |
| | **Roll [°]** | 0.00 | -2.54 | 9.50 |
| | **Pitch [°]** | -37.90 | -38.69 | -26.00 |
| | **Yaw [°]** | -131.18 | -129.71 | -139.57 |
| Sensor #2 | **X [m]** | 0.00 | -0.42 | -0.53 |
| | **Y [m]** | 0.00 | -0.17 | -0.50 |
| | **Z [m]** | 2.17 | 2.11 | 2.33 |
| | **Roll [°]** | 0.00 | -0.79 | -1.34 |
| | **Pitch [°]** | -32.74 | -30.80 | -26.45 |
| | **Yaw [°]** | 44.38 | 42.85 | 35.96 |
| Sensor #3 | **X [m]** | 5.90 | 5.98 | 6.04 |
| | **Y [m]** | 0.00 | -0.19 | 0.17 |
| | **Z [m]** | 1.50 | 1.49 | 1.36 |
| | **Roll [°]** | 0.00 | -0.08 | -7.49 |
| | **Pitch [°]** | -27.83 | -27.15 | -13.22 |
| | **Yaw [°]** | 139.82 | 138.54 | 140.30 |
| Sensor #4 | **X [m]** | 0.00 | -0.60 | -0.13 |
| | **Y [m]** | 4.10 | 3.98 | 4.01 |
| | **Z [m]** | 1.55 | 1.55 | 1.47 |
| | **Roll [°]** | 0.00 | 0.76 | 5.60 |
| | **Pitch [°]** | -14.95 | -15.47 | -17.88 |
| | **Yaw [°]** | -43.33 | -37.38 | -39.31 |

## 5 Conclusion and future work

As can be seen, it is possible to calibrate the Ubisense sensors by considering them as virtual cameras and estimating their extrinsic parameters using multi-camera calibration methods.

Using Zhang's method, we obtain reconstruction error that compares very favorably with the Ubisense Location engine. It can also be seen from Table 1 that the Ubisense system puts greater weight on the TDoA information that purely on AoA. Sensor parameters obtained by Zhang's method are also sufficiently close to parameters obtained with standard calibration.

With Svoboda's method, the results are slightly more distorted. There are several reasons for that; the input data contains more noise than the method was designed for (at the end it even warns that reprojection error is relatively high). When estimating the parameters, the method does not separate the intrinsic and extrinsic parameters but instead estimates the projection matrix as a whole. Therefore, the method attempts to compensate for the input noise by adjusting the instrinsic parameters, which in turn results in bogus extrinsic parameters.

Another aspect that makes Svoboda's method less suitable for the proposed sensor calibration is the coordinate system alignment. Because the survey points' coordinates are not needed, the method estimates points' coordinates and sensor parameters in a coordinate system that is centered in the points cloud and has an arbitrary orientation and scale. Since our survey points comprised a grid, it was easy to determine the scale factors, rotation matrix and translation that align the coordinate system with the world one. So even though survey points' coordinates do not need to be measured, at least four points' coordinates are needed afterwards to align the coordinate system.

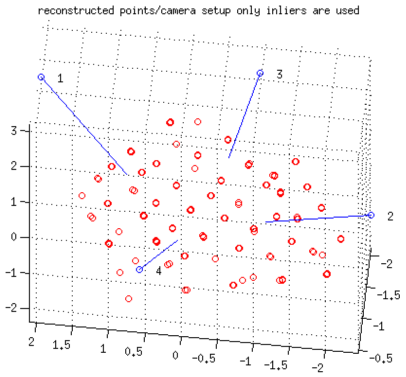

Figure 2: Unaligned result of Svoboda's method

Therefore, our future work will most likely focus on the application of Zhang's method, as our preliminary experiment confirmed its feasibility. As the next step, comparison of accuracy of Ubisense system, calibrated with the proposed alternative method, and of system, calibrated with the standard calibration method, should be performed. Also, the optimal amount of control points for reliable calibration should be determined with an experimental deployment in the sports hall.

## References

[1] Ubisense. LocationEngineConfig User Manual, October 2007

[2] Ubisense Research Network — Version 2 Software Compatibility Matrix Request, June 2010, `http://www.ubisense.org/viewtopic.php?t=191`

[3] Ubisense Research Network — Individual Sensor Raw Data, June 2010, `http://www.ubisense.org/viewtopic.php?f=7\&t=16`

[4] Z. Zhang. Flexible Camera Calibration by Viewing a Plane from Unknown Orientations. In ICCV99, pages 666–673, 1999.

[5] Camera Calibration Toolbox for Matlab, June 2010, `http://www.vision.caltech.edu/bouguetj/calib_doc/index.html`

[6] DLT Method, June 2010, `http://www.kwon3d.com/theory/dlt/dlt.html`

[7] T. Svoboda, D. Martinec, T. Pajdla. A convenient multi-camera self-calibration for virtual environments. PRESENCE: Teleoperators and Virtual Environments, pages 407-422, 14(4), August 2005.

[8] Multi-Camera Self-Calibration, June 2010, `http://cmp.felk.cvut.cz/~svoboda/SelfCal/index.html`