"University of Turku Technical Reports, No.10 — September 2015"

DEPTH SENSORS IN AUGMENTED REALITY SOLUTIONS

Literature Review

Mika Taskinen | Olli Lahdenoja | Tero Säntti | Sami Jokela | Teijo Lehtonen

Mika Taskinen
University of Turku, Technology Research Center, 20014 Turun yliopisto, Finland
mika.taskinen@utu.fi

Olli Lahdenoja
University of Turku, Technology Research Center, 20014 Turun yliopisto, Finland
olli.lahdenoja@utu.fi

Tero Säntti
University of Turku, Technology Research Center, 20014 Turun yliopisto, Finland
tero.santti@utu.fi

Sami Jokela
University of Turku, Technology Research Center, 20014 Turun yliopisto, Finland
sami.jokela@utu.fi

Teijo Lehtonen
University of Turku, Technology Research Center, 20014 Turun yliopisto, Finland
teijo.lehtonen@utu.fi

www.trc.utu.fi

ISSN 2341-8028 | ISBN: 978-951-29-6225-9

Technology
Research
Center

Turun yliopisto
University of Turku

"University of Turku Technical Reports, No.10 — September 2015"

Abstract

The emergence of depth sensors has made it possible to track – not only monocular cues – but also the actual depth values of the environment. This is especially useful in augmented reality solutions, where the position and orientation (pose) of the observer need to be accurately determined. This allows virtual objects to be installed to the view of the user through, for example, a screen of a tablet or augmented reality glasses (e.g. Google glass, etc.). Although the early 3D sensors have been physically quite large, the size of these sensors is decreasing, and possibly – eventually – a 3D sensor could be embedded – for example – to augmented reality glasses. The wider subject area considered in this review is 3D SLAM methods, which take advantage of the 3D information available by modern RGB-D sensors, such as Microsoft Kinect. Thus the review for SLAM (Simultaneous Localization and Mapping) and 3D tracking in augmented reality is a timely subject. We also try to find out the limitations and possibilities of different tracking methods, and how they should be improved, in order to allow efficient integration of the methods to the augmented reality solutions of the future.

Keywords

RGB-D, Depth, Sensor, SLAM, Point Cloud, Plane Filtering

Contents

1 Introduction

Today the use of augmented reality solutions is increasing and more efficient methods for tracking are being developed for motion sensors and conventional cameras. Using depth sensors in this manner is a fairly new area, in comparison as they have not been accurate or mobile enough to be used in augmented reality. Over the last few years this has changed drastically and now some algorithms and methods have finally been researched for the depth sensors.

All sensors have their faults when tracking the pose. A conventional camera might lose its track when the captured image lacks details. Motion sensor has to be accurate and record data in high frequency. Even then the pose might drift unless corrected with other methods. A depth sensor usually does not work under sunlight conditions or when the distance to the tracked surroundings is too great. These, and possibly other methods, need to be combined to achieve more robust tracking.

The advantage of depth sensors in tracking is the actual depth data which contains real distances to target areas. These distances can be used to isolate basic formations like planes, cubes and spheres. The data can also be used when building up a more robust model of the surroundings.

In this review we will find out which kind of tracking solutions there are for the depth sensors or RGB-d cameras. This information can then be used when trying to track the pose or to model the surroundings.

1.1 Definition of depth sensor

The most basic definition of depth sensor is a device that measures distance to a target. This target can be a dot, line or even an array of dots forming an area. Three sensor technologies exist and they are laser, ultrasonic and infrared (usually the RGB-d sensors). All of these types work the same way: By sending a signal and measuring distance from the signal when it is mirrored back from an object. This signal can contain a code for identification to remove noise from surroundings.

Every sensor type can be divided to two categories according to the type of data in the signal. Either the depth information is purely based on intensity of the returning signal or the sensor could use a time of flight (TOF) technology to measure distance. TOF requires that a pattern or data is recorded to the signal so that the specific timestamp can be recorded into it. When the signal returns, the exact time can be measured by using the signal data and return time. The TOF technology improves on noise filtering and thus improves the depth sensor reliability on the whole but it is new technology and many of todays sensors are still intensity based.

Papers in this review focus mainly on infrared and intensity based sensors. These sensors are usually the most cost effective when scanning environments and measuring distances.

1.2 Augmented Reality

The purpose of augmented reality is to enhance the world around observer. This means additional virtual objects in an environment or enhancement of objects that already exist. To accomplish this, there must be someway to track these objects in real time. Until now the most popular tools for tracking have been conventional cameras and motion sensors. The conventional cameras track the environment by extracting stable features from image and then frame by frame track these features to find correct poses of the observer and the observed. Multiple frames are required to gain actual model of objects. Motion sensors rely on active data, giving the observer information on rotation speed and orientation in relation to surroundings.

Depth sensors, like cameras, extract features from image but with additional scope because of the actual distance data. Depth sensors do not require multiple frames to gain three dimensional data and in such can detect simple formations which are easier to find from the next frame than the features detected by color cameras. At the very least, the depth sensor can be used to map the environment for a later use with other sensors.

2 Review Process

The review process started with selected keywords that present the question: "How can depth sensors be used in augmented reality tracking?" The exact search sequence used in the systematic literature search is as follows:

```
((
  (depth NEAR/3* sensor)
    OR
  (depth NEAR/3* camera)
    OR
  "rgb-d"
)
  AND
(
  "tracking"
    OR
  "pose estimation"
    OR
  "augmented reality"
    OR
  "localization"
))
```

*NEAR/#: Given words are within # words from each other in text

Several academic search engines were used and the initial result of the searches counted up to 4861 papers. From these we filtered 111 by reading the titles. By reading the abstracts we filtered the count to 39 and finally, by reading the papers, we got to our result: 19 papers. In addition, two technical reports were found from MIT archives to support our survey. The review was conducted according to the guidelines of a systematic literature search [O2] with the exeption of elimination process where only one person filtered the papers and consulted another person on obscure cases.

All of the selected papers provide information on RGB-d tracking. This does not necessarily mean that the solutions are based purely on depth sensors. A depth sensor is usually used when a model of object or surroundings is made. The tracking part can still be done by regular camera. Some of the papers theorized about depth tracking while most of them only used it for modeling purposes.

Following is the table describing all the paper counts through the elimination process per search engine.

Search Engine	**Query Result**	**Title Filtering**	**Abstract Filtering**	**Text Filtering**
ACM Digital Library[1]	83	6	0	0
CiteSeerX	548	24	7	2
Elsevier Science Direct[2]	512	0	0	0
IEEE Xplorer	498	62	25	16
Springer Link	2546	19	7	1
Web of Science[3]	674	0	0	0
Total	**4861**	**111**	**39**	**19**

Table 1: Systematic literature review stages

[1] Original search result was 2461 papers. The search was later limited to abstracts instead of whole texts because of the unrelated information found.

[2] Search method found only unrelated papers.

[3] Papers that were acceptable were duplicate findings from IEEE Xplorer engine.

3 Analysis

3.1 Articles

The following is a list of the selected papers with brief descriptions.

- **Ataer-Cansizoglu et al., "Tracking an RGB-D camera using points and planes", 2013 [P1]**
  Tracking is done by extracting points and planes from depth data and an extended prediction algorithm is used.

- **Biswas and Veloso, "Depth camera based indoor mobile robot localization and navigation", 2012 [P2]**
  In this paper, a method for depth tracking is introduced. This method, called "Fast Sampling Plane Filtering" (FSPF), handles massive data amounts from the sensor to calculate pose.

- **Biswas and Veloso, "Planar polygon extraction and merging from depth images", 2012 [P3]**
  Fast Sampling Plane Filtering method testing. This paper provides results compared to ground truth and demonstrates how the method works.

- **Bylow et al., "Real-Time Camera Tracking and 3D Reconstruction Using Signed Distance Functions", 2013 [P4]**
  This paper presents a real-time RGB-D camera tracking using signed distance function (SDF) instead of iterative closest point algorithm (ICP) used for instance in KinectFusion. According to this paper, SDF provides more accurate and robust data than ICP.

- **Ceriani et al., *Single and Multi Camera Simultaneous Localization and Mapping Using the Extended Kalman Filter*, 2014 [P5]**
  An extended kalman filter is used for combining results of multiple tracking sources. In this case the sources are conventional cameras or depth sensors.

- **Chen and Lin, "RGB-D Sensor Based Real-time 6DoF-SLAM", 2014 [P6]**
  The basis of the tracking is the color information. Depth information is used when building up model of the environment. SIFT matching and RANSAC are used in this operation.

- **Dryanovski et al., "Real-Time Pose Estimation with RGB-D Camera", 2012 [P7]**
  The tracking is done by feature extraction with both color and depth information.

- **Fallon, Johannsson, and Leonard, "Efficient scene simulation for robust monte carlo localization using an RGB-D camera", 2012 [P8]**
  Kinect Monte Carlo Localization (KMCL) calculates pose from point cloud provided by a depth sensor. This is done by using a 3D-map of the surroundings and using simulated depth and color images as a comparison.

- **Hu et al., "A robust RGB-D SLAM algorithm", 2012 [P9]**
  This paper presents ideas to optimize the RGB-D SLAM.

- **Kerl, Sturm, and Cremers, "Robust odometry estimation for RGB-D cameras", 2013 [P10]**
  The usual visual odometry algorithms like SURF or SIFT are too slow for low latency applications. This paper suggests a new algorithm with lower computational requirements.

- **Klüssendorff et al., "Graph-based visual SLAM and visual odometry using an RGB-D camera", 2013 [P11]**
  Visual SLAM is done by using color images for tracking and depth sensor for scanning. This paper proposes to use FAST algorithm for feature extraction and BRIEF as feature descriptor.

- **Lee, Kim, and Myung, "GPU-Based Real-Time RGB-D 3D SLAM", 2012 [P12]**
  A visual SLAM can be done by GPU for optimization purposes. This paper shows how.

- **Liu et al., "A robust fusion method for RGB-D SLAM", 2013 [P13]**
  RGB-D cameras can be used in environments even when there are poor 3D geometry by combining depth and color information.

- **Maier, Hornung, and Bennewitz, "Real-time navigation in 3D environments based on depth camera data", 2012 [P14]**
  This paper purely tracks with depth sensor. The environment is scanned and then rebuilt for a robot to find its path in the surroundings.

- **Newcombe et al., "KinectFusion : Real-Time Dense Surface Mapping and Tracking", 2011 [P15]**
  KinectFusion can be used in either tracking or modeling and uses GPU for performance purposes.

- **Somlyai, "Mobil robot localization using RGB-D camera", 2013 [P16]**
  Combination of IMU sensors and RGB-D tracking is used by combining movement sensors, color and depth data.

- **Strasdat et al., "Double Window Optimisation for Constant Time Visual SLAM", 2011 [P17]**
  Double window optimization framework intends to improve on visual SLAM. This paper is not depth sensor specific but can be applied to it.

- **Su, Shen, and Cheung, "A robust RGB-D SLAM system for 3D environment with planar surfaces", 2013 [P18]**
  A problem with planar surface tracking is handled by use of both color and depth data.

- **Valenti et al., "Autonomous Quadrotor Flight Using Onboard RGB-D Visual Odometry", 2014 [P19]**
  RGB-D visual odometry is used for micro aerial vehicles for independent flight along 4DOF (4 degrees of freedom) paths.

- **Whelan et al., "Kintinuous: Spatially extended kinectfusion", 2012 [P20]**
  KinectFusion tracks a very small volume. This volume can be extended using a method called Kintinuous.

- **Whelan et al., "Robust Tracking for Real-Time Dense RGB-D Mapping with Kintinuous", 2012 [P21]**
  Kintinuous can be expanded with advanced coloring and more optimal graphics algorithms.

3.2 Position and orientation tracking and feature extraction

Position and orientation of the tracking device can be acquired by variety of methods. In this case, these methods are based on either color or depth information. A very popular approach is to use SLAM (Simultaneous Localization and Mapping) based methods [O1]. SLAM is mainly used for building a model (mapping) of the environment. It is usually done by series of stages: Feature extraction, data association, loop closure and mapping. Feature extraction finds key elements from data like corners or certain objects. Data association finds similarities between current and last data. Loop closure means data refinement to reach more robust result. After the first three stages, if successful, the pose has been found and mapping of area based on current frame can be done.

SLAM does not require the use of depth data but it is preferred practice when mapping environment. In some cases the depth data has even been used in feature extraction stage [P1, P2, P3, P7, P8, P13, P14, P16, P18]. This usually means extracting simple formations, like planes, from the data. One article suggested using depth data in a similar way the color data is commonly used: By finding corners and edges [P7].

Not all tracking methods are purely based on SLAM and neither are all of the methods presented in this reviews selected articles. SLAM does not exclude the use of prebuilt maps but it is not in the definition. SLAM does the tracking comparison to initial frame where the tracking begun. Some articles do tracking by using premapped data [P8, P12]. Methods that use point clouds directly to build formations for comparison are usually so different from SLAM process that they are not really part of it [P14].

3.3 3D SLAM methods

3.3.1 KinectFusion and its extensions

Most of the papers presented RGB-D SLAM algorithms for reconstruction and pose estimation. After the milestone paper [P15] presenting the tracking with KinectFusion algorithm (in 2011), several improvements and extensions have been proposed. KinectFusion first extracts the point cloud data from the environment and maps the surfaces of the objects into proper locations by ICP (Iterative-Closest-Point) algorithm. In the reconstruction step a TSDF (Truncated Signed Distance Functions) method is used for fusing partial depth maps. In general, SDF (Signed Distance Function) is a depth representation that is negative if the point is closer than the estimated surface and otherwise positive. TSDF filters out distances that are either too far or too near to the estimated surface. Unlike traditional RGB tracking, the KinectFusion can operate even in complete darkness due to the use of depth data.

The original KinectFusion algorithm has been extended in many ways. In [P4] signed distance functions were proposed directly for 3D reconstruction without traditional mapping required by the ICP algorithm. The application scenario was the position control of an autonomous quadrocopter. With regard to pose estimation, in [P10] RGB-D tracking method suitable for embedded devices was proposed. As 3D SLAM methods typically require a large working memory, the approach in [P10] was to use robust estimation to allow the mapping of IMU (e.g. gyroscope, accelerometer) data for the tracking. Two successive frames of the RGB-D data were used directly for tracking without a global reconstruction of the environment to reduce the memory footprint.

In [P20] and [P21] the KinectFusion algorithm was extended to larger scale environments. In [P20] the full mapping of a two story apartment was performed. The FOVIS (Fast Odometry from Vision) system was considered in [P20] as an alternative to ICP, which is prone to fail when there are not enough details in the environment. FOVIS uses sparsely sampled feature correspondences detected by the FAST algorithm. In [P21] the Kintinuous, which was originally proposed in [P20] was extended to operate with surface coloring and GPU acceleration of the visual odometry algorithm.

3.3.2 Monte carlo localization

Sequential monte carlo localization (particle filter) was proposed for determining the pose in [P8] with an application to large scale indoor navigation. The principle was first to construct a model of the environment by extracting large planar regions such as walls. To relax the requirements for the localization the dimension of the particle filter was reduced using an assumption of horizontal motion (usage of a ground robot). The FOVIS visual odometry algorithm was used with only data from an RGB-D sensor. Given the previous measurements, the particle filter then estimated the most probable next pose of the system and updated this according to new measurements. In [P14], monte carlo localization was used for real-time navigation and obstacle avoidance with an application to RGB-D tracking of a Nao humanoid robot. The system was also able to correctly react to non-static obstacles.

3.3.3 Plane filtering methods

A common problem in RGB-D SLAM methods is their high computational complexity, which generally requires a desktop computer equipped with a modern GPU. The number of points generated by Kinect, for example, becomes very large in dynamic operation and handling this may require excessive usage of the memory. Simplifying the computation with limited hardware resources may be needed in embedded and resource constrained applications. One way to reduce the computational load is to make assumptions on the structure of the environment, e.g. in extracting only dominant planes from it [P1, P2, P3]. The authors in [P2] used a RANSAC type method

FSPF (Fast Sampling Plane Filtering) to extract planes and mapped these to the existing 2D maps of the environment. In general, the utilization of RANSAC allows rapid and effective filtering of the depth data and it can efficiently operate in a resource constrained environment. The same authors in [P3] described real-time operation of RGB-D SLAM with full 640x480 depth images with relaxed CPU requirements. The application environment was a real-world indoor scene. With Kinect, for example, an increase in depth also increases the noise, and it becomes severe in the limits of the operation range. Finding dominant planes can also be used to filter out this noise. The authors in [P1] used RANSAC for this purpose by finding plane and point correspondences with an application to RGB-D SLAM.

3.3.4 Loop closure and operation in large scale environments

In [P9] a switching heuristics was used to determine automatically whether to use the RGB or the depth image for tracking. As the operation distance of low cost 3D sensors is quite restricted (e.g. <6m), it is advantageous to switch between the operation modes of RGB (monocular) tracking and RGB-D (depth based) tracking. As heuristics for determining which mode to use, depth association was used among others, i.e. finding out how well the depth map aligns with the RGB image structure. If it matches poorly, the scene is likely to be far away and monocular tracking is selected. Also the poor operation of ICP in the case when there were not enough details was accounted for so that the RGB cues were used in that case.

SLAM loop closure means detecting the places where the observer has visited before based on e.g. feature correspondences from selected keyframes. The purpose is to compensate the drift caused by noisy features of the sensors by accommodating the tracked path to the previous – probably more correct – location. Graph-based slam framework [P6] allows seeing the path of the observer with nodes as visited places, and edges which model the relation between each node as a transformation matrix. When a loop closure is detected the corresponding nodes can be updated to eliminate the drift. It should be observed that the corrected nodes (with a loop closure) can also be used as a base for future loop closures [P6].

The authors in [P17] considered SLAM in large scale environments. As stated, 3D SLAM is generally computationally intensive and may suffer from linear to cubic computation time in the number of variables. Also the memory usage may be heavy in many cases. In [P17] a double window optimization framework was used to allow the accurate mapping of local environment, while also taking account the structure of the environment in larger scale. The inner window maps local BA (Bundle Adjustment) to handle small scale tracking scenarios, while the outer window optimized the relations between the keyframes in a global graph. In [P17] both monocular and RGB-D applications were considered.

4 Results and discussion

The tracking methods and their operation environments among the selected papers are shown in Table 2. It can be observed, that loop closure has been implemented in about half of the papers. The applications of KinectFusion (KF) algorithm are generally limited to room (R) and small (S) to medium (M) size indoor environments. KinectFusion algorithm was used in six of the selected papers. The methods which take advantage of monocular cues also, can operate in larger scale environments. Three papers used plane filtering and they can achieve very high operation framerates. The other methods (denoted as "other" in Table 2), such as tracking based on brightness change of consecutive frames and frame-to-model registration approaches can operate with a small memory footprint.

Microsoft Kinect was used in most of the papers (as the RDB-d sensor), but due its compact size also Asus Xtation Pro Live has been applied in application scenarios requiring compactness. Most of the papers focused on localization (L) while eight papers also considered the reconstruction of the environment (as both processes are mutually linked in SLAM).

When considering the limitations of the methods, starting from the actual depth sensors to the algorithms that handle the data, the tracking process, in general, has room for improvements. Depth sensors today are still limited by weather conditions and useable operation range. In addition, no depth sensor has been embedded to mobile devices, which means that depth devices still have some mobility issues.

Because of the three dimensional nature of the depth data, the data is slow to process and thus requires extra calculation power from the CPU or GPU. This would require some level of filtering or simply: more powerful computers (and mobile devices). Other limitation is the core memory of the computing device used. By using SLAM related solutions, a lot of dynamic memory is required. For instance, KinectFusion can only model and track a very small cube in front of the sensor. In many cases, the SLAM derived system uses voxel (a 3D-pixel) mapping to recreate environment in digital format. To fully understand this problem, let's take a cube with sides of 1 meter. Divide it to 1mm sided cubes, and you get 1 billion small voxels. Storing the state of each of these cubes takes a lot of memory (in this case, at least 1 giga bit). What happens if you want larger space? You can expand the cube, which makes the voxels bigger and use larger but slower memory units (HDD and SSD) which eventually makes the algorithm slower.

Depth sensors are still used very much like conventional cameras. Tracking is done with the color information and in some cases depth tracking is done the same way. Problem in this is that being the newer technology, depth sensors are not very accurate in terms of actual measured distances. A color camera can have a very precise pixel by pixel level accuracy of colors whereas depth sensor pixels can have much more error in value. If you detect a corner in one frame by using depth data, you may not find it in a logical place in next frame anymore. This causes problems when trying to estimate correlations between frames. Plane filtering is a step in to the right direction as it eliminates errors in individual pixels. Going further in this kind of thinking could be the next step. The entire environment could be presented with planes, curves and other objects.

Paper no.	Application[1]	3D sensor type	Computing platform / Operation speed	Env.[2]	Tracking methods[3]	Loop closure[4]
[P1]	L+R	Microsoft Kinect	Intel Core i7 PC / 10Hz (3 Hz with map update)	IM	P (points and planes)	Yes (only for planes)
[P2]	L	Kinect + Hokuyo URG-04LX rangefinder	single CPU, high frame rate	IL	P (FSPF)	N/A
[P3]	R	Microsoft Kinect	Single CPU, high frame rate (400 Hz)	IM	P (FSPF)	No
[P4]	R	Asus Xtation Pro Live	laptop+Quadro GPU	IM	KF	N/A
[P5]	L	Stereo camera	Intel Core 2 Duo T9300 / 30Hz	IL	Extended Kalman Filter SLAM	No
[P6]	L	Microsoft Kinect	i7-2600K CPU / OpenCV / PCL / 19Hz	RS	Other / Graph based	Yes
[P7]	L	Microsoft Kinect	Desktop PC Dual Quad Core Xeon CPU / 10Hz	RS	Other / 2D morphology + ICP	No
[P8]	L	Microsoft Kinect + LIDAR	Laptop + Quadro 1700M GPU / 10Hz	IM	M (particle filter)	N/A
[P9]	L	Microsoft Kinect	Robot assist plarfotm (ACRA 2010)	IM + O	Other / RGB / RGB-D (ICP) heuristics	Yes (if depth available)
[P10]	L	Microsoft Kinect (TUM dataset)	single CPU, small memory footprint / 30Hz	IM	Other / brightness change + IMU	N/A
[P11]	L	Microsoft Kinect	Laptop Intel i7, OpenCV+ROS+g2o	IL	Other / graph based 2D to 3D	Yes
[P12]	L+R	Microsoft Kinect	Inter Core i7 + Nvidia GT 560 / >20Hz	RS	Other / 3D RANSAC + ICP	Yes
[P13]	L+R	Microsoft Kinect	Intel i7 CPU + ROS / 5-10Hz	IM	KF + Graph based g2o	Yes
[P14]	L	Asus Xtation Pro Live	Quad code PC / 6Hz	RS	M (particle filter)	N/A
[P15]	L+R	Microsoft Kinect	CPU + GPU	RS	KF (original)	Yes
[P16]	L	Microsoft Kinect	Laptop PC	RS	Other / 2D + 3D combination + IMU	No
[P17]	L	PrimeSensor	Desktop PC Core 2 Duo	OL	Other / keyframe + monocular	Yes
[P18]	L	Microsoft Kinect	N/A / Real time	RS	KF + color	N/A
[P19]	L+R	Asus Xtation Pro Live	Core 2 Duo + 2xARM7 + IMU / 30Hz (1kHz IMU)	IM	Other / frame to model registration	Yes
[P20]	R	Microsoft Kinect	Desktop PC + GeForce GTX 560 GPU	IM	Extended KF	Yes
[P21]	L+R	Microsoft Kinect (Freiburg dataset)	Desktop PC + GeForce 680GTX GPU / 15-30Hz	IM	Extended FK + FOVIS+Color	N/A

Table 2: Article main points

[1] Localization (L), Reconstruction (R)

[2] Environment: Room (R), Indoor (I), Outdoor (O), Small (S), Medium (M), Large (L) scale

[3] Kinect Fusion (KF), Plane Filtering (P), Monte Carlo (M), Inertial Measurement (IMU)

[4] Yes/No/ N/A

5 Conclusions

This paper provided a systematic literature study on the usage of 3D sensors in augmented reality tracking solutions. The research suggests that the field of tracking based on depth data is increasingly important, and thus many high quality works have been proposed in the literature. Perhaps the most important individual study has been the KinectFusion algorithm, which has been extended in several ways. Plane and point based filtering methods – on the other hand – can provide efficient integration to embedded hardware, which is essential if real-time performance is needed and compact size with low power consumption are searched for. Most of the methods in this review were implemented within a room or medium scaled indoor environment. However, by switching between the operation modes of monocular and depth modes (RGB<->D), some of the papers which were proposed could also operate in larger outdoor settings. Loop closure, which means finding out the locations which have earlier been visited to and updating the tracked pose accordingly, was also frequently taken advantage in the papers.

Acknowledgements

The research has been carried out during the MARIN2 project (Mobile Mixed Reality Applications for Professional Use) funded by Tekes (The Finnish Funding Agency for Innovation) in collaboration with partners; Defour, Destia, Granlund, Infrakit, Integration House, Lloyd's Register, Nextfour Group, Meyer Turku, BuildingSMART Finland, Machine Technology Center Turku and Turku Science Park. The authors are from Technology Research Center, University of Turku, Finland.

Bibliography

Selected papers

[P1] Esra Ataer-Cansizoglu et al. "Tracking an RGB-D camera using points and planes". In: *Proceedings of the IEEE International Conference on Computer Vision* (2013).

[P2] Joydeep Biswas and Manuela Veloso. "Depth camera based indoor mobile robot localization and navigation". In: *Proceedings - IEEE International Conference on Robotics and Automation* (2012).

[P3] Joydeep Biswas and Manuela Veloso. "Planar polygon extraction and merging from depth images". In: *IEEE International Conference on Intelligent Robots and Systems* (2012).

[P4] Erick Bylow et al. "Real-Time Camera Tracking and 3D Reconstruction Using Signed Distance Functions". In: *Robotics: Science and Systems (RSS) Conference* (2013).

[P5] Simone Ceriani et al. *Single and Multi Camera Simultaneous Localization and Mapping Using the Extended Kalman Filter.* 2014.

[P6] Hsi-yuan Chen and Chyi-yeu Lin. "RGB-D Sensor Based Real-time 6DoF-SLAM". In: *International Conference on Advanced Robotics and Intelligent Systems* (2014).

[P7] Ivan Dryanovski et al. "Real-Time Pose Estimation with RGB-D Camera". In: *IEEE International Symposium on Mixed and Augmented Reality* (2012).

[P8] Maurice F. Fallon, Hordur Johannsson, and John J. Leonard. "Efficient scene simulation for robust monte carlo localization using an RGB-D camera". In: *Proceedings - IEEE International Conference on Robotics and Automation* (2012).

[P9] Gibson Hu et al. "A robust RGB-D SLAM algorithm". In: *IEEE International Conference on Intelligent Robots and Systems* (2012).

[P10] Christian Kerl, Jurgen Sturm, and Daniel Cremers. "Robust odometry estimation for RGB-D cameras". In: *Proceedings - IEEE International Conference on Robotics and Automation* (2013).

[P11] Jan Helge Klüssendorff et al. "Graph-based visual SLAM and visual odometry using an RGB-D camera". In: *9th International Workshop on Robot Motion and Control, RoMoCo 2013 - Workshop Proceedings* (2013).

[P12] Donghwa Lee, Hyongjin Kim, and Hyun Myung. "GPU-Based Real-Time RGB-D 3D SLAM". In: *9th International Conference on Ubiquitous Robots and Ambient Intelligence* (2012).

[P13] Tong Liu et al. "A robust fusion method for RGB-D SLAM". In: *2013 Chinese Automation Congress* (2013).

[P14] Daniel Maier, Armin Hornung, and Maren Bennewitz. "Real-time navigation in 3D environments based on depth camera data". In: *IEEE-RAS International Conference on Humanoid Robots* (2012).

[P15] Richard a Newcombe et al. "KinectFusion : Real-Time Dense Surface Mapping and Tracking". In: *IEEE International Symposium on Mixed and Augmented Reality* (2011).

[P16] László Somlyai. "Mobil robot localization using RGB-D camera". In: *ICCC 2013 - IEEE 9th International Conference on Computational Cybernetics, Proceedings* (2013).

[P17] H. Strasdat et al. "Double Window Optimisation for Constant Time Visual SLAM". In: *IEEE International Conference on Computer Vision (ICCV)* (2011).

[P18] Po Chang Su, Ju Shen, and Sen Ching S Cheung. "A robust RGB-D SLAM system for 3D environment with planar surfaces". In: *2013 IEEE International Conference on Image Processing, ICIP 2013 - Proceedings* (2013).

[P19] Roberto G Valenti et al. "Autonomous Quadrotor Flight Using Onboard RGB-D Visual Odometry". In: *2014 IEEE International Conference on Robotics and Automation* (2014).

[P20] Thomas Whelan et al. "Kintinuous: Spatially extended kinectfusion". In: *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras* (2012).

[P21] Thomas Whelan et al. "Robust Tracking for Real-Time Dense RGB-D Mapping with Kintinuous". In: *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras* (2012).

Other references

[O1] H Durrant-Whyte and Tim Bailey. "Simultaneous localization and mapping". In: *Robotics & Automation Magazine, IEEE* (2006).

[O2] B.A. Kitchenham and S. Charters. "Guidelines for Performing Systematic Literature Reviews in Software Engineering". In: (2007).