"University of Turku Technical Reports, No. 8 — August 2015"

RECENT ADVANCES IN MONOCULAR MODEL-BASED TRACKING

A Systematic Literature Review

Olli Lahdenoja | Rami Suominen | Tero Säntti | Teijo Lehtonen

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

Rami Suominen
University of Turku, Technology Research Center, 20014 Turun yliopisto, Finland
rajusuo@utu.fi

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

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

www.trc.utu.fi

ISSN 2341-8028 | ISBN:978-951-29-6215-0

Technology Research Center

Turun yliopisto
University of Turku

Abstract

In this paper, we review the advances of monocular model-based tracking for last ten years period until 2014. In 2005, Lepetit, et. al, [19] reviewed the status of monocular model based rigid body tracking. Since then, direct 3D tracking has become quite popular research area, but monocular model-based tracking should still not be forgotten. We mainly focus on tracking, which could be applied to augmented reality, but also some other applications are covered. Given the wide subject area this paper tries to give a broad view on the research that has been conducted, giving the reader an introduction to the different disciplines that are tightly related to model-based tracking. The work has been conducted by searching through well known academic search databases in a systematic manner, and by selecting certain publications for closer examination. We analyze the results by dividing the found papers into different categories by their way of implementation. The issues which have not yet been solved are discussed. We also discuss on emerging model-based methods such as fusing different types of features and region-based pose estimation which could show the way for future research in this subject.

Keywords

monocular, model-based tracking, cad model, 3D, point cloud, pose estimation, augmented reality, a review

Contents

1 Introduction

Model-based monocular visual tracking can be used to obtain the position and the orientation (pose) of the observer when a complete or partial model of the environment pre-exists. The rise of compact depth sensors (such as Microsoft Kinect) has made the acquisition of 3D point cloud model from the environment easier than before. However, these sensors still face many limitations related to the operation environment (such as operation in bright sunshine) and obtained maximum depth acquisition distances. Also, obtaining a point cloud model of the environment beforehand does not preclude using model-based monocular visual tracking for obtaining the pose. Applications of model based tracking include Augmented Reality (AR), robot navigation (e.g. automotive), robotic object manipulation, and others. A very basic example of model based tracking implementation is finding corresponding points or lines between the image and the 3D model. In the initialization phase preliminary correspondences are found manually or automatically and in the tracking phase pose is constantly followed, for example by maintaining the correspondences when the observer moves. The pose can be resolved from the correspondences, for example, using an appropriate iterative solver (e.g. [8]).

The paper is divided into five sections. After the introduction, the research questions and the focus areas are defined in Section 2. The analysis of the selected papers is located in Section 3, which considers initialization, feature extraction methods, feature generation, feature filtering and other main aspects of the found material. Section 4 summarizes the main results, and discussion and conclusions are given in section 5.

2 Research questions and the literature search implementation

The way of implementing this survey is a Systematic Literature Review (SLR), which means that the search protocol is documented and repeatable (to a degree) to avoid bias caused by single persons opinion [17]. Instead of covering all detailed aspects of different model-based tracking methods, our focus is to find a representative and complete set of papers which would provide the reader a broad view on the different disciplines related to model-based tracking. Thus, we try to address the main issues in designing a complete model based tracking system. The research questions for this study are defined in the following;

- **to identify the current state-of-the-art for model based tracking in the literature**
- **to identify the weaknesses of monocular model based tracking**
- **to identify the emerging approaches in the field of model based tracking**

The research questions are quite broad in terms of which kind of papers are to be included to the search. Therefore, in order to avoid the search from widening too much in terms of number of papers and amount of work, we used only three most important academic search machines. These were IEEExplore, ACM Search, and Springerlink Search. There are also some other well known search databases, such as ProQuest and ISI web of knowledge which could have been included (however, mostly containing collections of other databases), but as mentioned, since it was not possible in this case to judge the suitability of the articles in question for further analysis based on title only, a large number of papers were screened already at the early stages of the search, thus increasing the required amount of work.

The search query used was the following;

(("edge based" OR "line tracking" OR "model based tracking" OR "corner detection" OR "natural feature")) AND (pose OR position* OR track* OR loca*).

The searches were performed in November and December in the year 2014. The search was limited to metadata/abstract search of the articles published in years 2004-2014. The number of tentative matches in IEEExplore search was (1067), in

ACM search (56), and in Springerlink search (2515). The Springer search was divided in two parts, Computer science articles, and Engineering articles, which were partially overlapped. Also, the article search emphasizing journal articles was used in the case of Springerlink search, due to very large number of matches of the conference chapter search. Searches were also performed using Google with typical queries related to model based tracking in order to complement the original search. The number of articles selected for the abstract reading phase was; IEEE (190), ACM (4) and Springer (140). Two of the final 23 papers were selected through the Google search. In the case of final paper selection there were two persons involved to avoid bias, so that at least one person had to recommend the inclusion of a paper and the other to agree. The selected papers will be discussed in this review paper in more detail. As a limitation of the study some relevant conference papers may still have been not accounted for, for instance for reasons like accessibility restrictions.

Five related studies of the subject area were identified, and those are briefly introduced here. Feature detectors have been surveyed in [18], which describes the properties of different kind of local features. An experimental survey of different tracking methods have been recently published in [14], it focuses on different approaches which define a bounding box around a given target and evaluates them using standard datasets. In [9] ten years of ISMAR papers were surveyed with a wide scope also including other aspects than pure AR tracking. SLAM methods were surveyed in [12]. In that work different SLAM methods were classified in terms of sensing device, SLAM algorithm, detector type and used image descriptors. Appearance models for visual object tracking were surveyed in [20]. In general, it appears that most of the existing surveys focus on object tracking with fixed datasets, and stationary camera while model-based tracking specific to AR with moving camera has not been recently extensively studied.

3 Analysis of the selected papers

3.1 Initialization and recovery

Initialization is performed when the tracking starts, and it means solving an initial pose of the system without a priori knowledge of the previous states of the system. Pose recovery means that if the tracking is lost, the pose can be recovered based on, for instance, information on the previous correct poses or their corresponding features. In [P7] the authors present a semi-automatic initialization method, which is based on taking a photograph from the target object and defining four corresponding point pairs between the image snapshot and the 3D model. Then the system back-projects features from the images to the 3D model by ray casting, assuming that more succesfull correspondences can be found. The SIFT (Scale-Invariant Feature-Transform) [10] features were used, and the pose was estimated with the POSIT algorithm [8] while utilizing RANSAC [3].

Some automated methods have been recently proposed to detect if the track (pose) has been lost. For example, occlusion can cause feature points to be lost, and detecting whether this happens can improve the tracking reliability [13]. In [13] the locations of the features in consecutive frames both forwards in time (i.e. t. t+1, t+2...) and backwards in time (i.e. t+2, t+1, t...) were considered and if the location of a feature at the time instant t is the same despite of the direction of the time-flow the feature point was declared to be valid. In [4] [5] a whole Track-Evaluate-Correct (TEC) framework was proposed for measuring the performance of different types of tracking failure detection methods.

The authors in [P8], while not implementing an initialization to the system, used a frame store to prevent the tracker from locking into a wrong pose when occlusion happens. The features were stored from a maximum of 10 frames to the frame store and were used for evaluating the reliability of the new features from the upcoming frames by accelerated SSD (Sum-of-Squared-Differences) match between stored features and query features. The approach of [P14] was to implement a complete model-based frame store (key frames) accross the test video sequence off-line, where information on the features on specific planar image regions were stored as keyframe images. To reduce memory consumption, the keyframe images were downsampled into a coarse resolution. In [P9] the authors perform initialization on objects that are assumed to be square sized, and the initialization was implemented by utilizing a priori information based on objects appearance in the 2D image and 3D model. While some approaches for model-based initialization can be found, it generally appears that the literarure lacks a working solution, which would not require excessive

extra work by the user. A more comprehensive survey of model-based initialization can be found in [2].

3.2 Low-level feature extraction

One common way of feature representation for tracking is the usage of uniformly or non-uniformly sampled line edge gradients across the direction of the edge normal, e.g. the moving edges algorithm [P10], [P8], [P11], [P4], [P22] and [P3]. It was pointed out in [P11] that this method is computationally simple, since it can further be reduced into separate convolutions applied to the whole image. In [P10] this approach was extended in an adaptive way so that the number of samples among the edges was determined recursively, depending on the distance of the feature edges. Also, in detecting the sample points multiple hypothesis method was used (i.e. also other points near to the edges were included for consideration), which resulted in more robust edge detection. A typical approach to adjust the edge normals to the 3D model has been iterative optimization, where an error function is minimized, see e.g. [P11] and [P22]. Also canny edge detector has been used in many studies, especially when the approach is to project the 3D model edges to image edges, and to solve the pose this way, see [P17].

Point based approaches for model-based tracking include corner detectors, such as FAST, Harris and SUSAN [P6], [P8], [P1], [P12], [P5], [P15], [P22] and [P14]. In [P5] a machine learning approach to efficient corner detection was proposed. Typically, feature vectors are then generated from the feature neighborhoods and matched across successive frames. In some studies, the corners have been divided into specific saliency levels depending on their strength, e.g. [P12]. For example, the authors in [P22], use a sum of squared difference between the RGB colour channels of small patches to calculate the matches. It was indicated in [P22], that the matching of the features this way on-line is computationally expensive and the number of features is limited.

Also optical flow methods such as KLT [P13], [P16] and region based (level-set) methods such as [16], [P23] can be incorporated into a model-based tracking system. Robust matching of keypoints achieving full or partial scale invariance have also been proposed [P7], [P12], [P4], [P20], [P13] and [P16]. A survey on the tracking performance of the keypoint descriptors was included to this study for completeness [P20], although it is more related to keyframe tracking. The paper [P20] measured the performance of the local descriptors for tracking of planar textures, in the presence of image transformations such as rotation, motion blur and lighting changes. The study was restricted into planar scenes, since the homography can only be succesfully estimated from them. In [16] region based method for obtaining the pose from level-set evolution constrained by image and 3D model was proposed.

The weaknesses of the aforementioned (point based and edge/line based) methods include that point based methods may fail when there is not adequate level of texture in the scene, and on the other hand, edge/line based methods work best with objects having less clutter (plain texture). An approach to solve this has been to fuse point based and edge/line based approaches complementarily to provide better performance in both settings [P6]. In [P6] the feature points were first matched to provide robustness to larger motions, and fine-tuning of the pose was performed using the line features. In [P2] the assumption that the objects are textureless was made to improve the suitability of the tracking to heavy background clutter. It was

also discussed in [P2], that the region based methods such as [16] are computationally intensive, and may require the utilization of GPU for efficient implementation.

3.3 Model creation and feature generation from/into 3D model

In some of the included papers the model of the environment was a simple line sketch [P6], [P22]). In papers [P18] and [P14] a more detailed model was produced as the outcome of the tracking. In general, the SLAM approaches create a model of the environment simultaneously with tracking (simultaneous *localization* and *mapping* [P18]). The authors in [P8] used a coarse but textured model of outdoor scenes, utilizing information on the details of the model features projected onto the image plane using a GPU (Graphics Processing Unit). In [P23] there actually did not exist a model, but the properties of the regular planar regions searched for UAV landing places were known.

In [P14] the application area for AR tracking was a large industrial plant. These kind of areas can cover several square kilometers. It was estimated that generating even just a one model of a small part of the plant could require hours of work, which is unfeasible. Also, it was pointed out that even if CAD models would be available they can be out-of-date and not contain the necessary information that the tracking would require. Some studies have tried to solve these problems, (e.g. [P4]), where sharp edges from any kind of polygon mesh model were extracted for tracking. Naturally, the size of the mesh model need to be reasonable. For completeness, the paper [P21] presents some other issues that need to be considered when a mesh model is utilized for tracking. First, any industrial product is usually designed by multiple companies with several different CAD packages. The conversion of the CAD native format may break the geometric data and there may also be other issues related such as confidentiality. Thus, a simple and general format for extracting the features is needed. One possible future direction would be feature extraction from point cloud data, which may be available in some cases [P21].

3.4 Feature filtering, sensor fusion and SLAM approaches

In order to obtain reliable feature correspondences across consecutive frames feature filtering is required. For feature filtering M-estimators [15], Expectation maximization (EM) [1] and RANSAC (Random Sample Consensus) [3] have been proposed. M-estimators have been used in [P7], [P10], [P11], [P4], [P22] and [P2] to obtain reliable correspondences. The M-estimators have usually been combined with IRLS (Iteratively Reweighted Least Squares) framework. The idea is that the error related to the correspondences is minimized so that each correspondence contribute differently to the overall estimate. Expectation maximization was used in [P6], and RANSAC and its variants have been used in papers [P7], [P4], [P20] and [P13]. RANSAC takes random samples from the correspondences defining a hypothesis model iteratively and evaluates how well the remaining correspondences suite to this hypothesis. If there are enough matches to support the hypothesis it is chosen and the process is terminated. The authors in [P23] used spatio-temporal MSER (Maximally-Stable Extremal-Regions) with shape regularity constraint to find landing places for UAVs

in structured environments. This paper was included to the study to point out the possibility to use man-made structures as a constraint for increasing the reliability of model-based feature extraction.

Especially in AR applications it is unconfortable if the successive pose estimates contain temporal jitter. Therefore the pose measurements need to be filtered. Pose filtering can be accomplished with Extended Kalman Filter (EKF) [P8], [P12], [P19], Iterated Extended Kalman Filter (IEKF) [P22], Expectation Maximization (EM) [P6] or particle filter [P17]. Each of these methods have their own special characteristics. The authors in [P6] used a Gaussian Mixture-Model (GMM) representation for the distribution of the model edges so that narrow gaussians were used when the correspondences were correct and very broad gaussians when they were incorrect. The pose filtering was based on Expectation Maximization (EM), which is an iterative method to estimate the probability distribution of the variables. In [P8] gyroscope, accelerometer and magnetic field sensor were integrated for model-based tracking framework with EKF. IEKF was used in [P22] to integrate model-based and model-free cues to tracking. IEKF is capable of handling highly non-linear measurements. It was pointed out in [P22], that using a particle filter would require projecting the model for each hypothesis, which is time consuming and can evidently lead to fewer feature edges tracked on-line, thus causing problems with highly textured objects. A succesful particle filtering approach for pose estimation has been later proposed in [P17].

SLAM (Simultaneous Localization and Mapping) approach has been incorporated in some of the selected papers [P18], [P12], [P14] and [P19]. Particularly interesting, the authors in [P18] model the 3D model edges as instances of Unscented Kalman Filter (UKF), so that the model which is used for tracking is constantly updated by new reliable information from the 2D scene. If a particular line edge is visible for enough time, the feature is updated into the 3D model. The operation of the system was demonstated in rather simple planar scenes with high edge contrasts. The authors in [P12] propose SLAM integrated with EKF, and in [P19] different parametrizations of the coordinates for EKF were studied. A parametrization means that the Euclidean coordinates are converted to a slightly other form in order to make the monocular depth estimates to fill the requirements of the EKF. To recover the camera trajectory the authors in [P14] used the freely available scalable monocular SLAM framework in [7]. Other monocular SLAM approaches include for example the one presented in [11].

3.5 Tracking in constrained and unconstrained environments

A model based tracking system can be divided in parts such as feature extraction from 2D images, feature generation from/into the 3D model, and the optimization methods for pose calculation and feature filtering. In order to obtain a working system each of these parts need to operate at a sufficient reliability level. For instance, if corners are found from images, but they do not correlate well to the corners extracted from the 3D model the task of correspondence matching might end up impossible despite of robust filtering and pose optimization. On the other hand, if the system assumes that it is used in an environment which contain only a certain type of texture, the wider utilization of the tracker in unconstrained environments may not be possible. Similarly, assumption of textureless objects restricts the us-

age. The implementation of a reliable model-based tracking system is challenging and the operation environment still needs to be fixed at a certain level in order to maintain hopes of a working tracking solution.

Given a target application area, purpose of use, and operation environment, when should model based tracking be applied? Even if a 3D model of the environment would pre-exist it is not certain that model-based tracking would be the best choice. If the 3D model is constrained and the application area is limited to a suitable scope such as industrial robotic manipulation of objects, the model based tracking could be preferred since it can estimate the location of objects very accurately [P22]. The more complex and unconstrained the environment and the 3D model are, the more difficult it will be to apply model-based tracking for the task. However, it is quite unlikely that the application area would not contain any constraints (e.g. man-made structures, lines and splines).

3.6 Real-time operation and GPU acceleration

In [P17] the runtime of the particle filter based tracking algorithm was compared between CPU only and GPU (CUDA) implementation. One of the critical issues in utilizing the GPU for tracking was the extra time needed for CPU-GPU data transfer [P17], [P16]. A useful property of GPUs taken advantage in several papers was z-culling, i.e. the capability of determining which surface is nearest to the observer. For instance in [P16] z-culling was used to a slightly different task to mask regions before transferring them to the CPU and in [P9] to separate the object from background. Most of the selected tracking papers mention that they run in real-time. However, it is evident that this requires many simplifications and shortcuts for the algorithms for specific hardware platforms. When the tracking system is optimized for real-time operation for specific application and platform (e.g. mobile) it may be difficult to port it directly into another kinds of environments. However, some of the methods are available for use as source codes, e.g. [P11], [P16] which can give a positive impact on the wider usage of model-based tracking.

Also the different types of feature generation methods can take advantage of GPUs. In 3D-2D projection based trackers, the pose of the model needs to be aligned to the 2D view within a sufficient accuracy in order to achieve the correct match. This requires more computation power from GPU in order to consider the different views and managing them. If edges or corners are extracted from a 3D mesh automatically (e.g. using a GPU), the important question is which kind of criteria to use to obtain only the edges which really correspond to strong edges in the monocular view, and not the other. This kind of reliable feature extraction from 3D models could be a potential future direction for research.

3.7 Application areas

The application areas of the papers included to this survey are divided into following broad categories. Indoor tracking - including simple object tracking to accomplish the pose estimation, as well as indoor navigation [P7], [P10], [P6], [P18], [P11], [P9], [P17], [P13], [P22], [P3] and [P2]. Applications related to robotic object manipulation are defined as another category, these are addressed in papers [P11], [P4], [P22]. Tracking in industrial environment, see e.g. papers [P7] and [P14]. Finally, outdoor tracking and navigation in outdoor settings (including some aerial tracking systems)

are covered in papers [P8], [P11], [P12] and [P23]. Naturally, these divisions are coarse, and the tracking methods can be applicable to many separate categories.

4 Results

In Table 1, all of the selected papers have been divided into categories according to their initialization method, feature representation (extraction, generation, filtering), and hardware acceleration. It appears (Table 1), that initialization and/or recovery has been implemented in only four papers. Corners have been used as main features in 11 and line/edges in 13 papers. Importantly, five of the papers considered using the combination of these both. Feature generation methods could further be divided on the basis in which direction the features are mapped. Feature correspondences may be ray casted into the 3D model through some image plane location, such as keypoint or derived from the 3D model towards the image plane (see the paragraph feature generation). It can also be observed, that most of the papers that are using a direct 3D-2D projection take advantage of GPU processing.

The first research question of the study was to find the current state-of-the-art of model-based tracking in the literature. Edge/line based methods, especially the ones taking sample points of the edge gradients to the direction of the edge normal have been popular. This RAPID style method is not new, but it is still widely used. Regarding the automated generation of new features (3D-2D correspondences), some approaches use direct 3D-2D correspondence matching e.g. from the visible edges of the 3D model, or features derived from the texture of the model. One of the included papers provided a method to extract edge information from any kind of polygon mesh, which is particularly interesting [P4]. Also, model-based SLAM has been proposed to generate the 3D model on-line from reliably detected lines of the 2D view. For feature filtering, although not being a new method, M-estimators [15] have shown robustness and are still very popular (Table 1).

The second research question was to identify the weaknesses of model-based tracking. One apparent weakness of monocular model-based tracking is the challenge of reliable feature extraction from mesh models. This has been quite rarely addressed in the literature. As mentioned, this is not trivial and the models of the environment may be incomplete, inaccurate, out-of-date, or not easily accessible for reasons such as compability issues of CAD formats or confidentiality. Another weakness of model-based tracking include, that the current implementations only work in quite restricted environments, where the amount of edge clutter needs to be sufficiently small. A general model-based tracking system, that would give the observer free access to manouver in complex environments and operate reliably in cluttered scenes has not been proposed in the literature. The examples of operation environments usually restrict to certain types of limited sized objects. Also, as mentioned the initialization and recovery has been quite rarely addressed in the found articles (Table 1), and even then it required certain amount of manual work. On the other hand, some successful methods have been proposed in order to handle pose recovery and the validity of the feature tracking paths [13] [5] [4].

The third research question was related in identifying which kind of emerging solutions have appeared that could improve model-based tracking. An emerging approach which has been proven successfull is to use many different types of features (Table 1), e.g. model-based and model-free cues [P22], lines and points [P6], etc. Fusion methods, which take advantage of many different trackers could also be an option, however, with the cost of increased system complexity and increased time-lag (affecting to the AR user experience). Also, there might be other issues involved, when using multiple methods simultaneously, such as the effect of switching between different operation modes. Other emerging methods include region based pose calculation [16], which could gain more success in the future when the GPUs evolve (as the methods are quite computationally intensive, [P2]). Some of the feature extraction methods (e.g. keypoint extraction) have been implemented with GPU as open source implementations (e.g. [P16]), which can facilitate the real-time implementation of model-based tracking. For instance, in [6] a method for extracting direct SIFT correspondences between the monocular camera view and realistic rendering of the 3D view of point cloud data was proposed. Also, some new corner detection methods have been introduced (e.g. [P1], [P15]), and it would be interesting to test, whether they could give robustness on the other earlier proposed tracking systems.

5 Discussion and Conclusions

5.1 Discussion

Model-based tracking can benefit from the advancements of other tracking types such as keyframe tracking. Therefore, some additional methods (such as keypoint based tracking) were briefly considered in this study as well. It would appear that today the mainstream approach for visual monocular tracking is the keyframe method. A challenge related to monocular model-based tracking is that without a working solution for (re)initialization, the system would need to operate at a high reliability level to give a satisfactory user experience in AR, for instance. On the other hand, in the case of model-based tracking there is not need to collect pre-recorded training data from the environment as in keyframe tracking. Still, keyframe tracking may provide better user experience if the training phase could be made straightforward. Utilizing the widely available pre-recorded point cloud data from the environment (e.g. RGB-D) and direct 2D-3D feature correspondences could be an interesting future research direction in tracking with monocular cameras [6]. Also, new compact sensor technologies (e.g. 3D, mobile, gyros, accelerometers) have improved their performance, which could help the implementation of reliable model-based tracking systems.

5.2 Conclusions

This paper reviewed the recent progress on monocular model-based tracking. The following main observations were made; the approach of using uniform sampling points across edges and their normals seem still to be one of the most commonly used feature extraction methods in model-based tracking. It was also observed, that the initialization and recovery was rarely addressed in the selected papers, which is a drawback in many tracking methods. When considering the tracking aspects only, making a decision on whether to apply model-based tracking in a given purpose seems not trivial. One clue based on the literature search could be to consider how constrained the model and the environment are. The more constrained they are, the more likely model-based tracking could possibly success. Besides aspects related to tracking only, the designer of a tracking system also needs to consider issues related to the accessibility of the model, conversion of CAD model file formats and information security, which are emphasized in systems operating on mobile devices and which utilize a model-based tracking approach.

#	Initialization	Feature extraction			Feature generation	GPU acceleration	Feature filtering	#
	Init.(i)/Recov.(r)	Corner	Line/Edge	R/KeyP./Opt.F.				
P1		x	x					P1
P2			x		textureless 3D object		M-Est. (+IRLS)	P2
P3			x		3D-2D projection	x (Mobile)		P3
P4			x	Keypoint(K)	any 3D polygon mesh		RANSAC/M-est.	P4
P5		x		Keypoint (K)				P5
P6		x	x		line model (by hand)		GMM / EM	P6
P7	i			Keypoint (K)	ray casting to 3D model	(OpenSG)	M-est., RANSAC	P7
P8	r	x	x		from textured 3D model	x	M-est., EKF (+gyro)	P8
P9	i			Opt. flow (OF)	3D-2D projection/GPU	x	KF	P9
P10			x		adaptive from 3D model	x	M-est.	P10
P11			x				M-est., VVS (servoing)	P11
P12		x		Keypoint (K)			EKF/SLAM	P12
P13		x		K / OF	(keyframe)		(keyframe)	P13
P14	i/r	x	x		model-based keyframe		SLAM [7]	P14
P15		x						P15
P16				K /OF		x (Open source)		P16
P17			x		3D-2D projection	x (CUDA)	Particle filter	P17
P18			x		automatic model creation		RANSAC/UKF/SLAM	P18
P19		x					EKF-SLAM	P19
P20		x		Keypoint (K)			RANSAC	P20
P21			x		edges from 3D mesh			P21
P22		x	x		lines + model free		IEKF/M-Est.	P22
P23				K / Region (R)				P23
Total	4	11	13	9		7		

Table 1: The selected articles are divided into different categories based on how initialization and recovery are handled, by the type of features used, feature generation, GPU acceleration and feature filtering. The total number of papers in each of the categories is also shown.

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] Willis A. and Sui Y. "An Algebraic Model for fast Corner Detection". In: *International Conference on Computer Vision* (2009), pp. 2296–2302.

[P2] Seo B-K., Park H., and Park J-I. et. al. "Optimal Local Searching for Fast and Robust Textureless 3D Object Tracking in Highly Cluttered Backgrounds". In: *IEEE Transactions on Visualization and Computer Graphics* 20.1 (2014), pp. 99–110.

[P3] Seo B-K., Park J., and Park J-I. "3-D Visual Tracking for Mobile Augmented Reality Applications". In: *International Conference on Multimedia and Expo (ICME)* (2011), pp. 1–4.

[P4] Choi C. and Christensen H. I. "Real-time 3D Model-based Tracking Using Edge and Keypoint Features for Robotic Manipulation". In: *International Conference on Robotics and Automation* (2010), pp. 4048–4055.

[P5] Rosten E., Porter R., and Drummond T. "Faster and Better: A Machine Learning Approach to Corner Detection". In: *IEEE Transactions on Pattern Analysis andd Machine Intelligence* 32.1 (2010), pp. 105–119.

[P6] Rosten E. and Drummond T. "Fusing Points and Lines for High Performance Tracking". In: *International Conference on Computer Vision* (2005), pp. 1508–1515.

[P7] Bleser G., Pastarmov Y., and Sticker D. "Real-time 3D Camera Tracking for Industrial Augmented Reality Applications". In: *Proceedings of Graphics, Visualization and Computer Vision, WSCG* (2005), pp. 47–53.

[P8] Reitmayr G. and Drummond T. W. "Going out: Robust Model-based Tracking for Outdoor Augmented Reality". In: *International Symposium on Mixed and Augmented Reality (ISMAR)* (2006), pp. 109–118.

[P9] Ruiter H. and Benhabib B. "Visual-model-based, realtime 3D pose tracking for autonomous navigation: methodology and experiments". In: *Autonomous Robots* 25.3 (2010), pp. 267–286.

[P10] Wuest H., Vial F., and Sticker D. "Adaptive Line Tracking with Multiple Hypotheses for Augmented Reality". In: *International Symposium on Mixed and Augmented Reality (ISMAR)* (2005), pp. 62–69.

[P11] Comport A. I., Merchand E., and Pressigout M. et. al. "Real-Time Markerless Tracking for Augmented Reality: The Visual Servoing Framework". In: *IEEE Transactions on Visualization and Computer Graphics* 12.4 (2006), pp. 615–628.

[P12] Artieda J., Sebastian J. M., and Campoy P. et. al. "Visual 3-D SLAM from UAVs". In: *Journal of Intelligent Robotic Systems* 55.4-5 (2009), pp. 299–321.

[P13] Kim J., Park C., and Kweon I. S. "Vision-based navigation with efficient scene recognition". In: *Intelligent Service Robotics* 4.3 (2011), pp. 191–202.

[P14] Neubert J., Pretlove J., and Drummond T. "Rapidly constructed appearance models for tracking in augmented reality applications". In: *Machine Vision and Applications* 23.5 (2012), pp. 843–856.

[P15] Awrangjeb M., Lu G., and Fraser C. S. "A Comparative Study on Contour-based Corner Detectors". In: *International Conference on Digital Image Computing (DICTA)* (2010), pp. 92–99.

[P16] Sinha S. N., Frahm J-M., and Pollefeys M. "Feature tracking and matching in video using programmable graphics hardware". In: *Machine Vision and Applications* 22.1 (2011), pp. 207–217.

[P17] Azad P., Münch D., and Asfour T. et. al. "6-DoF Model-based Tracking of Arbitrary Shaped 3D Objects". In: *International Conference on Robotics and Automation* (2011), pp. 5204–5209.

[P18] Gee A. P. and Mayol-Cuevas W. "Real-Time Model-Based SLAM Using Line Segments". In: *Advances in Visual Computing* (2006), pp. 354–363.

[P19] Ceriani S., Marzorati D., and Matteucci M. et. al. "Single and Multi Camera Simultaneous Localization and Mapping Using the Extended Kalman Filter - On the Different Parametrizations for 3D Point Features". In: *Journal of Mathematical Modelling and Algorithms in Operations Research* 13.1 (2014), pp. 23–57.

[P20] Gauglitz S., Höllerer T., and Turk M. "Evaluation of Interest Point Detectors and Feature Descriptors for Visual Tracking". In: *International Journal of Computer Vision* 94.3 (2011), pp. 335–360.

[P21] Yamakawa S. and Shimada K. "Polygon crawling: feature edge extraction from general polygonal surface for mesh generation". In: *Engineering with Computers* 26.3 (2010), pp. 249–264.

[P22] Kyrki V. and Kragic D. "Tracking rigid objects using integration of model-based and model-free cues". In: *Machine Vision and Applications* 22.2 (2011), pp. 323–335.

[P23] Sun X., Christoudias C. M., and Lepetit V. et. al. "Real-time landing place assessment in man-made environments". In: *Machine Vision and Applications* 25.1 (2014), pp. 211–227.

Other references

[1] Dempster A. and Laird N. et. al. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistics Society* 39 (1977), pp. 1–38.

[2] Euranto A., Lahdenoja O., and Suominen R. et. al. "Model-based tracking initialization in ship building environment". In: *University of Turku Technical Reports* 2 (2014).

[3] Fischler M. A. and Bolles R. C. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". In: *Communications of the ACM* 24.6 (1981), pp. 381–395.

[4] T. A. Biresaw, Cavallaro A., and Regazzoni C.S. "Correlation-based self-correcting tracking". In: *Neurocomputing* 152 (2014), pp. 345–358.

[5] T. A. Biresaw, Soto Alvarez M., and Regazzoni C.S. "Online failure detection and correction for Bayesian sparse feature-based object tracking". In: *IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)* (2011), pp. 320–324.

[6] Sibbing D., Sattler T., and Leibe B. et. al. "SIFT-Realistic Rendering". In: *International Conference on 3D Vision* (2013), pp. 56–63.

[7] Eade E. and Drummond T. "Scalable Monocular SLAM". In: *IEEE Conference on Computer Vision and Pattern Recognition* (2006), pp. 469–476.

[8] Dementhon D. F. and Davis L. S. "Model-based object pose in 25 lines of code". In: *International Journal of Computer Vision* 15.1-2 (1995), pp. 123–141.

[9] Zhou F., Duh H., and Billinghurst M. "Trends in Augmented Reality Tracking, Interaction and Display: A Review of Ten Years of ISMAR". In: *International Symposium on Mixed and Augmented Reality (ISMAR)* (2008), pp. 193–202.

[10] Lowe D. G. "Distinctive image features from scale-invariant keypoints". In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110.

[11] Davidson A. J., Reid I. D., and Molton N. D. et. al. "MonoSLAM: Real-Time Single Camera SLAM". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.6 (2007), pp. 1052–1067.

[12] Fuentes-Pacheco J., Ruiz-Ascencio J., and Rendon-Mancha J. M. "Visual simultaneous localization and mapping: a survey". In: *Artificial Intelligence Review* (2012), pp. 1–27.

[13] Z. Kalal, Mikolajczyk K., and Matas J. "Forward-Backward Error: Automatic Detection of Tracking Failures". In: *International Conference on Pattern Recognition* (2010), pp. 2756–2759.

[14] Smeulders A. W. M., Chu D. M., and et. al. Cucchiara R. "Visual Tracking: An Experimental Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.7 (2014).

[15] Huber P.J. "Robust estimation of a location parameter". In: *Ann. Math Stat.* 35 (1964), pp. 73–101.

[16] Dampreville S., Sandhu R., and Yezz Y. et. al. "Robust 3D pose estimation and efficient 2D region-based segmentation from a 3D shape prior". In: *European Conference on Computer Vision* (2008), pp. 169–182.

[17] Keele S. "Guidelines for performing systematic literature reviews in software engineering". In: *EBSE Technical Report EBSE-2007-01* 15.1-2 (2007), pp. 123–141.

[18] Tyutelaars T. and Mikolajczyk K. "Local Invariant Feature Detectors: A Survey". In: *Foundations and Trends in Computer Graphics and Vision* 3.3 (2007), pp. 177–280.

[19] Lepetit V. and Fua P. "Monocular model-based 3d tracking of rigid objects: A survey". In: *Foundations and trends in computer graphics and vision* (2005), pp. 1–89.

[20] Li X., Hu W., and Shen C. et.al. "A Survey of Appearance Models in Visual Object Tracking". In: *ACM Transactions on Intelligent Systems and Technology* 4.4 (2013).