Simultaneous Localization and Mapping In Smart Bikes

ALEX VAN GEMUND, University of Twente, The Netherlands

Cycling can be a dangerous form of transportation with many casualties around the world. With this new era of smart technologies, why do we not make bikes smarter and let technology assist us with cycling safely. Simultaneous Localization And Mapping (SLAM) is already being used for autonomous vehicles but can potentially be a useful technology for assisting cyclists to drive more safely. This research will conclude to what extent this SLAM technology can be used in smart bikes. First studying which SLAM technology and algorithm fits this smart bike environment best and creating an easy to grasp overview of the two SLAM problems and its paradigms. Secondly discovering what possible applications this SLAM technology can have and what problems it can potentially solve. While also discussing visual odometry as a similar but alternative technology.

Additional Key Words and Phrases: SLAM, Localization, Mapping, Smart Bikes, Biker's Safety, Cycling.

1 INTRODUCTION

Cycling is a widely used form of transportation and is rapidly growing, however being so exposed and vulnerable is very dangerous. Research done by ETSC (European Transport Safety Council) shows that in the period of 2010-2018, cyclist fatality in the EU has only changed with an average annual decrease of 0.4%. However, countries such as the Netherlands and Ireland, well developed countries, are topping the charts in terms of an increase in cycling fatalities.[1]

One way to make cycling safer could be to make it smarter, there have been several studies on smart IoT integrated bikes in order to make cycling a safer method of transportation.[28][38][2][34]. Another potential method to make bikes smarter can be through the use of SLAM. SLAM is mainly being used for autonomous vehicles or robots with the research of Wen et al. about the performance of SLAM for autonomous vehicles in diverse typical driving scenarios of Hong Kong being an example [39].

SLAM (Simultaneous Localization and Mapping) consists of mapping the environment and locating itself within that mapped environment [10]. It perceives landmarks via a variety of possible sensors, LIDAR, monocular or RGB-D to name a few and builds a model of the environment with the collected information [14]. Then it simultaneously finds its pose (position and orientation) within that environment.

There a several existing SLAM technologies that all have several algorithms to chose from. Each technology or algorithm has their own strong points and is fit for a different purpose. With some of the early and first back-ends being EKF SLAM, FastSLAM and GraphSLAM[4][17][37]. The first two being online and the last one being offline SLAM, this is one way to categorize solutions to the SLAM problem. The difference between the two is that online SLAM needs to process as it is going, needs to be fast and can thus be sloppy, whereas offline slam is used to refine the overall map quality and be very accurate[16].

The amount of research on the SLAM problem is vast, the amount of research on smart bikes is significantly less and the amount of research on the combination of the two is almost non-existent. Thus stressing the new and mostly unexplored grounds this research will explore.

In this research, existing literature on SLAM technologies and algorithms will be used to find which would suit the environment of smart bikes the best in terms of assisting the cyclist. But also to see what problems this chosen algorithm(s) could potentially solve. To eventually make traveling with a bicycle a safer method of transportation.

2 PROBLEM STATEMENT

The safety of cycling is concerning and by making bikes smarter and assisting the cyclist, this transportation method can perhaps be made safer. SLAM could potentially be a method to help with making a bike smarter, however very little to no research has been done on this topic. This paper will therefore analyse the SLAM technologies and algorithms to discover which would be best suited for the application on smart bikes and what problems it could solve.

2.1 Research question

With the aforementioned problem statement we can get to the main research question: to what extent can SLAM technologies be used in smart bikes?

To answer this question it will be split up into a larger first sub question and smaller second sub question.

1. Which SLAM algorithms would be the most useful to the smart bikes environment?

2. What are the possible applications of this SLAM technology and is this goal also reachable via other technologies?

3 RELATED WORKS

In this section we will go over the related works in the field of SLAM combined with smart bikes.

There is no literature for using SLAM in smart bikes in order to assist the cyclist and make cycling safer, hence why this research is being done. But there are various similar papers either talking about SLAM in autonomous bicycles or cars, or smart technology combined with bicycles to make them safer.

In 2017, Stasinopoulos et al.[33] presented an article with a comprehensive theoretical framework for applying SLAM in bicycles and akin vehicles. However this is with the intention of creating autonomous bicycles and not something that assists the cyclist. Even though it is about autonomous two wheeled vehicles, they claim that this study could be the basis for more detailed research with specific regards to environment perception and SLAM.

TScIT 37, July 8, 2022, Enschede, The Netherlands

There are also multiple works offering solutions or potential solutions to improve the safety of bikers [28][38][2][34] with the use of smart technology. They all address the safety issue of cyclists and have similar ideas on how to approach this. Either with a smart helmet, mobile or other device and use this to detect a specific scenario. With the papers ranging from regular collisions to stopped cars opening their doors in the biker lane. These papers are also after the improvement of the biker's safety, however not through the use of SLAM technologies.

4 METHODOLOGIES

In this part we will discuss the methodology of this research, addressing how the research question and both sub questions will be handled.

The structure of this research is as follows, first doing some general research to understand SLAM better. Then secondly deciding which algorithm(s) could be the best for the smart bikes environment. Thirdly, seeing what the possible applications of this SLAM technology are and what problems will be solved using this. Then finally discussing the research question and drawing a conclusion with the previously gathered information.

4.1 Sub question 1

Firstly, to see which SLAM algorithm(s) would be most suitable for the SLAM bikes environment, we first need to assess which SLAM technology or technologies would be best suited in order to reduce the number of algorithms needing to be researched. This is necessary since there are too many to individually do research on. Once a fitting SLAM technology or technologies has been identified, algorithms can be thoroughly research in order to decide on the best one(s). This data will be collected through the use of multiple data bases such as Springer, IEEE or google scholar, however most of the information will be gathered via Scopus.

4.2 Sub question 2

Here we discuss the possible applications of the previously decided algorithm(s) on the smart bike environment. But also identify where and how the use of this SLAM algorithm could help improve the safety of riding a bicycle. Next to this, another technology and its potential for this same environment will also be discussed. Useful data and information will be gathered through the same means as mentioned in the previous sub question.

5 SLAM

The literature on the basics of SLAM and its categories is very slim and the information on this topic will mainly be gathered from a paper called the Handbook Of Robotics and then specifically the chapter about SLAM, written by Stachniss et al[32]. We will go over the difference in the two problems and see which problem will be most applicable to the smart bike environment. After this the different SLAM paradigms will be discussed and assessed based on the applicability on the smart bike environment.

5.1 Online vs Full SLAM

The SLAM problem can split up into two different problems, namely the online SLAM problem and the Full SLAM problem. Full SLAM estimates and computes the posterior of the entire path while also creating a map. These Full SLAM algorithms process all of the data at the same time which makes it very computationally expensive. Online SLAM only computes the current pose instead of the entire path and its algorithms can process one data item at a time, which makes them computationally faster.

Now assessing which technology would be most suitable for the smart bike environment. SLAM on a bicycle would be used to assist the driver with decisions and make the driver aware of its surroundings. These decisions will need to be made fast since it would need to alert the driver ahead of time. Also, there is no real need for computing the entire path since the main goal is to be helpful in present times and not to create a map of where the driver has been. Therefore, the applicable SLAM problem to this smart bike environment would be the Online SLAM problem.

5.2 SLAM paradigms

This section will go over the three main SLAM paradigms and assess which of these has the best potential applicability.

5.2.1 EKF SLAM. The oldest of the three and the one with the most influence on SLAM as a whole is EKF SLAM, with EKF standing for Extended Kalman Filter. With it being the oldest, it has become slightly disfavoured due to its narrowed computational abilities. While moving through a certain environment, the covariance matrix and the state vector are updated using this Extended Kalman Filter. With this method, the covariance matrix and the state vector grow quadratically, thus posing significant scaling limitations. As well as that the linearization could cause the system to produce inconsistent maps. Some extensions of this method are more promising, splitting the map into submaps and with that improving the scaling ability. However these algorithms are more similar to the graph-based method.

5.2.2 Graph based SLAM. The graph-based method addresses the Full SLAM problem and solves the problem in a graphical approach, as the name suggests. It visualizes the situation of location and landmarks as nodes in a graph. Every successive pair of locations is connected by an edge, this edge portrays the information provided by the odometry reading. The working of this method is well illustrated in figure 1(Siciliano et al. 2016). This is the main method for building large-scale maps and it is being supported by the SLAM community with open source licenses to aid further development. Although this method is used to solve the Full SLAM problem, multiple cross-overs exist that remove variables of the last location to change the graph to online, as shown by Bosse, Thrun and Paskin [7][23][36].

5.2.3 Particle filter. The third method revolves around particle filters, which has only been popularized for the last twenty five years or so and is the favoured method for the online SLAM problem. This method sees the environment as a set of particles, with every particle being thought of as a guess as to what the real value of the state potentially is. All these guesses are then collected into a

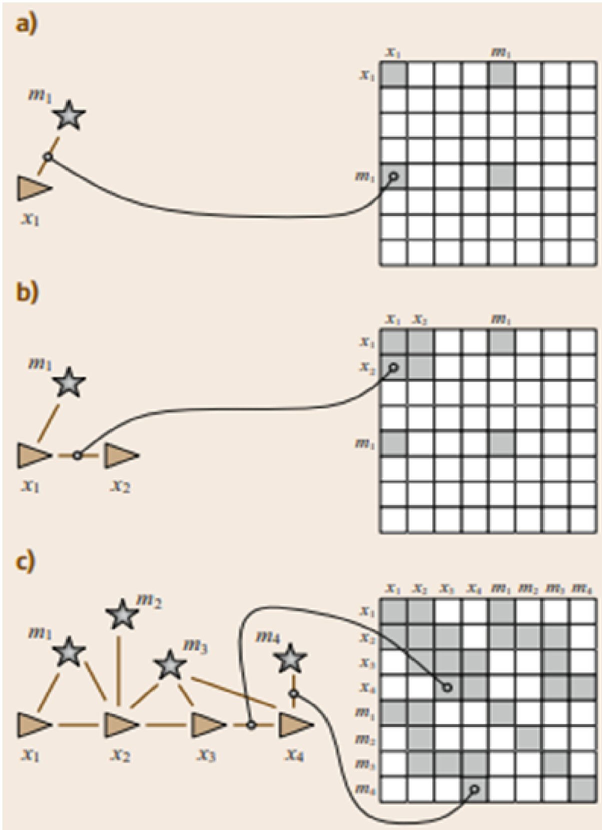

Fig. 1. Illustration of the graph construction. The (a) diagram shows the graph, the (b) the constraints in matrix form. (a) Observation ls landmark m1. (b) Robot motion from x1 to x2. (c) Several steps later
[8]

set of guesses, or a set of particles if you will, and this estimates the posterior distribution. There are two important downsides of this method, first of all the number of samples a system needs to produce consistent maps is often decided manually via an educated guess. Secondly, the combination of re-visiting previously mapped areas and nested loops can cause particle depletion. Which in turn will counter the making of consistent maps of the environment. However, this method also provides advantages with it being computationally efficient and able to sample over data associations.

When looking at the three paradigms and assessing their qualities, we at the base see that each method has their own advantages and disadvantages. The graph-based approach is suited for the full slam problem, apart from the few algorithms focused on the online problem as aforementioned. Therefore this approach, aside from the few online algorithms based on this approach, will not be the considered for this smart bike environment. EKF slam is a possibility, however with the potential computational issues and producing inconsistent maps as a consequence, the particle filter approach looks like the better option. This approach being most popular for the online slam problem recently and granting computationally efficient updates. Those quick updates would be very important to capture the real-time situations and be able to make fast and accurate decisions to secure the safety of the cyclist. Accordingly, a deeper dive into algorithms using the particle filter approach will be taken.

5.3 SLAM algorithms

In this section, several SLAM algorithms in line with the previously decided category will be discussed and assessed. These algorithms will be assessed based on the ability to handle dynamic environments and running in real-time, its computational abilities, accuracy, speed and the algorithm's own general advantages and disadvantages. This all in relation to the possible applicability on the smart bike environment.

5.3.1 CD SLAM. CD SLAM is an abbreviation for Continuous Localization and Mapping in a Dynamic world and is a method proposed by Pirker et al. in 2011 [26]. This is a visual SLAM method that deals with short and long term dynamic scenes in large environments using one individual camera. It applies viewpoint-dependent visibility information to certain map points and can therefore efficiently chose potentially visible points for a certain view. To handle the dynamic environment, the system gives a proper weight to each viewpoint dependent on how many times it is observed.

This system proposes several advancements compared to standard structure-from-motion approaches. It constructs a view-graph with camera poses to achieve loop closure and sliding window bundle adjustment. Secondly, it uses a different rule for keyframe selection in order to limit the amount of views added to the map. Furthermore, they have improved the Hoc descriptor (Histogram of Oriented Cameras)[25] to strengthen the data association in terms of robustness and speed.

This system was presented in 2011 and thus could not make use of the well-known KITTI dataset, that was introduced in 2013 [12], for their testing. The results from their tests are gathered from their own recordings and so are not universally comparable. However they detected that their mean deviation ranges from 3.06 to 38.08cm and that using the HoC descriptor, they reduced the matching effort by 81.87%. With this, they claim that the system can run in real-time when doing sliding window bundle adjustment and tracking, excluding the loop closing which would take about five seconds. However, it seems that this system is mainly focussed on mapping the environment without the dynamic obstructions and thus filtering those away.

5.3.2 CO SLAM. CoSLAM stands for Collaborative Visual Simultaneous Localization And Mapping. This SLAM system using multiple cameras was presented by Danping Zou and Ping Tan[41] in 2013. These cameras can even be placed on various platforms and move independently to together produce a global map. Although placing cameras on different platforms is beyond the scope of this research. With the use of these multiple cameras, they introduce intercamera mapping and intercamera pose estimation to be able to deal with dynamic environments. At this time, several systems were already presented using Visual SLAM with multiple cameras[13][22][24]. However, these systems were used for static environments and not

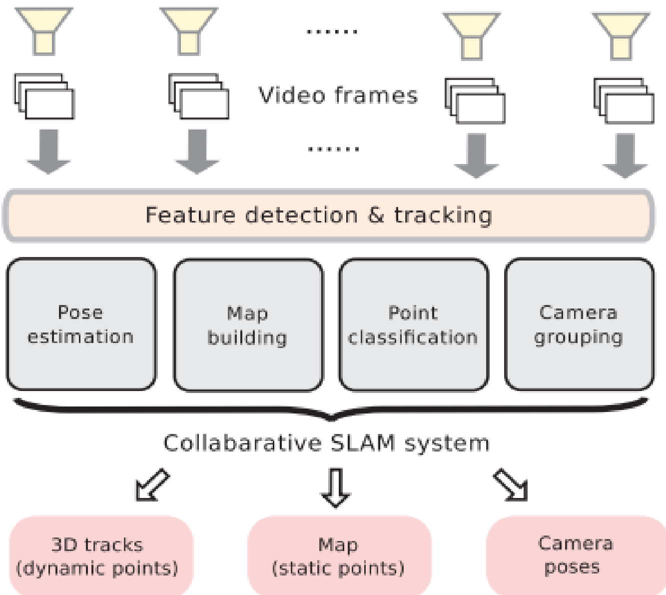

Fig. 2. CoSLAM system architecture
[11]

useable in dynamic environments. This system consists out of four components: camera pose estimation, map building, point classification and camera grouping. At every frame, the point classification and the camera pose estimation component group the points into different types and compute the camera poses, respectively. The map building component generates new map points regularly and the camera grouping split the cameras into different groups based on whether the cameras observe the same image or not. The system is well illustrated in figure 2(Tan et al. 2013).

They tested this system in static and dynamic scenes, this research will only be concerned with the dynamic tests. Two tests were concluded, one indoor dynamic test with the camera trajectories having an average length of 28.7 meters and one in a garden with a larger average length of 63 meters. The distance drift in the tests were 1.2 meters and 5.3 meters, respectively. This drift is quite large and too much to accurately decide on a cyclist's environment. Also according to the tests, the system's runtime was decently efficient with it taking 38 ms to process a frame with around a thousand map points. However, this testing was done offline with pre-recorded data. The paper mentions the aspiration for integrating a data capturing component in order to use this system online, but no evidence was found for this. This system also primarily focusses on the ability to merge cameras from different platforms, which is beyond the scope of this research.

5.3.3 ORB-SLAM2 and DynaSLAM. This system for monocular, stereo and RGB-D was presented in 2017 by Mur-Artal et al.[21] and is the successor of the feature-based ORB-SLAM[20]. It works in a multiple environments, ranging from being indoor with a handheld device to cars driving around town and flying drones. The back-end of this system is based on bundle adjustment (BA), it also has a lightweight localization mode that uses visual odometry for unmapped environments and connects to map points that leads to zero-drift localization. This system is also open source for the benefit of the slam community.

There are three main threads that run parallel from each other: tracking, local mapping and loop closing. This can cause a fourth thread to be created to perform full BA after the loop has been closed. The tracking part is to, at every frame, find feature matches to the local map to be able to localize the camera. Local mapping is used to update the local map and perform local BA to optimize it. Then the third thread, loop closing is to find big loops and perform pose-graph optimization to straighten out the compiled drift. This third thread starts a fourth thread were full BA is performed in order to find the motion solution and optimal structure.

The paper[21] also shows the testing that has been done with multiple datasets. We will be taking a look at the tests of the KITTI dataset since this dataset has recorded sequences from a car driving in urban areas. The outcomes were compared to, at this time, the state of the art stereo LSD-SLAM and the results are heavily in favour of ORB-SLAM2. With it outperforming LSD-SLAM in nine out of the eleven sequences and having an overall relative translation error of less than one percent, as shown in figure 3(Mur-Artal et al. 2017). Since this system is open-source, extensions are being built to further address specific issues. One of these extensions is called DynaSLAM and was presented in 2018 by Bescos et al.[6]This extension was created with the intention to handle highly dynamic situations even better than the original system. This was also tested using the same KITTI dataset with monocular and stereo sensors. Overall it was

TABLE I
COMPARISON OF ACCURACY IN THE KITTI DATASET

	ORB-SLAM2 (stereo)			Stereo LSD-SLAM		
Sequence	t_{rel} (%)	r_{rel} (deg/100 m)	t_{abs} (m)	t_{rel} (%)	r_{rel} (deg/100 m)	t_{abs} (m)
00	0.70	**0.25**	1.3	**0.63**	0.26	**1.0**
01	**1.39**	**0.21**	10.4	2.36	0.36	**9.0**
02	**0.76**	0.23	5.7	0.79	0.23	**2.6**
03	**0.71**	**0.18**	**0.6**	1.01	0.28	1.2
04	0.48	**0.13**	0.2	**0.38**	0.31	0.2
05	**0.40**	**0.16**	**0.8**	0.64	0.18	1.5
06	**0.51**	**0.15**	**0.8**	0.71	0.18	1.3
07	**0.50**	**0.28**	0.5	0.56	0.29	0.5
08	**1.05**	0.32	**3.6**	1.11	**0.31**	3.9
09	**0.87**	0.27	**3.2**	1.14	**0.25**	5.6
10	**0.60**	**0.27**	**1.0**	0.72	0.33	1.5

Fig. 3. Comparison of Accuracy between ORB-SLAM2 and stereo LSD-SLAM
[27]

Table 1. Overview of the different SLAM algorithms and their attributes

Algorithm	Handle dynamic environments	Run in real-time	Environment size	Open source
CD SLAM	Yes	Yes, without loop closing	Large, Outside	No
CO SLAM	Yes	No, only with pre-recorded data	Medium, Outside	Yes
ORB-SLAM2	Yes	Yes	Large, Outside	Yes
DynaSLAM	Yes	Yes, but not optimized	Large, Outside	Yes
RD SLAM	Yes	Yes	Small, Inside	No
MonoSLAM	Yes	Yes, for small trajectories	Medium, Inside	Yes
SLAM++	No	Yes	Large, Inside	No

TABLE V
COMPARISON OF THE RMSE OF THE ATE [M], THE AVERAGE OF THE RPE [%] AND THE RRE [°/100 M] OF DYNASLAM AGAINST ORB-SLAM2 SYSTEM FOR STEREO CAMERAS

Sequence	ORB-SLAM2 (Stereo) [1]			DynaSLAM (Stereo)		
	RPE [%]	RRE [°/100m]	ATE [m]	RPE [%]	RRE [°/100m]	ATE [m]
KITTI 00	**0.70**	**0.25**	**1.3**	0.74	0.26	1.4
KITTI 01	**1.39**	**0.21**	10.4	1.57	0.22	**9.4**
KITTI 02	**0.76**	**0.23**	**5.7**	0.80	0.24	6.7
KITTI 03	0.71	0.18	0.6	**0.69**	0.18	0.6
KITTI 04	0.48	0.13	0.2	**0.45**	**0.09**	0.2
KITTI 05	0.40	0.16	0.8	0.40	0.16	0.8
KITTI 06	0.51	**0.15**	0.8	**0.50**	0.17	0.8
KITTI 07	**0.50**	**0.28**	0.5	0.52	0.29	0.5
KITTI 08	1.05	0.32	3.6	1.05	0.32	**3.5**
KITTI 09	**0.87**	**0.27**	3.2	0.93	0.29	**1.6**
KITTI 10	**0.60**	**0.27**	**1.0**	0.67	0.32	1.2

Fig. 4. Comparison of ORB-SLAM2 and DynaSLAM
[5]

slightly less accurate than the original ORB-SLAM2, except for the highly dynamic sequences where it actually performed better, the results of both are shown and compared in figure 4 (bescos et al. 2018).

5.3.4 Other. There are several other algorithms that are worth mentioning, but not worth taking a deeper dive into due to obvious shortcomings. RD SLAM presented by Tan et al.[35] in 2013 showed potential to handle dynamic environments, however it could only operate in small spaces. MonoSLAM, presented a year later by Russo et al.[29], could make the leap to medium sized spaces but this was the limit and therefore also not fit for a cycling environment. Lastly, SLAM++ presented in 2013 by Salas-Moreno et al.[30] is able to handle larger environments, however this is algorithm is more suited for indoor applications such as large buildings.

5.3.5 Best algorithm. After assessing several algorithms based on their ability to handle dynamic scenes, running in real-time and various other qualities as mentioned in the introduction of the SLAM algorithms section, the best potential algorithm for the smart bike environment within the realm of this research will be chosen.

Table 1 shows an overview of the different SLAM algorithms and their abilities. It considers whether it can handle dynamic environments and if it can run in real-time with live footage. The environment size combined with the location is also shown, as well as whether the algorithm is open source or not. The three sizes all reflect different environments, with small representing the desk of an office, medium a room or a normal garden and large representing factory or a regular urban outside environment. The size, as well as the location of operation, is the main reason for the bottom three algorithms not being discussed elaborately.

CD SLAM and CO SLAM are both quite a bit older, being presented in 2011 and 2013 respectively. CD SLAM has the issue of trying to filter away the dynamic obstructions and is mainly focused on creating the map. Whereas the information necessary for the smart bike would be more focused on the environment at a certain time or small time frame, not on the mapped environment of a minute ago which holds no valuable information for the smart bike. The CO SLAM system also laid the focus elsewhere, this time on merging the footage of multiple cameras situated on multiple platforms and therefore causing it not to be able to run in real-time.

ORB-SLAM2 and its extension DynaSLAM are relatively newer, with them releasing in 2017 and 2018 respectively. This system's usability ranges from a handheld device to cars driving around a city, with cycling then also fitting within this range. Both the system and the extension were tested with a well-known data-set that resembles the situation of driving bicycles in urban areas and performed better than the state of the art system at that time. Another important quality of this system, is that it is open source and therefore is easy to get into, as well as find information on it. Both ORB-SLAM2 and DynaSLAM show qualities that, within the scope of this research, have been decided to be most important when searching for the SLAM algorithm that could potentially handle this environment. These two algorithms are very similar, where DynaSLAM outperforms ORB-SLAM2 slightly in highly dynamic environments, ORB-SLAM2 returns the favour in the other sequences. Both show potential to handle this environment and should therefore also both be tested to see where that slight edge in accuracy is more favourable.

6 VISUAL ODOMETRY

This research was started with zero knowledge on SLAM and the intention of finding out to what extent SLAM technologies can be used in a smart bike environment. During this research, other

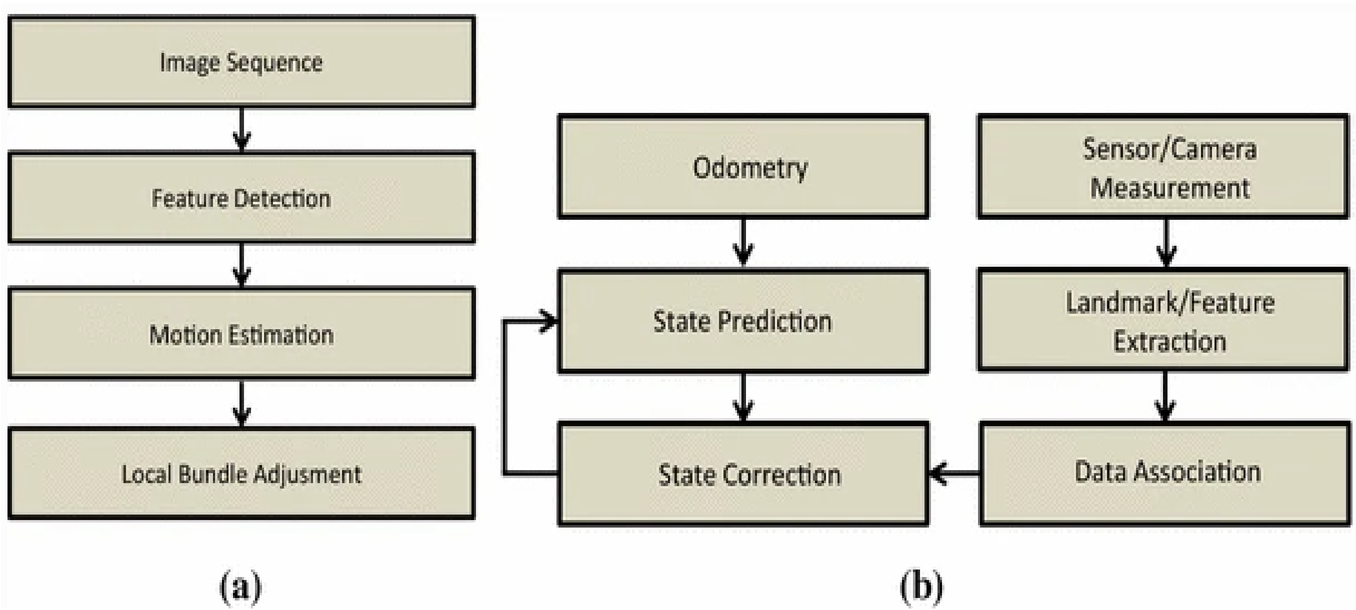

Fig. 5. The main components of: a) Visual Odometry and b) a filter based SLAM system [40]

methods also pass by that bare resemblance to SLAM, but are not quite the same. Even having qualities that would potentially fit this environment better than SLAM. The method that crossed this research often and will therefore also be assessed further, is visual odometry.

The paper by D. Scaramuzza[31] describes visual odometry nicely by saying it can be seen as building block of a SLAM algorithm. Visual odometry only cares about the local consistency of the path and therefore also only uses a local map to more accurately estimate this path, whereas SLAM is more involved with the global consistency of the map. VO recovers the path pose after pose and possibly optimizes over the last n poses, which is called windowed or local bundle adjustment. This method reduces drift and as shown by Konolige et al.[15], decreases the final position error by a factor of 2 to 5. We also see that this method was used in ORB-SLAM2 as well as CD SLAM. Figure 5(Bab-Hadiashar 2015) shows an overview of the main components of a VO and a filter based SLAM system, really showing the difference in complexity between the two.

The most important difference, and certainly within the scope of this research, between VO and SLAM is the difference in consistency, performance and complexity of implementation. VO trades off the global consistency for real-time performance, since it does not have to keep track of all the previous data, but just n amount of frames[31]. These differences that VO brings could actually be very beneficial for a smart bike environment. For this application, global map consistency is not of importance since it is of no use to know the environment of were the cyclist was a couple minutes ago. Thus with VO only looking at the last n frames fits this purpose way better and also improves the performance in real-time, which is one of the most important things since the speed of decision making is key for the cyclist's safety. VO is already being used in automotive applications and has quite a lot of literature, such as [19][9][18], or even papers on VO being used as an advanced driver-assistance system [3].

7 APPLICABILITY OF SLAM INTEGRATED SMART TECHNOLOGY IN A SMART BIKE ENVIRONMENT

The SLAM technology would be used to get information on the environment and the path of the cyclist and then with that information make quick decisions on the safety of the cyclist. However, the smart technology that would be combined with SLAM is beyond the scope of this research. The literature on the topic of smart bikes is actually quite minimal, where most of the smart bikes literature is about bike sharing and thus not relevant. The few papers that do address the right topic are also mentioned in the related works, but will be discussed here.

Clarke et al. published a paper that presented a safety management framework for IoT-integrated bikes[28]. This system works by communicating with other vehicles and performing calculations on whether a dangerous situation is about to occur. While this is an entirely different way of working, the goal is the same, improving the safety of cyclists. The main use case of the framework is vehicle collision avoidance as expected, since this is how most accidents occur. This will most likely also be the main use case for smart bikes when using SLAM technology, alerting the cyclist about potential collisions. However it does not need to be limited to this ability only, thinking of features such as recognizing whether the cyclist is close to the edge of the road, just like some cars give a warning when you almost cross a white line on the road. Another potential feature could be the detection of dooring, which is when still standing cars next to the bicycle lane still have passengers in them and then suddenly open the car door. Grauschopf et al.[38] tackled this issue and presented a paper in 2021, they did not specifically talk about the system they used, but rather about the usefulness of alerting cyclist on this specific issue and what would be the best way to alert them. Eventually concluding that it in general could support the cyclists avoiding opening doors. Thus adding another possible concern that the SLAM integrated smart bike should take into account. This shows the wide of range of dangerous situations that the system could possibly prevent.

8 CONCLUSION AND FUTURE WORK

In this paper, we researched what SLAM algorithm would be most suited for the smart bike environment, what other technologies could potentially do this task as well or perhaps even better and what the potential applicability of this SLAM integrated smart bike system could be. At the same time creating an easy to grasp overview of the different SLAM problems and paradigms. This literature research on SLAM algorithms lead to ORB-SLAM2 and DynaSLAM being the favourites, with them possessing the most sought after qualities in relation to this research. Another technology that is quite similar to SLAM, namely visual odometry, was also looked into further. Showing that it could possibly even be a better match for the smart bike environment than SLAM. Potential applications of the SLAM integrated smart bike system were also discussed and this presented several ways it could make cycling safer.

For future research, ORB-SLAM2 and DynaSLAM should be tested within the required environment to eventually pick one algorithm and continuing the development of the smart system. However, visual odometry should first deserve some further research in order to decide whether this technology would be more or less useful than SLAM.

REFERENCES

[1] 2020. HOW SAFE IS WALKING AND CYCLING IN EUROPE? 38 (Jan. 2020). https://etsc.eu/wp-content/uploads/PIN-Flash-38_FINAL.pdf

[2] Ajmain Inqiad Alam, Mahfuzur Rahman, Sharmin Afroz, Mahbubul Alam, Jia Uddin, and Md. Ashraful Alam. 2018. IoT Enabled Smart Bicycle Safety System. In *2018 Joint 7th International Conference on Informatics, Electronics & Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*. IEEE. https://doi.org/10.1109/iciev.2018.8641067

[3] Ignacio Parra Alonso, David Fernández Llorca, Miguel Gavilan, Sergio Álvarez Pardo, Miguel Ángel Garcia-Garrido, Ljubo Vlacic, and Miguel Ángel Sotelo. 2012. Accurate Global Localization Using Visual Odometry and Digital Maps on Urban Environments. *IEEE Transactions on Intelligent Transportation Systems* 13, 4 (Dec. 2012), 1535–1545. https://doi.org/10.1109/tits.2012.2193569

[4] Tim Bailey, Juan Nieto, Jose Guivant, Michael Stevens, and Eduardo Nebot. 2006. Consistency of the EKF-SLAM Algorithm. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. https://doi.org/10.1109/iros.2006.281644

[5] Berta Bescos and Jose M. Facil and Javier Civera and Jose Neira. 2018. DynaSLAM: Tracking, Mapping, and Inpainting in Dynamic Scenes. https://doi.org/10.1109/lra.2018.2860039 [Online; accessed june 13, 2022].

[6] Berta Bescos, Jose M. Facil, Javier Civera, and Jose Neira. 2018. DynaSLAM: Tracking, Mapping, and Inpainting in Dynamic Scenes. *IEEE Robotics and Automation Letters* 3, 4 (Oct. 2018), 4076–4083. https://doi.org/10.1109/lra.2018.2860039

[7] Michael Bosse, Paul Newman, John Leonard, and Seth Teller. 2004. Simultaneous Localization and Map Building in Large-Scale Cyclic Environments Using the Atlas Framework. *The International Journal of Robotics Research* 23, 12 (Dec. 2004), 1113–1139. https://doi.org/10.1177/0278364904049393

[8] Bruno Siciliano and Oussama Khatib. 2016. Springer Handbook of Robotic. https://doi.org/10.1007/978-3-319-32552-1 [Online; accessed june 2, 2022].

[9] Martin Buczko, Volker Willert, Julian Schwehr, and Jurgen Adamy. 2018. Self-Validation for Automotive Visual Odometry. In *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. https://doi.org/10.1109/ivs.2018.8500560

[10] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, Jose Neira, Ian Reid, and John J. Leonard. 2016. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics* 32, 6 (Dec. 2016). https://doi.org/10.1109/tro.2016.2624754

[11] Danping Zou and Ping Tan. 2013. CoSLAM: Collaborative Visual SLAM in Dynamic Environments. https://doi.org/10.1109/tpami.2012.104 [Online; accessed june 13, 2022].

[12] A Geiger, P Lenz, C Stiller, and R Urtasun. 2013. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research* 32, 11 (Aug. 2013), 1231–1237. https://doi.org/10.1177/0278364913491297

[13] Michael Kaess and Frank Dellaert. 2010. Probabilistic structure matching for visual SLAM with a multi-camera rig. *Computer Vision and Image Understanding* 114, 2 (Feb. 2010), 286–296. https://doi.org/10.1016/j.cviu.2009.07.006

[14] Muhammad Shahzad Alam Khan, Danish Hussain, Kanwal Naveed, Umar S. Khan, Imran Qayyum Mundial, and Anas Bin Aqeel. 2022. Investigation of Widely Used SLAM Sensors Using Analytical Hierarchy Process. *Journal of Sensors* 2022 (Jan. 2022). https://doi.org/10.1155/2022/5428097

[15] Kurt Konolige, Motilal Agrawal, and Joan Solà. 2010. Large-Scale Visual Odometry for Rough Terrain. In *Springer Tracts in Advanced Robotics*. Springer Berlin Heidelberg, 201–212. https://doi.org/10.1007/978-3-642-14743-2_18

[16] J. Lehmann. 2022. *The differences between online & offline SLAM*. Retrieved may 7, 2022 from https://www.exyn.com/news/differences-between-online-and-offline-slam

[17] Mingwei Lin, Canjun Yang, and Dejun Li. 2019. An Improved Transformed Unscented FastSLAM With Adaptive Genetic Resampling. *IEEE Transactions on Industrial Electronics* 66, 5 (May 2019). https://doi.org/10.1109/tie.2018.2854557

[18] Steven Lovegrove, Andrew J. Davison, and Javier Ibanez-Guzman. 2011. Accurate visual odometry from a rear parking camera. In *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. https://doi.org/10.1109/ivs.2011.5940546

[19] Aleksandr Mikov, Audren Cloitre, and Igor Prikhodko. 2021. Stereo-Vision-Aided Inertial Navigation for Automotive Applications. *IEEE Sensors Letters* 5, 2 (Feb. 2021), 1–4. https://doi.org/10.1109/lsens.2021.3049503

[20] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardos. 2015. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics* 31, 5 (Oct. 2015), 1147–1163. https://doi.org/10.1109/tro.2015.2463671

[21] Raul Mur-Artal and Juan D. Tardos. 2017. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics* 33, 5 (Oct. 2017), 1255–1262. https://doi.org/10.1109/tro.2017.2705103

[22] David Nistér, Oleg Naroditsky, and James R. Bergen. 2004. Visual odometry. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* 1 (2004), I–I.

[23] Mark Paskin. 2004. Thin Junction Tree Filters for Simultaneous Localization and Mapping. (06 2004).

[24] L.M. Paz, P. Pinies, J.D. Tardos, and J. Neira. 2008. Large-Scale 6-DOF SLAM With Stereo-in-Hand. *IEEE Transactions on Robotics* 24, 5 (Oct. 2008), 946–957. https://doi.org/10.1109/tro.2008.2004637

[25] Katrin Pirker. 2010. Histogram of Oriented Cameras - A New Descriptor for Visual SLAM in Dynamic Environments. In *Procedings of the British Machine Vision Conference 2010*. British Machine Vision Association. https://doi.org/10.5244/c.24.76

[26] Katrin Pirker, Matthias Ruther, and Horst Bischof. 2011. CD SLAM - continuous localization and mapping in a dynamic world. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. https://doi.org/10.1109/iros.2011.6094588

[27] Raul Mur-Artal and Juan D. Tardos. 2017. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. https://doi.org/10.1109/tro.2017.2705103 [Online; accessed june 13, 2022].

[28] M. A. Razzaque and Siobhan Clarke. 2015. A security-aware safety management framework for IoT-integrated bikes. In *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. IEEE. https://doi.org/10.1109/wf-iot.2015.7389033

[29] Ludovico Russo, Stefano Rosa, Basilio Bona, and Matteo Matteucci. 2014. A ROS Implementation of the Mono-Slam Algorithm. In *Computer Science &amp Information Technology (CS &amp IT)*. Academy & Industry Research Collaboration Center (AIRCC). https://doi.org/10.5121/csit.2014.4131

[30] Renato F. Salas-Moreno, Richard A. Newcombe, Hauke Strasdat, Paul H.J. Kelly, and Andrew J. Davison. 2013. SLAM: Simultaneous Localisation and Mapping at the Level of Objects. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. https://doi.org/10.1109/cvpr.2013.178

[31] Davide Scaramuzza and Friedrich Fraundorfer. 2011. Visual Odometry [Tutorial]. *IEEE Robotics &amp Automation Magazine* 18, 4 (Dec. 2011), 80–92. https://doi.org/10.1109/mra.2011.943233

[32] Bruno Siciliano and Oussama Khatib (Eds.). 2016. *Springer Handbook of Robotics*. Springer International Publishing. https://doi.org/10.1007/978-3-319-32552-1

[33] Sotirios Stasinopoulos, Mingguo Zhao, and Yisheng Zhong. 2017. Simultaneous localization and mapping for autonomous bicycles. *International Journal of Advanced Robotic Systems* 14, 3 (May 2017). https://doi.org/10.1177/1729881417707170

[34] S. J. Swathi, Shubham Raj, and D. Devaraj. 2019. Microcontroller and Sensor Based Smart Biking System for Driver's Safety. In *2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)*. IEEE. https://doi.org/10.1109/incos45849.2019.8951409

[35] Wei Tan, Haomin Liu, Zilong Dong, Guofeng Zhang, and Hujun Bao. 2013. Robust monocular SLAM in dynamic environments. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE. https://doi.org/10.1109/ismar.2013.6671781

[36] Sebastian Thrun, Yufeng Liu, Daphne Koller, Andrew Y. Ng, Zoubin Ghahramani, and Hugh Durrant-Whyte. 2004. Simultaneous Localization and Mapping with Sparse Extended Information Filters. *The International Journal of Robotics Research* 23, 7-8 (Aug. 2004), 693–716. https://doi.org/10.1177/0278364904045479

[37] Sebastian Thrun and Michael Montemerlo. 2006. The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures. *The International Journal of Robotics Research* 25, 5-6 (May 2006). https://doi.org/10.1177/0278364906065387

[38] Tamara von Sawitzky, Thomas Grauschopf, and Andreas Riener. 2021. "Attention! A Door Could Open."—Introducing Awareness Messages for Cyclists to Safely Evade Potential Hazards. *Multimodal Technologies and Interaction* 6, 1 (Dec. 2021). https://doi.org/10.3390/mti6010003

[39] Weisong Wen, Li-Ta Hsu, and Guohao Zhang. 2018. Performance Analysis of NDT-based Graph SLAM for Autonomous Vehicle in Diverse Typical Driving Scenarios of Hong Kong. *Sensors* 18, 11 (Nov. 2018), 3928. https://doi.org/10.3390/s18113928

[40] Yousif, K. and Bab-Hadiashar, A. and Hoseinnezhad, R. 2015. An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics. [Online; accessed june 20, 2022].

[41] Danping Zou and Ping Tan. 2013. CoSLAM: Collaborative Visual SLAM in Dynamic Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 2 (Feb. 2013), 354–366. https://doi.org/10.1109/tpami.2012.104