Application and evaluation of direct sparse visual odometry in marine vessels

Citation for the original published paper (version of record):

Nguyen, B., Blanch, K., Petersson, A. et al (2022). Application and evaluation of direct sparse visual odometry in marine vessels. IFAC-PapersOnLine, 55(31): 235-242.
http://dx.doi.org/10.1016/j.ifacol.2022.10.437

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-31 (2022) 235–242

Application and evaluation of direct sparse visual odometry in marine vessels

Björnborg Nguyen * Krister Blanch * Anna Petersson *
Ola Benderius * Christian Berger **

* *Chalmers University of Technology, Göteborg, Sweden (e-mail: {bjornborg.nguyen,krister.blanch,ola.benderius}@chalmers.se, annpeter@student.chalmers.se)*
** *University of Gothenburg, Göteborg, Sweden (e-mail: christian.berger@gu.se)*

Abstract:
With the international community pushing for a computer vision based option to the laws requiring a look-out for marine vehicles, there is now a significant motivation to provide digital solutions for navigation using these envisioned mandatory visual sensors. This paper explores the monocular direct sparse odometry algorithm when applied to a typical marine environment. The method uses a single camera to estimate a vessel's motion and position over time and is then compared to ground truth to establish feasibility as both a local and global navigation system. Whilst it was inconsistent in accurately estimating vessel position, it was found that it could consistently estimate the vessel's orientation in the majority of the situations the vessel was tasked with. It is therefore shown that monocular direct sparse odometry is partially suitable as a standalone navigation system and is a strong base for a multi-sensor solution.

Keywords: Monocular direct sparse odometry, Visual odometry, Heading tracking, Position estimation, Optical flow, Computer vision, Marine systems

1. INTRODUCTION

With rising autonomous drive development in vehicles on roads, in air, and at sea with both promising and demonstrated results, there has been a plethora of developed and proposed systems dealing with different pieces of the problems and challenges within autonomous driving. For such capabilities, self-localisation and pose estimation are as vital as route planning and motion control. With the market demand and regulations at play, there is a need for a robust vision system that can fulfil or complement the self-localisation of autonomous vehicles. For example, the international community calls for clarity regarding autonomous maritime vessels and *Rule 5*, which specifies the need for a look-out (Zhou et al., 2020) in the international regulations for the prevention of collisions at sea (COLREGs). This paper proposes that future autonomous systems will likely stem from vision being a mandatory requirement, and there is the need for a vision-based control suite with the ability to perceive itself and its surrounding environment robustly.

Since the conception of self-driving maritime vehicles, this has been achieved mostly through internal gyroscopic units, compasses, and satellite-based odometry. More recent attempts also use sensors that are expensive, such as *light detection and ranging* (LIDAR), or prone to noise, such as radar. With compasses and satellite systems also subject to interference, a solution is to use local systems that are independent of such variables, making a camera a reasonable alternative (Hayakawa and Dariush, 2019), even if a vision-based watch would not be mandatory. This method, known as visual odometry, is effective in various land-based platforms but is still quite limited in research on maritime platforms. From the little research that has been conducted, systems typically use stereo or composite imagery, or some form of cartography for comparison, which would not be feasible for micro navigation or investigation of a naval structure. Composite cameras would be an ideal setup, but for typical use cases, there is an upper limit of data throughput and compute capability (Benderius et al., 2021). Therefore, this paper looks at the use of a singular camera as a foundation, refines the concept of *direct sparse odometry* (DSO), applies it to a maritime platform, and provides analysis to show that DSO is a viable solution to the aforementioned problems.

1.1 *Visual odometry*

Odometry, or ego-motion, is the concept of estimating one's position, pose, or motion over time in a tangible and observable manner. For an observer, this odometry can be defined as the relation between the observer and its surroundings. In the case of visual odometry, the odometry is mainly determined by visual input. Such a system would greatly benefit from a visual feature-rich environment thus increasing the performance. A dynamic environment with independently moving objects would require a segmentation process in order to exclude them in visual odometry. Thus, visual motion may be categorised into two different types: the motion of the observer, and the motion of the object in the environment. Both can be moving simultaneously or individually, and both will have to be resolved

10.1016/j.ifacol.2022.10.437

and segmented to provide accurate odometry. When this is resolved with only a visual medium like a camera, this system now provides *visual odometry* (Khan and Adnan, 2017).

1.2 Marine use-case

For an autonomous surface vehicle to maintain compliance with the COLREGs, especially the aforementioned *Rule 5*, the vessel must maintain a proper look-out for the prevention of a collision. There are numerous digital systems aiding human operators, including radio-based automatic identification systems (AIS), and doppler radar, but none can work without some form of human supervision. A completely digital method would therefore employ the same capabilities of a human skipper, in that a visual system needs to be employed for navigation and object avoidance. The functional requirements for such a system would include some form of vision traversal, the ability to identify and diagnose objects of interest, and some form of control logic to establish a safe path. Whilst not completing this entire setup, determining the vessel's motion through visual odometry resolves a section of the control logic, and provides redundancy for the vessel's other digital systems whilst employing the sensors required by *Rule 5* of the COLREGs.

1.3 Optical flow

In littoral waters, where the risk of collision is typically highest, there is also the opportunity for the most data capture in a visual-based system. The ever-changing landscape of shoreline, surface objects and the vessel's orientation in three-dimensional (3D) space provides a significant challenge for a visual system, especially as camera sensors flatten this scene into a two-dimensional (2D) data frame. One common method to develop visual odometry, whilst resolving the issues of 2D data in a 3D space, is to exploit optical flow. The definition of optical flow is the changes of structural light caused by the relative motion of the observer and its environment. The optical flows emerge from spatiotemporal changes in a set of sequential image frames. Quantifying this optical flow field through estimations can therefore give further spatial information about the observed objects in the frames and not least the observer. The optical flow can be regarded as an estimate or truncation of the *motion field*, which in turn is a projection of 3D points in the scene space, or scene flow, onto the 2D image space (Khan and Adnan, 2017). This motion of the three-dimensional points, translated into the vessel's reference frame, provides the odometry estimation of the vessel, and the solution to the COLREG use case.

1.4 Research questions

RQ-1 How well can a vision based odometry estimator perform in marine settings for local navigation purposes with respect to positioning and orientation of the vessel?

RQ-2 What are the ideal environment settings and pitfalls for such monocular vision-based odometry deployment?

2. RELATED WORKS

Several approaches for a vision-based self-localisation of autonomous boats have been studied. The majority of these studies are focused on localisation within river settings, since it may not be possible or feasible to use *global navigation satellite systems* (GNSS) due to vegetation, and man-made structures at the shores disrupting GNSS signal strength. As all of these studies are within the definition of a littoral environment, they are a fair comparison to the methods presented here. However, it was noted that the vast majority of the research used short data collection runs. A comparison of feature-based and appearance-based visual odometry algorithms on stereo-image data has been investigated by Kriechbaumer and colleagues. Their results showed the feature-based method had considerably better performance (Kriechbaumer et al., 2015).

There have also been studies using a simultaneous localisation and mapping (SLAM) algorithm for the localisation of an autonomous boat. For example, Meier et al. developed a novel method which exploits the reflections in a river to segment water from land. By matching the symmetric features above and below the waterline, i.e. water reflection, in each stereo image, they were able to estimate the height and normal of the water surface. As a result, a robust algorithm for detecting the waterline from the height and normal was introduced. Incorporating this newly introduced feature as an input for localisation solving became the inception of Curve SLAM (Meier et al., 2021).

Further studies have been extended to harbour area settings and conducting localisation and mapping using a monocular camera. Wang et al. included a feature-based visual SLAM approach as well as an optical flow method based on DSO in their study investigating the feasibility of visual SLAM in such an environment. Since water occupied a large portion of the image, the feature-based method had difficulties finding suitable visual features, in contrast to the optical flow method (Wang et al., 2018). Finally, works conducted in 2017 (Terzakis et al.) looked particularly at monocular visual odometry for a marine vessel, and are the closest comparative work to the methods described in this paper. However, their methodology is based on the research of Kneip et al. (2011), which relies on an *inertial measurement unit* (IMU) to complement the visual sensors used in their experimentation.

3. METHOD

3.1 Data collection and pre-processing

As one of the more notable shortcomings with the related works was the lack of diverse experiment arenas, the authors of this paper looked at conducting their marine dataset collection through the littoral waters of Gothenberg, Sweden (Benderius et al., 2021). The vessel in use is a 12.6 m pilot boat, with a combined sensor suite of LIDAR, radar, IMU, GNSS, and cameras. For this particular methodology, the camera in use was a 10 bit monochrome FLIR 10GigE Oryx with an Edmund 16 mm *f/4, 1" HPr FFL* lens with global shutter. For validation, a modular Trimble R9s connected to a single antenna was used to

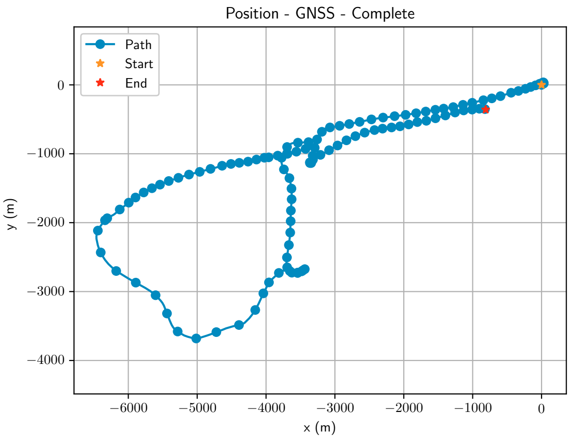


Fig. 1. The GNSS data of a completed route in the Gothenburg region is shown above. A point on the trajectory is placed every twenty seconds to give a sense of velocity during the data collection. The complete data set is further divided to smaller data sequences and presented below.

capture GNSS position. The camera was mounted on the fore-left region of the vessel and angled forward. A frame rate of approximately 60 frames per second was captured with a resolution of 1920 by 1200 pixels. The classical pinhole model is used for intrinsic calibration in order to eliminate *radial distortions*.

The GNSS antenna was mounted in the sensor mast of the vessel. The distance between the antennas and the camera was static throughout the testing period and accommodated accordingly during validation. For the purposes of this paper, the starting point of the initial run is defined as the origin in an (x, y) Cartesian plot, with the true north orienting in the positive y direction. An example of an entire run can be seen in Fig. 1, and covers a region of 8000 m by 4000 m. The data collection was conducted within the Gothenburg region in West Sweden, with an emphasis on moving through regions that were vibrant in static and dynamic objects, with varying weather and light conditions. The vessel was also deliberately inconsistent with speed and turning, as it was given a set of instructions in line with a typical use case for the vessel.

3.2 Direct sparse odometry

In short, direct sparse odometry (DSO) optimises the *photometric error* over selected keyframes. This section describes the details of the algorithm using monocular vision, including the formulation of the model for the photometric error, the windowed optimisation, and management of image frames and points (Engel et al., 2017). The whole algorithm is schematically illustrated in Fig. 2. Using the library developed by Engel et al. (2017) [1], the algorithm has been integrated into the *OpenDLV* [2] framework powered by *libcluon* [3] and a microservice architecture, where small functional entities in software jointly comprise a more complex software system to provide a service. This implementation is in line with the Reeds data set collection project (Benderius et al., 2021).

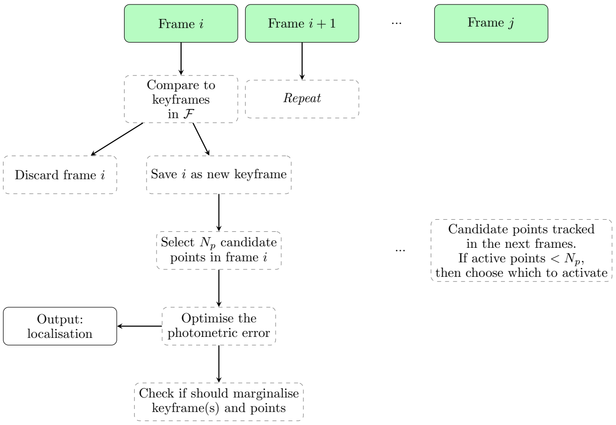

Frame i
Frame $i+1$
...
Frame j
Compare to keyframes in $\mathcal{F}$
Repeat
Discard frame i
Save i as new keyframe
Select N_p candidate points in frame i
...
Candidate points tracked in the next frames. If active points < N_p, then choose which to activate
Output: localisation
Optimise the photometric error
Check if should marginalise keyframe(s) and points

Fig. 2. A schematic illustration of the monocular direct sparse odometry algorithm showing the steps from receiving a new image frame to the odometry estimation.

First, let the transformation matrix $T \in \mathrm{SE}(3)$ denote the camera pose and define the transformation from the world-to the camera coordinate systems of the camera. In direct models, every p in the 2D image Ω is described using a single parameter of the inverse depth in the frame, while the indirect counterpart uses three unknown (Engel et al., 2017).

Next, the photometric error is explained. Let $I_i(\boldsymbol{x})$ denote the observed pixel intensity for frame i and pixel $\boldsymbol{x} \in \Omega$. Further, denote I_i as the reference image and I_j as the target image. For a point $\boldsymbol{p} \in \Omega_i$ in the reference image I_i that is also observed in the target image I_j, the photometric error is modelled as a weighted sum of squared differences, where the sum is taken over a small neighbourhood. This neighbourhood of pixel $\boldsymbol{p}$ is denoted as $\mathcal{N}_p$. Let $\boldsymbol{p}'$ denote the projection of $\boldsymbol{p}$ with inverse depth d_p as

$$\boldsymbol{p}' = \Pi_c(R\Pi_c^{-1}(p, d_p) + t) \qquad (1)$$

where R and $\boldsymbol{t}$ define the camera pose

$$\begin{bmatrix} R & \boldsymbol{t} \\ 0 & 1 \end{bmatrix} = T_j T_i^{-1}. \qquad (2)$$

Let the weight for point $\boldsymbol{p}$ be defined as

$$\omega_p = \frac{c^2}{c^2 + ||\nabla I_i(\boldsymbol{p})||_2^2}. \qquad (3)$$

Pixels with large intensity gradients are assigned small weights as a result. Furthermore, in frame i, let a_i and b_i be the brightness transfer function parameters. Finally, for point $\boldsymbol{p}$ in frame j the photometric error may be defined as

$$E_{\boldsymbol{p},j} = \sum_{\boldsymbol{p} \in \mathcal{N}_p} \omega_p \left\| (I_j(\boldsymbol{p}') - b_j) - \frac{t_j e^{a_j}}{t_i e^{a_i}} (I_i(\boldsymbol{p}) - b_i) \right\|_\gamma. \qquad (4)$$

Here, $||\cdot||_\gamma$ is the Huber norm, t_i and t_j the exposure times for frames i and j.

Extending the photometric error per pixel per frame, let $\mathcal{F}$ be the set of keyframes, $\mathcal{P}_i$ the set of tracked points in frame i, and obs$(\boldsymbol{p})$ the set of frames where the point p is visible. Then, the full photometric error is modelled as

[1] `https://github.com/JakobEngel/dso`
[2] `https://opendlv.org`
[3] `https://github.com/chrberger/libcluon`

$$E_{photo} = \sum_{i \in \mathcal{F}} \sum_{\boldsymbol{p} \in \mathcal{P}_i} \sum_{j \in \text{obs}(p)} E_{\boldsymbol{p},j} \tag{5}$$

The model defined is optimised using a *sliding window* in combination with the Gauss–Newton method. Leutenegger et al. (2015) proposed and implemented the optimisation of a model over a set of keyframes which is later adapted in to DSO. The set of variables, over which the model is optimised, contains the camera poses of all keyframes together with the brightness parameters, inverse depth values, and camera intrinsic parameters.

Furthermore, there are mechanisms for managing keyframes and points. More specifically, these decide on the sets $\mathcal{F}, \mathcal{P}_i$ and obs($\boldsymbol{p}$), defined above, including the initialisation of parameters. $\mathcal{F}$ contains up to N_f active keyframes and all $\mathcal{P}_i$ can together contain a total of N_p active points. Each new frame is compared to the newest keyframe using classical image alignment together with an image pyramid and a constant motion model. It is enough to compare to the latest keyframe since all active points are projected into it. In the next step, the frame is either saved as a new keyframe or discarded. There are three variables that are considered when determining a keyframe namely, the change in the field of view

$$f = (\frac{1}{n} \sum_{i=1}^{n} ||\boldsymbol{p} - \boldsymbol{p}'||^2)^{\frac{1}{2}}, \tag{6}$$

the translation

$$f_t = (\frac{1}{n} \sum_{i=1}^{n} ||\boldsymbol{p} - \boldsymbol{p}'_t||^2)^{\frac{1}{2}}, \tag{7}$$

and the change in camera exposure between frame i and j

$$a = |\log(t_j t_i^{-1} e^{a_j - a_i})|. \tag{8}$$

These are then put together, forming a weighted sum criterion when creating a new keyframe upon fulfilled

$$w_f f + w_{f_t} f_t + w_a a > T_{kf} \tag{9}$$

for some value-defined threshold T_{kf}. The Eq. (6) describes the mean square optical flow, which indicates changes in the field of view. In Eq. (7), $\boldsymbol{p}'_t$ is the projected point attained when R equals the identity matrix i.e. no rotation. The quantity in this equation measures the mean optical flow only considering the pure translational contribution.

A keyframe in $\mathcal{F}$ is determined to be marginalised if only a few of its points are found in the newest frame. The latest two keyframes I_1 and I_2 are always kept in consideration at all time. In addition, when the size of $\mathcal{F}$ exceeds N_f, a keyframe is also marginalised. To this end, let the Euclidean distance between two frames I_i and I_j be $d(i,j)$, and ϵ a small constant. This quantity defines a distance score I_i

$$s(I_i) = \sqrt{d(i,1)} \sum_{I_j \in \mathcal{F}, j \neq \{1,2\}, j \neq i} (d(i,j) + \epsilon)^{-1}, \tag{10}$$

where the keyframe with the highest distance score is marginalised. The corresponding active points are marginalised as well.

The last section of the algorithm governs point management. There are three types of points: candidate, active, and marginalised. For each new keyframe, N_p candidate points are sampled and selected to be spatially well-distributed and have a distinct image gradient magnitude compared to its neighbouring pixels. The procedure aims to find uniformly spread candidate points in areas with a high gradient and sparse points in regions with a lower gradient. The algorithm is based on blocks with adaptive sizes from which the pixel with the highest gradient is chosen if the value also exceeds an adaptive threshold.

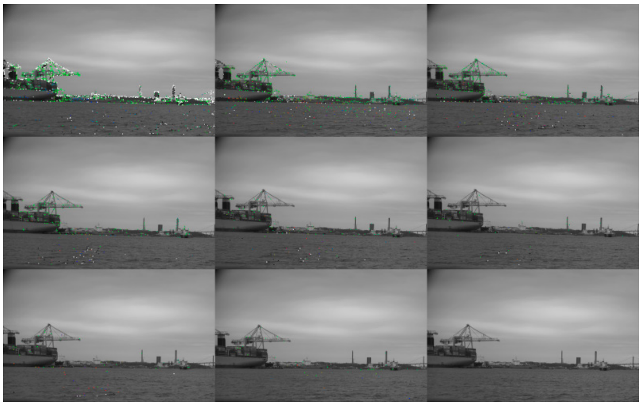


Fig. 3. A maximum of nine keyframes were used in the DSO algorithm and they are shown in the figure. The oldest keyframe is presented in the upper left corner and the most recent in the lower right corner. A large number of feature points are being tracked, managed, and labelled as *candidate, active*, and finally, *marginalised* when lost. In the figure, it can be seen that the crane provides a considerable amount of possible visual tracking features.

Through the *epipolar line*, the candidate points are tracked in the succeeding frames with the help of a discrete search by minimising the photometric error for that particular point. The initial values for the depth are found and set from this search. First, all active points and candidate points are projected onto the newest keyframe. Then, activation of candidate points with the largest distance to already active points is made. The distance threshold increases with the block size for every iteration made. Finally, active points are marginalised once the corresponding keyframe is also marginalised as described above and in Fig. 2.

3.3 Validation

Validation for the estimations from the DSO is conducted against the GNSS data in two different ways. Firstly, estimating the positioning, the comparative Cartesian x, y grid coordinates are chosen. These results of the DSO are not properly scaled and not aligned at run-time but can be made aligned with respect to e.g. GNSS data in post-processing for error analysis. One such common method is the Kabsch–Umeyama algorithm which attempts to align point patterns in order to minimise the least-squares estimation of the two sets of points (Kabsch, 1976; Umeyama, 1991). Such transformation is also known as similarity transformation (Sim(3)) which is a Lie group like SE(3) but SE(3) does not include the scaling of the dimensions. For this paper, the Kabsch–Umeyama algorithm has been applied with a chosen pivot point at the proximity of the starting point instead of the mean of the point patterns.

Secondly, whilst the units of distance are incompatible due to the lack of proper scale at run-time, the angular orientation is a definite computation and suffers considerably less from the lack of depth vision due to dealing with angles relying mainly on the intrinsic calibration of the camera. Thus it may be aligned with the help of similarity transformation as mentioned above to the GNSS data for a close ground-truth comparison for correcting the constant offset due to arbitrary orientation of the local reference system. Therefore, the comparative heading is computed through orientation and compared to the GNSS for a tangible result. The heading data for the GNSS was naïvely estimated and computed by retrieving the tangent of the path.

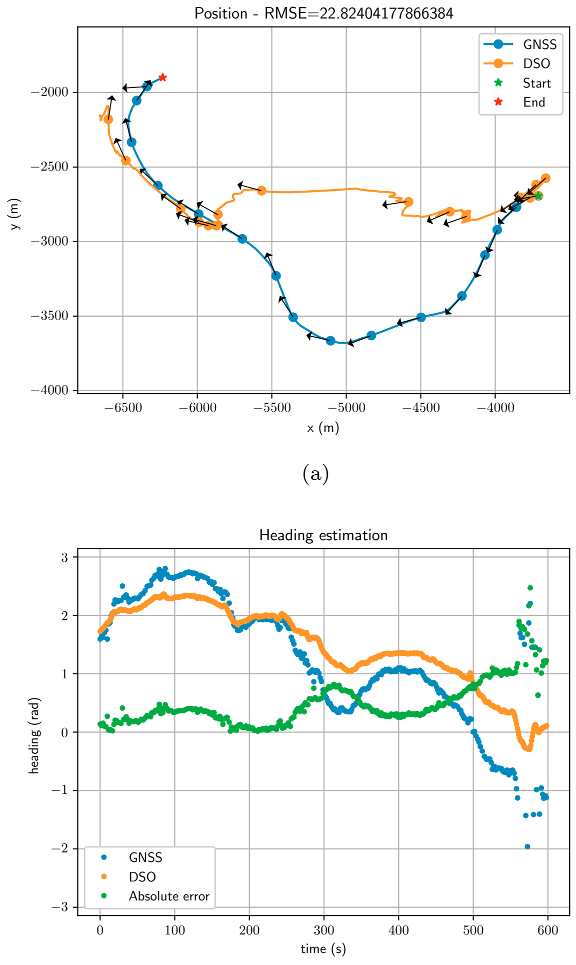
4. RESULTS

The complete route of the data collection is shown in Fig. 1 and has been divided in to six data sequences of which four is presented below. A visualisation of the visual feature tracking in the image keyframes using the DSO is shown in Fig. 3. The colours of tracked features represent different labels applied to them: candidate, active, and marginalised points. These points are essential for the DSO algorithm to determine an estimation of the odometry, which is shown as position estimates in Fig. 4a and 5a, and heading estimates in Fig. 4b and 5b respectively.

4.1 Position

There was significant variation in the absolute errors encountered over the six data sequences. In completely ideal situations, where the majority of the keypoints covered static objects, and there was little wind generating waves or cloud movements, the final odometry generated by the DSO was relatively smooth compared to the GNSS, see Fig. 6a. However, when the orientation of the vessel angled away from the shoreline, which occurred when conducting a left-hand turn around the headland, the camera lost vision of any static object. Instead, the DSO reinitialised the keypoints on the primary object in the frame, another vessel that was moving at the same speed and heading relative to the camera. Due to this, the relative motion of the camera to the registered keypoints started approaching zero, depicted in the difference seen towards the end of the plots in Fig. 7a. Each line marker is representative of twenty seconds, giving a visual indication of the estimated velocity. Markers further away from each other indicate a higher velocity. It can be seen that the position derived by the DSO shows the velocity dropping to almost zero, with the markers converging on their neighbours, which is true when compared to the relative velocity of the bulk carrier, but not when compared to the global frame.

When encountering winds, there was significant water and cloud movement. An example of this keypoint registration can be seen in Fig. 4a. Noticeably, there is considerable disconnection when compared to the GNSS position. Whilst there is some feasibility of DSO in a pristine, calm marine environment, this present build is too volatile to be used as a standalone positional suite.



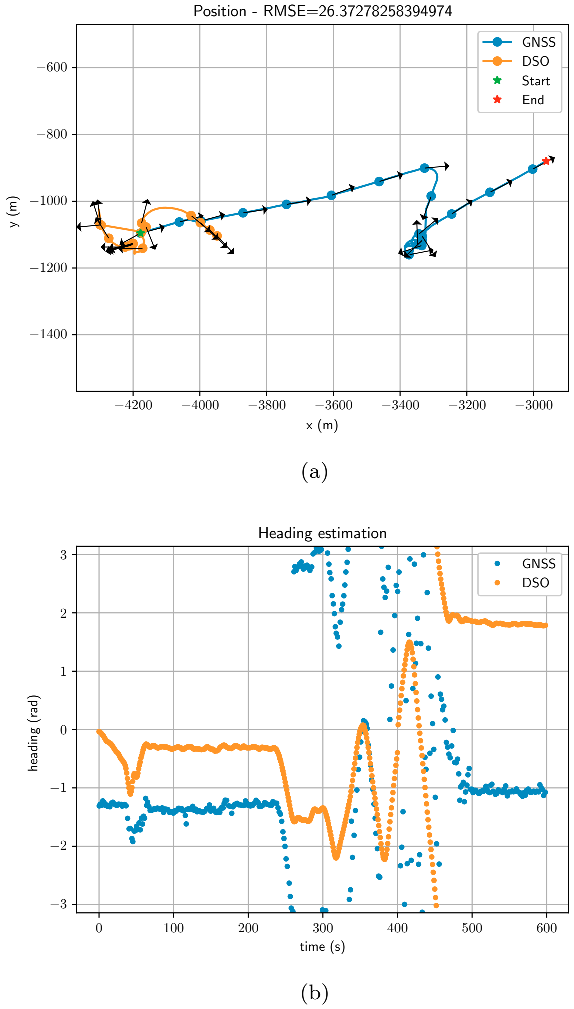
(a)

(b)

Fig. 4. Similarity aligned estimation odometry results from the works of Engel et al. (2017) using a monocular camera sensor is shown in (a) with the equivalent GNSS data trajectory during the same time period. A marker is placed every 20 s to give a sense of speed with the associated heading estimation depicted as an arrow. The abrupt jump and discrepancies in position can be partially explained where the algorithm placed keypoints to a non-static object, such as clouds or waves. In (b) the heading comparison of the GNSS and DSO is shown along with the absolute error. While the heading of GNSS is an estimation of the true north heading, the DSO counterpart is unable to bind to a global reference system at run-time without additional external information. This results in a constant offset when compared to the true north heading while the error drift is explained by the accumulative error and the problem of scaling in the DSO itself.

4.2 Heading

In contrast to the positing estimates, there was noticeable accuracy in the DSO heading when compared to the GNSS. Fig. 5b shows the unaligned oriented DSO heading alongside the GNSS heading counterpart during the same time window. This is also consistent in Figs. 4b, 6b, and 7b

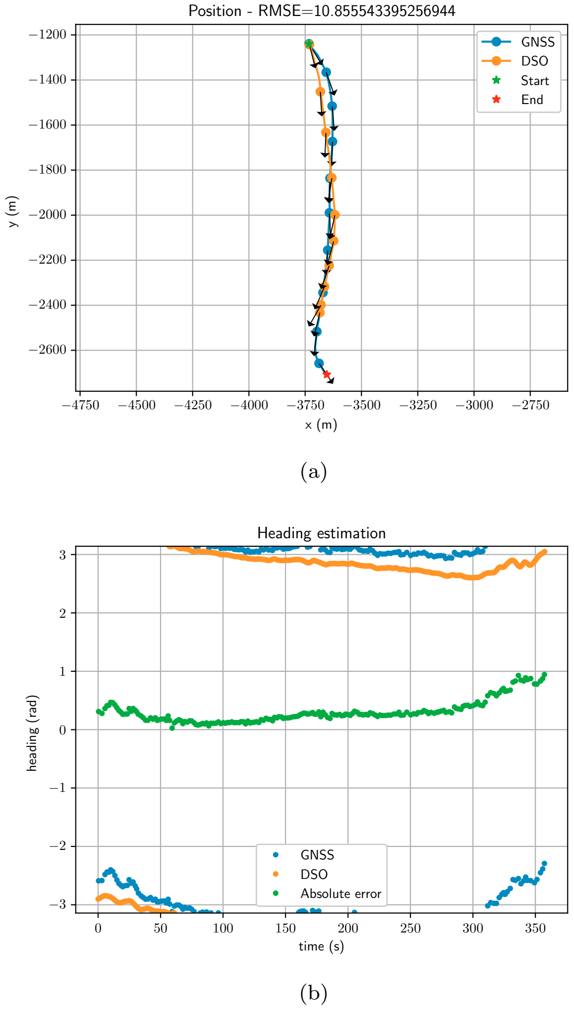


(a)

(b)

Fig. 5. In this data sequence, the DSO struggles to correctly capture the position properly due to relatively quick rotations causing the DSO to retrack its estimate back to the starting point of the data sequence. Sim(3) alignment of the odometry is not ideal for this sequence, due to the lack of similarity of the position points. While the position estimation of DSO is not performing well, the orientation estimation is largely unaffected and is still accurate during challenging conditions. Sim(3) alignment has been omitted in (b) for better visualisation and comparison.

which have significantly similar plots in both GNSS and DSO orientation when aligned. This odometry, unlike the previously mentioned position, was not impacted by wind, weather or other moving objects, and it is quite robust over the six different data sequences. In contrast to the positioning aspect of DSO, the orientation is quite feasible for vessel odometry, except when the vessel uses current and wind to change orientation whilst maintaining a stationary position. In this scenario, there are significant deviations in both position and orientation (see Figs. 5a and 5b respectively). There is also a slight deviation in Fig. 7b, at approximately $t = 230\,\mathrm{s}$, which is likely caused by a visual loss of static objects in the frame. However, the track quickly re-establishes itself and continues to follow the GNSS plot accordingly.



(a)

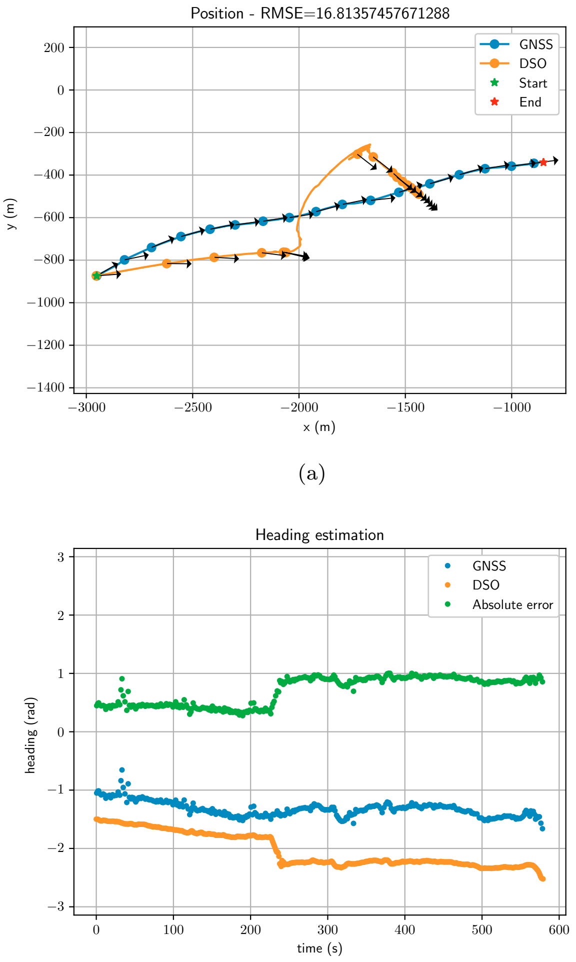
(b)

Fig. 6. A fairly straight trajectory data sequence is fed to the DSO with good performance under favourable conditions. The DSO heading estimate performs quite well in both position and heading albeit having a small accumulative drift error in the heading estimate.

5. DISCUSSION AND CONCLUSION

The primary point of discussion is the lack of automatic scaling and alignment at run-time, which is necessary to complete the relative odometry to the global frame during operation. As mentioned by Engel et al. (2017), the rate of scale (both positioning and orientation) between the reference frame of DSO and the reality is not fixed, and may actually drift over time, which prevents single frame initialisation for automatic scaling. Alternatively, active automatic scaling can be resolved through the use of a Kalman filter as the data progresses. It has further been shown that a feature-assisted approach significantly reduces this drift and may be a viable alternative (Younes et al., 2018).

Locally, it is quite feasible to manoeuvre the vessel using the unscaled and unaligned odometry output, based on the initial frame taken in the data capture. This is shown through the results, with all DSO positional plots taken with the first frame oriented along the localised x-axis. This also allows for easier observation of the known issues



(a)

(b)

Fig. 7. This data sequence is towards the end of the complete data collection of the route. It provides a feature-rich shoreline including moving objects. The moving objects in the image frame cause the DSO to lose continuity and re-track, resulting in a very abrupt change in its estimates of position and orientation. The large jump may be seen in the DSO estimate around $t = 230$ s while in reality, no rotation was actually applied to the vessel.

with using DSO as a standalone navigation system. Firstly, in Fig. 7a, it can be seen that as the vessel continues to stay oriented in the same direction, there is a brief moment where the DSO loses tracking, and attempts to reinitialise, resulting in significant positional sway. Secondly, in the same figure, with no found static keypoints, the DSO has acquired the other moving vessel and the estimated velocity drops close to zero due to this comparison in relative velocities. As the lack of static keypoints is the primary issue with velocity, there is a need for some form of complementary vision and sensor fusion to ensure that there remain some static keypoints in the frame, which is feasible given the littoral environments the pilot boat operates in.

5.1 Future work

The development of monocular visual odometry is a potential first step in developing a visual framework suitable for providing control logic in a navigation suite for marine vessels. However, there needs to be some form of sky and water segmentation to remove the current volatility, of which there is potential with a method proposed by Steccanella et al. (2020). Moreover, further segmentation of independently moving objects has to be performed in order to exclude them from image scenery in odometry estimation. The scaling and error drift challenges associated with DSO can be accounted for and mitigated with the help of a Kalman filter by fusing additional external sensor data and process modelling. With that being resolved, the next step would be to develop a complete visual coverage, in line with *Rule 5* of the COLREGs, which would be in the form of composite imagery. For this, the sensor platform will move towards a multi-camera setup, using an extrinsic calibration method, similar to that of Horn et al. (2021). From this, a thorough analysis of the merits of Composite-DSO against monocular DSO is needed.

ACKNOWLEDGEMENTS

This research is sponsored from the project *Referensdata och algoritmer till stöd för forskning och utveckling av smarta fartyg* in Sweden (Trafikverket, TRV 2019/1201031). Experimental support was given by the research laboratory Revere at Chalmers University of Technology, as part of preparatory work for the Reeds benchmarking dataset.

REFERENCES

Benderius, O., Berger, C., and Blanch, K. (2021). Are we ready for beyond-application high-volume data? the reeds robot perception benchmark dataset. *arXiv preprint arXiv:2109.08250*.

Engel, J., Koltun, V., and Cremers, D. (2017). Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3), 611–625.

Hayakawa, J. and Dariush, B. (2019). Ego-motion and surrounding vehicle state estimation using a monocular camera. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2550–2556. Paris, France. doi:10.1109/IVS.2019.8814037.

Horn, M., Wodtko, T., Buchholz, M., and Dietmayer, K. (2021). Online extrinsic calibration based on per-sensor ego-motion using dual quaternions. *IEEE Robotics and Automation Letters*, 6(2), 982–989.

Kabsch, W. (1976). A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5), 922–923.

Khan, N.H. and Adnan, A. (2017). Ego-motion estimation concepts, algorithms and challenges: an overview. *Multimedia Tools and Applications*, 76(15), 16581–16603. doi:10.1007/s11042-016-3939-4.

Kneip, L., Chli, M., and Siegwart, R. (2011). Robust real-time visual odometry with a single camera and an imu. In *Proceedings of the British Machine Vision Conference 2011*. British Machine Vision Association.

Kriechbaumer, T., Blackburn, K., Breckon, T., Hamilton, O., and Rivas Casado, M. (2015). Quantitative Evaluation of Stereo Visual Odometry for Autonomous Vessel Localisation in Inland Waterway Sensing Applications. *Sensors*, 15(12), 31869–31887. doi:10.3390/s151229892.

Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., and Furgale, P. (2015). Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3), 314–334. doi: 10.1177/0278364914554813.

Meier, K., Chung, S.J., and Hutchinson, S. (2021). River segmentation for autonomous surface vehicle localization and river boundary mapping. *Journal of Field Robotics*, 38(2), 192–211. doi:10.1002/rob.21989.

Steccanella, L., Bloisi, D.D., Castellini, A., and Farinelli, A. (2020). Waterline and obstacle detection in images from low-cost autonomous boats for environmental monitoring. *Robotics and Autonomous Systems*, 124, 103346. doi:10.1016/j.robot.2019.103346.

Terzakis, G., Polvara, R., Sharma, S., Culverhouse, P., and Sutton, R. (2017). Monocular visual odometry for an unmanned sea-surface vehicle. *arXiv preprint arXiv:1707.04444*.

Umeyama, S. (1991). Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4), 376–380. doi:10.1109/34.88573.

Wang, S., Zhang, Y., and Zhu, F. (2018). Monocular visual slam algorithm for autonomous vessel sailing in harbor area. In *2018 25th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*, 1–7. IEEE.

Younes, G., Asmar, D., and Zelek, J. (2018). Fdmo: feature assisted direct monocular odometry. *arXiv preprint arXiv:1804.05422*.

Zhou, X.Y., Huang, J.J., Wang, F.W., Wu, Z.L., and Liu, Z.J. (2020). A study of the application barriers to the use of autonomous ships posed by the good seamanship requirement of colregs. *The Journal of Navigation*, 73(3), 710–725.