PROJECT REPORT

LiDAR and Vision Based Pack Ice Field Estimation for Aided Ship Navigation

by

Mohamed W. Mehrez Said,

Harsh Environment Technology Center (HETC),

Memorial University of Newfoundland,

NL, Canada.

Date: 31st August, 2017

Project Supervisors: Oscar De Silva
George Mann
Claude Daley
John Dolny
Dan Oldford
Ray Gosine

Table of Contents

Introduction

Ships travelling through pack ice are exposed to structural damage in the hull due to collisions with ice-floe. GEM simulation environment, a Memorial University project, is an ice-ship interaction software that allows the study of the impact forces applied on a ship, when it maneuvers through pack ice [1]. At a rate much faster than the real-time, GEM is capable of simulating ship navigation through complex pack ice formations [1]. Such a tool is beneficial in predicting hazardous collisions that affect structural integrity and operational performance of ships and floating offshore structures.

In addition to performance prediction that GEM can provide, it can be also used for real operation of actual ships by generating operational planning commands. In order to use the software for this hyper-real time simulation, the near field ice information need to be accurately acquired. Upon availability of such information, GEM can also be used in a "feed forward" near-field hazard warning and avoidance system (HWAS). In the first phase of this project, a computer vision system was developed to detect and numerically reconstruct a pack ice field using the information received from a camera mounted on a ship. The developed system was tested in laboratory settings with fixed ice polygons [2].

The operation of the vision based system is highly challenged by the lighting and weather conditions, which can degrade the system performance. The 2D shape of an ice floe can be produced accurately based on a vision system. However, the reconstruction of ice-floe locations and dimensions is less reliable from a 2D image [2]. Therefore, in this project, a Light Detection and Ranging (LiDAR) sensor is integrated with a camera in order to achieve a better 3D perception of pack ice fields. LiDAR sensors provide reliable depth information, which is minimally affected by lighting conditions of the field.

Project Objective

The objective of this project is threefold. Firstly, build and develop a LiDAR-vision based instrumentation that is capable of simultaneously capturing a 3D point-cloud and a 2D image of the field. Secondly, collect a realistic data set, which mimics a ship moving in an ice-floe, using the developed sensor. Finally, develop and apply a tracking algorithm to the collected data, which detects and keeps track of ice-floe polygons in the measurement.

Technical Background

This project requires an object detection method to extract the polygonal locations of each ice floe seen in a pack ice field. In a prior project to this one, an image based object detection and 3D reconstruction using camera images were used [2]. Therein, a system was developed to detect and reconstruct ice fields using both synthetic and actual images from ships operating in pack ice conditions. Finally, the developed vision analysis module was integrated with the GEM software for performance testing.

A number of practical factors challenges the image based method developed previously: a) vision sensors are highly sensitive to lighting conditions and also the weather can severally affect their performance. For night operations, even with added illuminations, the field of view becomes highly restricted. b) vision techniques can extract the 2D shape information. However, depth information is not reliable. c) detection of ice-floe polygons from an image is not a straightforward process. Therefore, developing a sensor that treats the mentioned problems is the focus of the current project.

Laser range finders offers direct acquisition of 2D points with reliable depth information; a reasonable (large) amount of 2D points on surfaces; and independence of lighting conditions. Moreover, no laser reflections come from a water surface, and this makes it easy to detect floating object on water surfaces [3]. These features render a laser range finder a necessary component to be integrated with a vision sensor for our application. Combining a 2D laser range finder with a moving unit permits the simulation of a 3D laser range finder (LiDAR) [3]. In our experiments, we use a HUKUYO [4] laser range finder with a servo motor to ensemble a 3D laser scanner. Then, we use the available techniques to reconstruct surroundings with the help of mathematical transformations depending on the physical design, which results in a 3D point-cloud.

In this project, the developed sensor is used to collect laboratory experiments data, which mimic a moving ship in a pack ice field. The collected data is then analyzed and a tracking algorithm is developed to keep track of the sensor detections of ice polygons. The developed tracking algorithm is based on the Kalman filter [7] and the Hungarian assignment algorithm [8]. The results validated the proposed concept of using the laser scanner in detection and tracking of ice-floe.

Project Methodology

The project is implemented in three respective steps. These steps include developing a sensor; collecting data using the developed sensor; and finally applying a position tracking algorithm to the collected data. The three activities of the project are:

1- **Construction and calibration of a LiDAR-Camera sensor:** In this activity, an instrument combining a LiDAR and a camera is developed and calibrated. The calibration results in the rigid transformation between a 3D point-cloud assembled by the LiDAR and an image captured by the camera.

2- **Experimental Test bed and Data Collection:** In this activity, an experimental setup is created to mimic a ship moving in a pack ice field. Moreover, the developed sensor is used to collect real data form the developed setup, i.e. laser and image data of floating objects (plastic polygons), which ensembles pack ice. The experimental setup uses Memorial University's tow tank.

3- **Polygons Tracking:** In this activity, the detected polygons are tracked using a computer program, which is based on Kalman filtering and Hungarian assignment algorithm. The tracking algorithm combines historical measurements of the polygons and gives a unique identifier to each detected polygon.

Project Timeline

The time allocated for this project is 6 months and the below table shows the time line of the project.

Activity	March	April	May	June	July	August
Construction and calibration of a LiDAR-Camera sensor						
Experimental Test bed and Data Collection						
Polygons Tracking						
Project Documentation						

Completed Work

Activity 1: Construction and calibration of a LiDAR-Camera sensor

During this phase of the project, an instrument combining a LiDAR and a camera was developed. The constructed sensor was then calibrated to obtain the proper transformation between a 3D point-cloud assembled by the LiDAR and an image captured by the camera.

The Sensor:

The developed sensor employs a HUKUYO UST-20LX laser range finder; a Dynamixel AX servo motor; a Logitec monocular camera; and a single-board computer running a UNIX operating system. All sensor components are mounted on a 3D printed casing. The sensor assembly is shown in Figure 1.
Table 1 shows also some of the specifications of the laser range finder used; we refer to [4] for a complete specifications list.

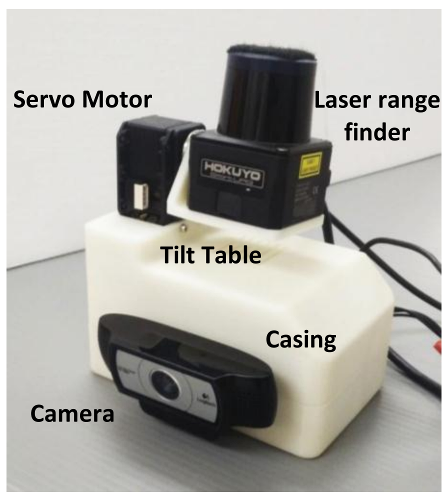

Figure 1 The developed LIDAR-Vision sensor.

Table 1 Specifications of HUKUYO UST-20LX laser range finder

Scan angle	270°	Detection range	0.06m to 20m
Angular resolution	0.25°	Accuracy	±40mm
Measurement steps	1081	Scan speed	25ms

Operation of the sensor:

The objective is to use the sensor to generate a 3D point-cloud using the LIDAR and a corresponding image using the camera. Sensor components are wired to the single-board computer, which runs a Robot Operating System (ROS) [5]. Generating an image from the camera is straightforward and is done using the `usb_cam` package implemented in ROS. However, creating a 3D point-cloud is a more involved process knowing that the used laser range finder gives readings only in its plane. Thus, tilting the laser range finder and assembling the readings for each tilt plane is necessary to generate the 3D point-cloud. As can be seen in Figure 1, the laser range finder is mounted on a tilt plane, which can be turned using a servo motor.

The tilt table is turned forward and backward at a rate of 10 rad/s while laser data is collected every 0.1 rad of angular change. In order to obtain a properly dense 3D point-cloud, the mechanism for tilting and registering laser readings is programmed to run for 8 seconds. The mentioned operational sequence is implemented on the sensor's computer using the following ROS packages: `dynamixel_tutorials`, `urg_node`, `laser_assembler`, and `point_cloud_converter`.

Operating the developed sensor results a monocular camera image and a 3D point-cloud. Figure 2 shows a sample data of the sensor. As can be seen in the created point-cloud, intensity information can be captured by the laser range finder, i.e. information on the color, at which laser reflections happen, is available by the LIDAR.

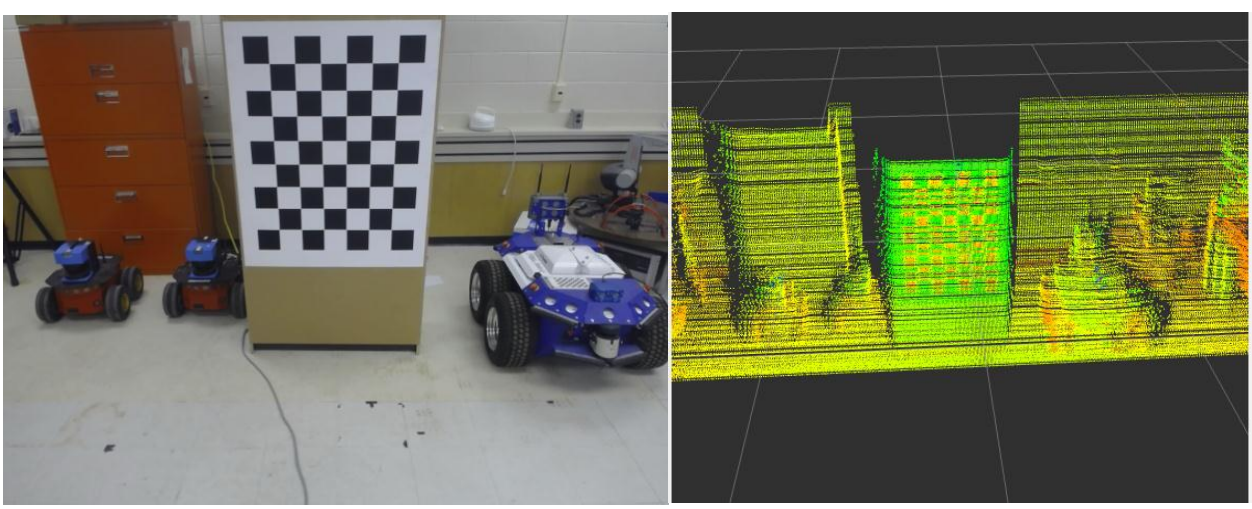

Figure 2 A sample data of the sensor outcome. Left: camera image. Right: corresponding 3D point-cloud visualized in ROS' Rviz.

Calibration of the sensor:

Calibration is basic requirement in multi-sensor platforms where data needs to represented in a common reference frame for the purpose of analysis and data fusion. On platforms where a camera provides intensity information in the form of an image and a laser supplies depth information in the form of a set of 3D points, external calibration allows reprojection of the 3D points from the laser coordinate frame to the 2D coordinate frame of the image [6].

The procedure proposed in [6] is used to calibrate our sensor. This method uses the same checkerboard calibration target commonly used for internal calibration of the camera. An interactive GUI is provided by this method, which allows the user to select a region of points in a range image which contain the planar calibration pattern. A robust fitting procedure, then, fits a plane to this selection to find estimates of the perpendicular direction and distance to the plane with respect to the coordinate frame of the laser. A separate procedure to internally calibrate the camera provides independent estimates in the coordinate frame of the camera.

The calibration procedure includes the following steps:

1- **Collection of camera and LiDAR data sets** using the sensor. These data sets employ a checkerboard target, see Figure 2 for a data set sample.
2- **Intrinsic calibration of the camera**: The camera intrinsic parameters are calibrated using the Camera Calibration Toolbox implemented in MATLAB. This procedure essentially involves supplying basic parameters like window size and number of squares in each dimension of the checkerboard grid, etc.
3- **Computing extrinsic parameters**: this procedure is implemented in MATLAB following the study [6]. It involves matching the collected images with their corresponding 3D LiDAR points. Then, checkerboard polygons are marked in the LiDAR data for each captured data set. Finally, an optimization routine is executed to estimate the rigid transformation parameters.

The above procedure results in a transformation matrix

$$T = [R \quad d],$$

where R is the rotation matrix and d is the translation vector between the LiDAR and the camera frames of reference. After estimating this rigid transformation matrix, coloring the point cloud can be done to visually verify the success of the calibration procedure. Figure 3 shows a sample result of the calibration procedure, where an image and a colored point-cloud are presented. As can be seen in the colored point cloud, some parts of the 3D points are outside the field of view of the camera, e.g. the 3D readings of the floor plane which are black colored.

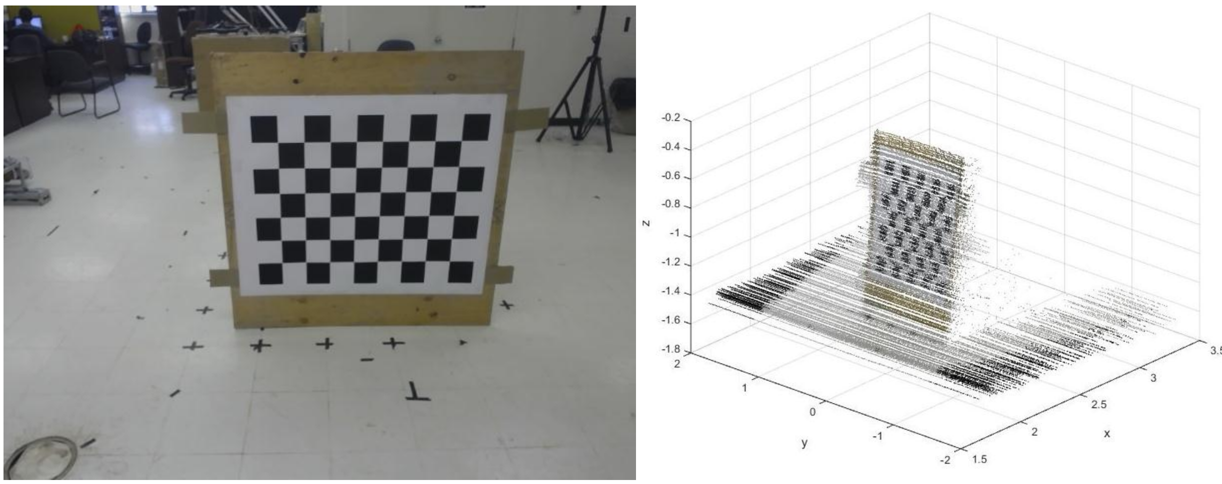

Figure 3 A sample calibration result. Left: camera image. Right: corresponding colored 3D point-cloud.

Activity 2: Experimental Test bed and Data Collection

In this phase of the project, the developed sensor was used to collect real data, which ensembles field ice-floe. The experimental setup used for this purpose mimics a ship moving in a pack ice field. Memorial University's tow tank was used at this step of the project. The tow tank has a large water tank (nearly 4 m wide and 40 m long) with a moving carriage. The sensor was mounted on the carriage while polypropylene floating polygons were used as ice-floe pieces. These polygons have nearly the same density of pack ice. The process of collecting data from the tow tank involved the following steps:

1- For a certain carriage position, run the sensor, and generate and store the image and the 3D point-cloud of the view.
2- Advance the carriage 10-20 cm.
3- Repeat the above two steps until an adequate number of data points is collected.

The above steps were performed and 53 frames of data were collected. Sample data of the first frame is visualized in Figure 4, where the point-cloud points are visualized in the planar view. A number of observations can be made from the figure:

a- No laser reflections come from the water surface and only the polypropylene objects can be clearly identified and matched with their corresponding locations in the image.
b- Both camera and LiDAR have different fields of view, e.g. not all ice-floe objects appearing in the camera appear in the point cloud and vice versa.
c- The point-cloud resolution is degraded for the far-sighted floes. This is expected as small tilt angles diverge laser beams greatly for longer distances.

Similar data to Figure 4 are collected for the remaining frames captured during the experiment.

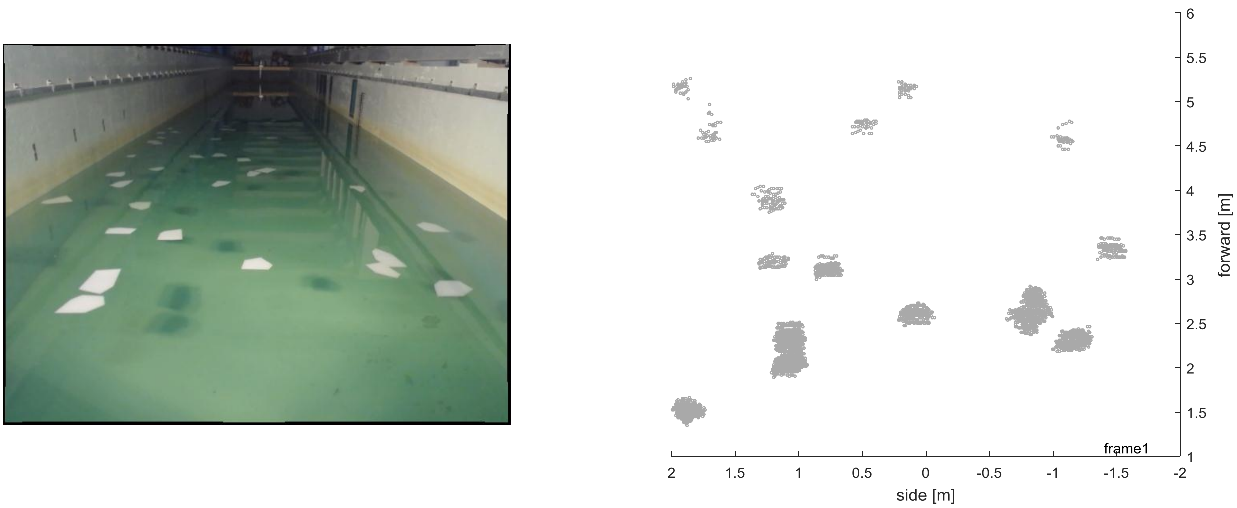

Figure 4 sample data collected from the tow tank. Left: camera image. Right: planar view of the corresponding point-cloud.

One of the advantages of using the LiDAR is that its depth readings are independent on the lighting conditions of the water surface. Moreover, another advantage of using the LiDAR here is that no laser reflections come from the water surface. This feature makes it easy to detect the ice-floe polygons, which is a rather more complicated process depending only on a camera.

Now, the LiDAR data of the polygons can be clustered and the resulting center of each cluster correspond to the center of a polygon. This is done using MATLAB. These centers can be used as the detections when a multi-object tracking algorithm is implemented; this will be discussed in more details in the following section. Figure 5 shows an example of clustering the laser data presented in Figure 4. Clusters' centers are visualized on the laser data in Figure 5. Additionally, these centers are mapped to the image using the transformation matrix resulted from the calibration process.

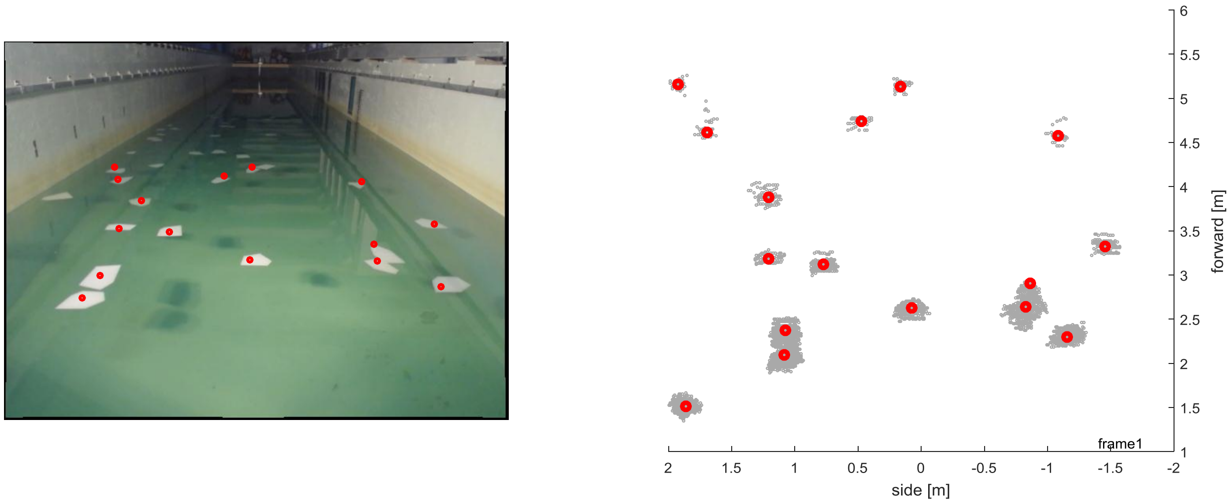

Figure 5 Result of clustering the laser data. The centers of the clusters are mapped to the image frame.

Activity 3: Polygons Tracking

In this phase of the project, the detected polygons are tracked using a computer program, which is based on Kalman filtering and Hungarian assignment algorithm. Tracking means combining historical measurements of the polygons and giving a unique identifier for each detected polygon while detections are changed from frame to frame.

The Kalman filter:

Kalman filter is an algorithm that uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables of dynamical systems. The filter employs motion and measurements models, which are linear and have the general form [7]:

$$\mathbf{x}_t = \mathbf{A}\,\mathbf{x}_{t-1} + \mathbf{B}\,\mathbf{u}_t + \epsilon_t,$$

$$\mathbf{y}_t = \mathbf{C}\,\mathbf{x}_t + \boldsymbol{\delta}_t,$$

where $\mathbf{x}_t$ is the state of the system to be tracked; $\mathbf{u}_t$ is the control input; $\mathbf{y}_t$ is the sensor measurements; and t is the current time step. $\mathbf{A}$, $\boldsymbol{B}$, and $\mathbf{C}$ are the state, input, and measurement matrices, respectively. $\boldsymbol{\epsilon}_t$, and $\boldsymbol{\delta}_t$ are additive Gaussian noise with covariance matrices $\mathbf{R}$ and $\mathbf{Q}$, respectively.

For a previous state estimate $\hat{\mathbf{x}}_{t-1}$ with covariance $\hat{\boldsymbol{\Sigma}}_{t-1}$, and current control action $\mathbf{u}_t$ and measurement $\mathbf{y}_t$, the Kalman filter estimates the current state $\hat{\mathbf{x}}_t$ over two consecutive steps a) state prediction b) measurement update. The prediction step is done by

$$\bar{\mathbf{x}}_t = \mathbf{A}\,\hat{\mathbf{x}}_{t-1} + \mathbf{B}\,\mathbf{u}_t,$$

$$\bar{\boldsymbol{\Sigma}}_t = \mathbf{A}\,\hat{\boldsymbol{\Sigma}}_{t-1}\mathbf{A}^{\boldsymbol{T}} + \mathbf{R}.$$

Next, the Kalman gain is calculated using $K = \bar{\boldsymbol{\Sigma}}_t\mathbf{C}^{\boldsymbol{T}}\left(\mathbf{C}\,\bar{\boldsymbol{\Sigma}}_t\mathbf{C}^{\boldsymbol{T}} + \mathbf{Q}\right)^{-1}$.

Finally, the corrected state estimate and its corresponding covariance are calculated via

$$\hat{\mathbf{x}}_t = \bar{\mathbf{x}}_t + K(\mathbf{y}_t - \mathbf{C}\,\bar{\mathbf{x}}_t)\,,$$

$$\hat{\boldsymbol{\Sigma}}_t = (\mathbf{I} - K\,\mathbf{C})\,\bar{\boldsymbol{\Sigma}}_t.$$

For our tracking problem, each polygon is considered as an object that moves in a 2D space, i.e. water surface, with instantaneously varying speed. Moreover, the measurement we get for each polygon is the polygon's x-y center position computed after clustering the point cloud as shown previously in Figure 5. Thus, the vectors $\mathbf{x}_t$, $\mathbf{u}_t$ and $\mathbf{y}_t$ are given by

$$\mathbf{x}_t = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix}, \mathbf{u}_t = a = 0, \text{ and } \mathbf{y}_t = \begin{bmatrix} x \\ y \end{bmatrix},$$

respectively. Moreover, the matrices $\mathbf{A}$, $\boldsymbol{B}$, and $\mathbf{C}$ are given by

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} \\ \Delta t \\ \Delta t \end{bmatrix}, \text{ and } \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

respectively. Δt is the sampling time between two captured frames and it is set to 1 second in our analysis.

Since in this tracking problem multiple objects (polygons) are tracked, the number of Kalman filters used is ideally equal to the number of tracked objects. Moreover, a major problem to be tackled in multi-object tracking is “when new detections are observed to which polygon they should be assigned?” This recalls the measurement assignment problem, which is discussed in the following section.

Hungarian assignment algorithm:

Here, we present the technique used to assign centers’ detections (measurements) to tracked polygons in our tracking algorithm. The technique is based on what’s known as the Hungarian algorithm [8]. The Hungarian method is a combinatorial optimization algorithm that solves the assignment problem. This algorithm is best explained by the following example [9].

We consider an example where four jobs (J1, J2, J3, and J4) need to be executed by four workers (W1, W2, W3, and W4), one job per worker. The matrix below shows the cost of assigning a certain worker to a certain job. The objective is to minimize the total cost of the assignment.

	J1	*J2*	*J3*	*J4*
W1	82	83	69	92
W2	77	37	49	92
W3	11	69	5	86
W4	8	9	98	23

Following the Hungarian algorithm steps, as shown in [9], leads to the following optimal assignment in the original cost matrix.

	J1	*J2*	*J3*	*J4*
W1	82	83	**69**	92
W2	77	**37**	49	92
W3	**11**	69	5	86
W4	8	9	98	**23**

The above solution means that worker 1 should perform job 3, worker 2 job 2, worker 3 job 1, and worker 4 should perform job 4. The total cost of this optimal assignment is to $69 + 37 + 11 + 23 = 140$.

Returning now to our tracking problem; when the prediction step of the Kalman filter is done, the measurement step is to start, but not before all the new centers detections are assigned as accurately as possible to the correct polygons’ predictions. At this step of the filter implementation, the Hungarian algorithm is applied. The algorithm uses a cost matrix in which the calculated cost is how far each polygon prediction from each detection. For example, assume that there are 4 polygons (P's) to be tracked and 4 corresponding detections (D's). In this case, the cost matrix will have the form

$$\begin{array}{ccccc} & D1 & D2 & D3 & D4 \\ P1 & P1-D1 & P1-D2 & P1-D3 & P1-D4 \\ P2 & P2-D1 & P2-D2 & P2-D3 & P2-D4 \\ P3 & P3-D1 & P3-D2 & P3-D3 & P3-D4 \\ P4 & P4-D1 & P4-D2 & P4-D3 & P4-D4 \end{array}$$

where $P_i - D_j$ is the Euclidean distance between the prediction of polygon P_i and the detection D_j for all $i, j \in \{1,2,3,4\}$. Using the Hungarian algorithm, the above cost matrix is optimized and each detection is assigned to the corresponding polygon prediction.

In our implementation of the tracking algorithm some rules are used in order to improve the performance of the tracking algorithm. These rules are:

1- After assignment of detections, if a detection is very far from a prediction, this detection is considered as a new polygon entering the field of view of the sensor. A new Kalman filter is triggered for such a detection. The margin for detection rejection used is $D_{margin} = 40$ cm.
2- If one of the tracked polygons is not getting any detection assignment for more than $N_{nd} = 3$ time steps, its Kalman filter is stopped.

D_{margin} and N_{nd} can be tuned to other values than the mentioned one depending on the tracking problem.

Tracking results:

The overall tracking algorithm implemented is summarized by the following steps:

1- **Detection**: measure the polygons' locations using the sensor.
2- **Kalman filter prediction**: apply the Kalman filter prediction equations.
3- **Detections assignments**: apply the Hungarian algorithm to assign measurements to polygons' predictions.
4- **Kalman filter update**: apply the Kalman filter measurement update equations.

Now, we present the tracking results. Figure 6 shows the tracking result between the first two frames collected from the tow tank. As can be seen, the tracking algorithm performed very well in estimating the tracks of motion of the polygons. Moreover, the convergence of these tracks to the actual tracks occurred immediately after obtaining frame 2 detections.

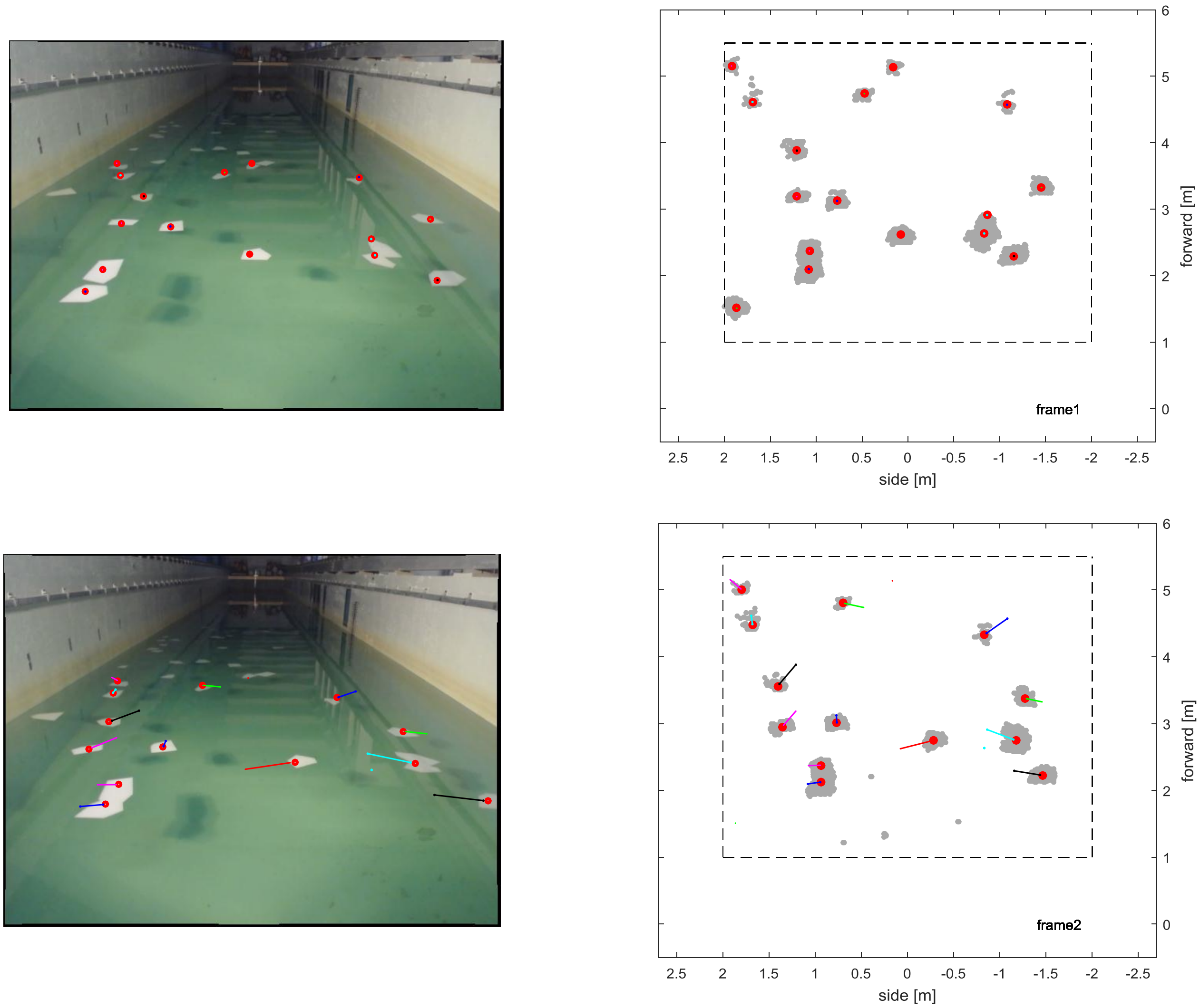

Figure 6 Tracking between the first two frames of the tank data.

One of the advantage of the used tracking algorithm is that after polygons go outside the field of view of the sensor, we still can estimate their future locations for a number of time steps (N_{nd}). This is best explained by Figure 7, where the tracking between 3 frames is presented. In the figure, two tracks are specified by two solid arrows. As can be concluded from the figure, the used tracking algorithm was able to estimate the locations of the two corresponding polygons even when they are beyond the sensor field of view.

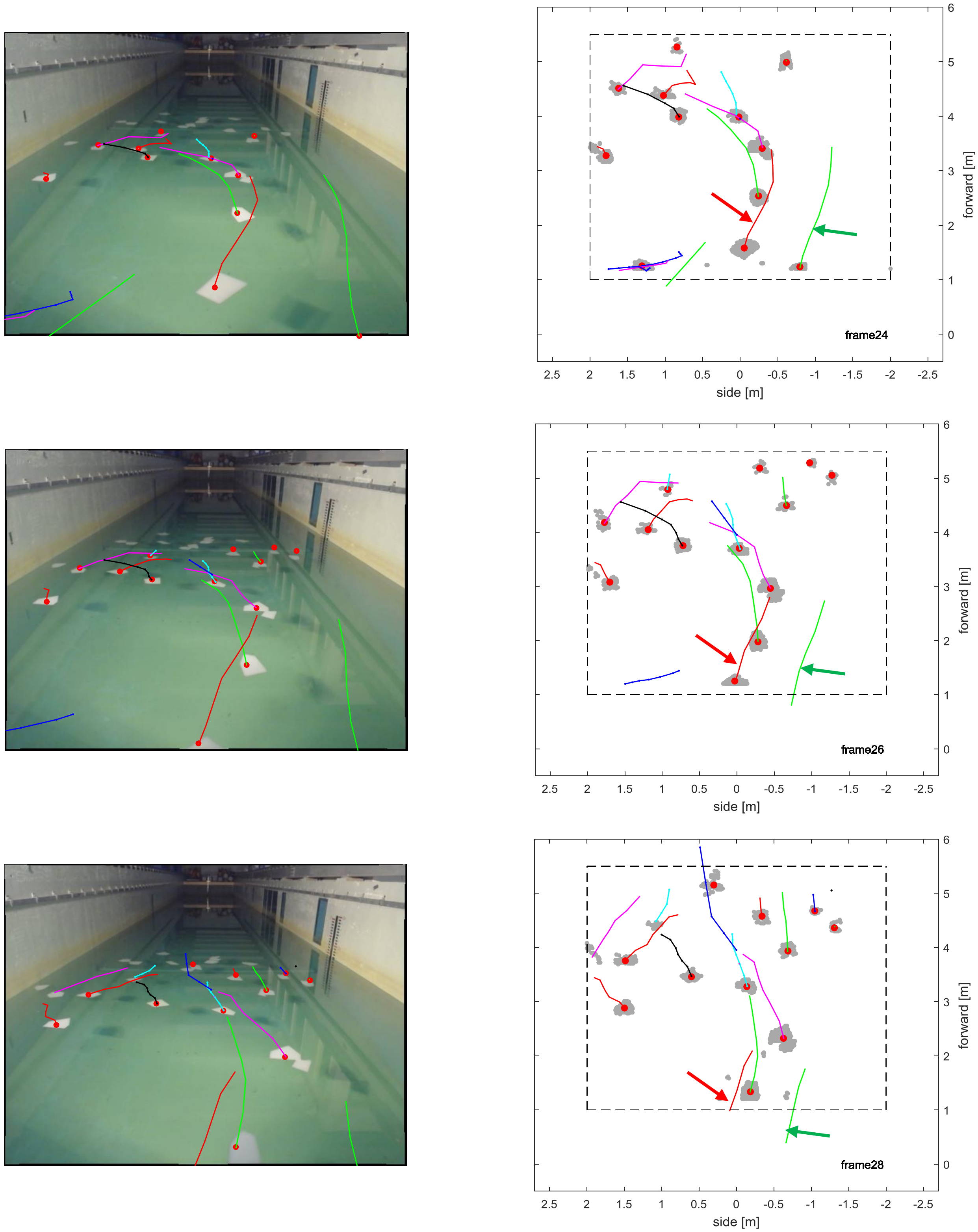

Figure 7 Tracking between frames 24, 26 and 28.

The overall tracking of all frames is illustrated in a video that is attached to the report.

Conclusions and Outlook

In this project, a LiDAR-Camera sensor is developed and calibrated to detect and track ice-floe. The developed device is necessary to implement hazard warning and avoidance system (HWAS) for ships travelling in pack ice. In contrast to the ice floe detection method (vision based) developed in the first phase of the project [2], a LiDAR sensor is used to detect such objects. Employing such a technique has several advantages over the vision detection method which are:

- Laser readings of the LiDAR sensor are less sensitive to environment (light) conditions than cameras.
- Laser range finders directly provides reliable depth information.
- When used to detect floating objects in water, no laser reflections come from the water surface, and, thus, floating objects can be easily detected.

The developed sensor was then used to collect experimental data that mimic ice-floe in front of a moving ship. The experiment was conducted in Memorial University's tow tank. The collected data set is more realistic than the one collected in the previous phase of the project [2], in which the collected data were for objects that were placed on the floor and not a water surface.

Finally, a tracking algorithm, which is based on Kalman filter and Hungarian algorithm, was implemented to keep track of each ice-floe. The results showed a good convergence and overall performance of the tracking algorithm. The developed LiDAR processing method and the tracking method completes the essential components for successful demonstration of a LiDAR/Vision based pack ice HWAS. However, the full software link from LiDAR/ image data to GEM was not completed during the project due to time constraints. This sub-component of the project will be completed and demonstrated as future work.

The overall performance of the sensor showed a noticeable enhancement to the vision only sensor developed in [2]. The conducted work provided also a proof of concept of using the LiDAR technology in ice-floe detection. However, the data collection was achieved at a low sampling rate because of the time needed to assemble 3D point-clouds. This rate can be improved by using an industrial LiDAR [10], which directly gives 3D point-clouds at rates up to 10 Hz.

Future work:

The developed work in this project can be further improved by the following:

- Adding polygons' areas as a state in the tracking algorithm and extracting this data from the image following the method (labelled watershed) shown in the first phase of the project [2]. This allows to exploit the high-resolution information that is available from the camera in the overall method.
- Using an industrial LiDAR sensor, which has a higher resolution, range, sampling rate than the developed sensor.
- Performing tracking analysis on actual field data of pack ice and demonstrating a hazard warning avoiding mechanism.

References

[1] Alawneh, S., Dragt, R., Peters, D., Member, S., Daley, C., & Bruneau, S. Hyper-Real-Time Ice Simulation and Modeling Using GPGPU. *IEEE Transactions On Computing*, 64(12), 3475–3487, 2015.

[2] Oscar De Silva, Vision Analysis of Pack Ice for Potential Use in a Hazard Warning and Avoidance System, *Technical Report, Harsh Environment Technology Center (HETC), 2016.*

[3] D. Klimentjew, M. Arli and J. Zhang, "3D scene reconstruction based on a moving 2D laser range finder for service-robots," *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1129-1134, 2009.

[4] HUKUYO Automatic CO., LTD., Scanning Laser Range Finder Smart-URG mini UST-20LX (UUST004) Specification.

[5] Robot Operating System (ROS): http://www.ros.org/

[6] Q. Zhang and R. Pless. Extrinsic calibration of a camera and laser range finder. *In IEEE Intl. Conference on Intelligent Robots and Systems (IROS)*, volume 3, pp. 2301– 2306, 2004.

[7] Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). *The MIT Press*, 2005

[8] H. W. Kuhn. The Hungarian method for the assignment problem. In *Naval Research Logistics*, *2(1-2)*, pp. 83–97, 1955.

[9] Hungarian Algorithm Example. Retrieved August 27, 2017, from http://www.hungarianalgorithm.com/examplehungarianalgorithm.php

[10] ILRIS-LR Terrestrial Laser Scanner. Retrieved August 27, 2017, from http://www.teledyneoptech.com/index.php/product/optech-ilris/