EDeformNet: Estimating Fishing Net Deformations from Sparse Observations

Isira Wijegunawardana, Jaime Valls Miro, Iñaki Quincoces, Liang Zhao, and Shoudong Huang

***Abstract*— This paper introduces EDeformNet, a novel method for real-time 3D reconstruction of fishing nets using sparse positional measurements. Currently, net deployment during large-scale fishing operations is challenging as the submerged lattice deformations that occur in response to the various environmental factors are not visible to the vessel operator. EDeformNet extends Embedded Deformation Graphs (EDGs), a commonly used technique in template-based non-rigid 3D reconstruction that allows control of embedded spaces through sparse control point correspondences. These can be suitably derived from acoustic tracking beacons attached to the net. EDeformNet enhances the standard EDG optimization scheme by including constraints that preserve surface normals at control points and guard distances between vertices in the template mesh. These improvements are proven to enable an accurate representation of the complex deformations and movements typical in purse seine nets, the fishing technique where the algorithm has been tested, which standard EDG is unable to attain. Moreover, EDeformNet also proposes a tailored strategy that dynamically adjusts the net template according to the known length of the deployed portion of the fishing net. This approach reconstructs exclusively the submerged portion of the fishing net, avoiding extraneous data from above-water sections and enhancing accuracy under realistic fishing conditions. The proposed method is validated using realistic 3D physics simulations in Blender, where quantifiable comparisons demonstrate that EDeformNet effectively captures the spatial dynamics of purse-seining. Compared to standard EDG, EDeformNet achieves superior performance, resulting in at least a 25% improvement across the array of challenging temporal scenarios studied.**

I. Introduction

Purse seining is commonly used in commercial fishing to catch large schools of pelagic fish swimming close to the surface. These fishing nets are vertically deployed in the water, in a cylindrical shape, surrounding a fishing school (see Fig. 1(a)). The net is then drawn together at the bottom, resembling a drawstring purse when closed [1], [2]. This technique has become popular in the harvesting of midwater schooling species like tuna and mackerel due to its efficiency in the capture of large quantities of fish in a single operation. It contributes significantly to global seafood production while playing an important role in satisfying global protein demands and providing economic benefits [3]. Being a widely used technique, the purse seine industry is constantly looking at the adoption of new technologies that may overcome some of the limitations latent in the use of traditional techniques, such as limited control, risking net damage, entanglement or fish escaping; inefficient decision-making, relying on manual adjustments; and reduced selectivity, causing unwanted bycatch [2]. Improving net deployment is one such area where introducing more advanced techniques can have a significant impact [4].

The adoption of robotics and AI techniques by the fishing sector is apparent given their potential to improve its sustainability credentials [5]. The literature provides frequent evidence of object detection being used to locate schools of fish and reduce bycatch [6]. Other works have focused on modelling aspects, showing how reconstructing the shape of the net during its setting was significant to analyze sinking performance [7], in turn leading to improvements in net design. These methods rely on spring-mass-damper systems that are subjected to unstable physical forces to model the behavior of the purse seine [8]. However, physics-based methods are highly dependent on critical parameters such as material properties, whilst also often requiring accurate measurement of volatile ocean currents. These prevailing uncertainties make modeling of purse seine and vessel dynamics using physics-based methods a challenge, deriving results under the assumption of idealistic conditions of limited accuracy and effectiveness.

With regards to 3D net reconstruction, the existing literature is rather limited. There are several reasons why technology has not yet enabled efficient real-time reconstructions of these open meshes. Predominantly, they are deployed underwater, over a large area (often kilometers) and in isolated locations, making traditional camera-based visualization of a net infeasible. Additionally, deformations in the fishing net caused by ocean currents yield non-rigid transformations on the net which are highly complex to solve. Existing non-rigid reconstruction methods often rely on camera-based solutions [9], [10] which, as mentioned above, are not suitable for the fishing net problem. Moreover, for it to be of practical use, the reconstruction of the purse seine is expected to be done while the net is being progressively released from the boat into the sea. Therefore, any mapping blueprint must be sensitive to the increments in the overall mesh density of the deforming object over time.

In order to overcome some of these challenges, this article proposes using sparsely attached position tracking systems to

Isira Wijegunawardana*, Jaime Valls Miro and Shoudong Huang are with the Robotics Institute, University of Technology Sydney, NSW, Australia {isiradamsith.d.wijegunawardana@student.uts.edu.au}, {jaime.vallsmiro|shoudong.huang@uts.edu.au}.

Jaime Valls Miro and Iñaki Quincoces are with AZTI Foundation, Bizkaia, Spain, {jvalls|iquincoces@azti.es}, and this work is contribution nº 1276 from AZTI, Marine Research, Basque Research and Technology Alliance (BRTA). Jaime Valls Miro is also with IKERBASQUE, Basque Foundation for Science, Bilbao, Spain.

Liang Zhao is with the Institute of Perception, Action, and Behaviour, School of Informatics, The University of Edinburgh, Scotland {liang.zhao@ed.ac.uk}.

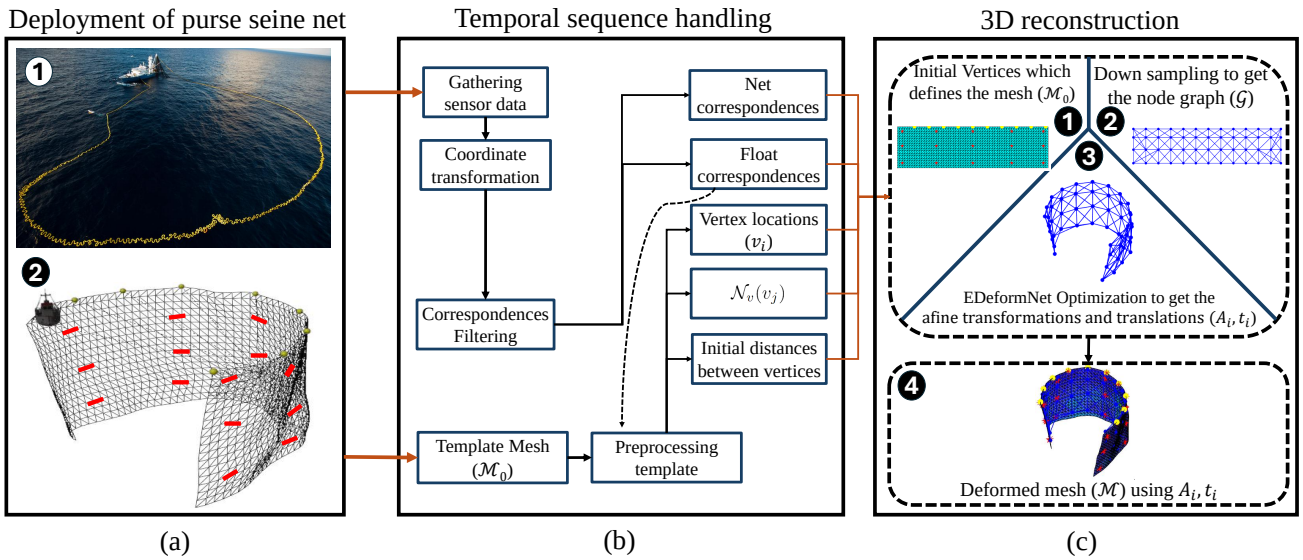

Fig. 1. (a1) Typical purse seiner setting the net whilst navigating on a circling path (yellow surface objects are the floats attached to the top layer of the net). (a2) Blender submerged simulated fishing net suspended from the top layer floats (control points are marked in red on the net). (b) The proposed EDeformNet strategy to handle the dynamic net sizes present during the net deployment scenario. (c) 3D reconstruction methodology: c1) generate a template mesh, c2) down-sample the mesh into a node graph, c3) optimize the deformation subject to the subset of control point measurements, and c4) apply the transformations to create the full scale 3D reconstruction.

the fishing net and vessel to predict the evolving deformation behavior of the net (see Fig. 1(a2)). The proposed scheme thus embarks on the challenge of deforming a meshed object with the aid of sparse position measurements. This paper adopts the Embedded Deformation Graphs (EDG) method introduced in [11] to address this problem. EDG offers computational efficiency by using a low-resolution template of the net (see Fig. 1(c)) and flexibility through least squares optimization, allowing customization with additional constraints. This enables non-rigid registration [12], aligning the undeformed and deformed net shapes using sparse control point correspondences, making it ideal for real-time applications. However, standard EDG does not preserve geodesic distance, a critical requirement for fishing net reconstruction, resulting in undesirable outcomes when correspondences become sparse. The proposed method addresses this limitation.

The EDeformNet method proposed in this article offers the following contributions.

- It is the first known attempt to use non-rigid 3D reconstruction techniques for real-time visualization of fishing nets.
- By adding two constraint terms to the traditional EDG objective function, EDeformNet achieves more accurate and realistic deformations compared to EDG. The first constraint – distance consistency – controls stretching or compression between neighboring vertices. The second – surface normals consistency – helps preserving the shape of the reconstruction in under-constrained areas.
- Leveraging the iterative nature of EDeformNet and using a dynamic template mesh, the purse seine can be reconstructed during the unfolding release.

II. Related Works

Non-rigid 3D reconstruction involves graphically constructing objects such as human bodies, animals, or soft materials like fabric which can deform in complex ways over time. This technique has currently become popular in various applications such as computer animation [13], medical imaging [14], and augmented reality [15], where precise modeling of deformable objects is essential. In recent literature, many techniques can be found which are dedicated to non-rigid 3D reconstruction. These can be clustered into Non-Rigid Structure from Motion (NRSfM) based template-free methods [16], and template-based methods [17].

NRSfM aims to reconstruct the 3D structure of deforming objects from 2D image sequences but is unsuitable for estimating the shape of fishing nets due to the impracticality of capturing full underwater scenes in field conditions, compounded by turbidity and occlusions. Additionally, NRSfM is computationally expensive and sensitive to noise and missing data. In contrast, template-based methods, which use a known initial shape and maintain topological consistency, have gained popularity [17]. These methods deform the template to align with the target shape through optimization techniques based on photogrammetry [18] or point-cloud data [19]. This alignment is defined as non-rigid registration in [12].

Deforming a template to match a target shape that evolves over time is a complex challenge, typically addressed with non-rigid registration techniques. These techniques include different parameterization schemes and constraints that can make the solution more definite [12]. The commonly used parameterization schemes to represent the deformation field include point-wise transformations [20], graph-based methods [11], skeleton priors [21], grid-based methods [22], and spline functions [23]. Among them, graph-based methods have competitive capabilities in dimensionality reduction while accurately preserving local details. EDG [11] is one such graph-based method specifically supporting reconstruction with sparse correspondence. In regard to the constraints

used in these registration techniques, the most common constraints can be classified as alignment and regularization terms [12]. In this domain, alignment terms reduce the distance between known correspondences. These terms can also perform the aligning of the normal vectors of the correspondences. Regularization terms, on the other hand, attempt to preserve the uniformity of the surface, safeguarding details and reducing distortions.

EDG-based methods have been widely applied in a variety of domains including animation rigging in computer graphics [24], human performance capture [25], 3D bone reconstruction from X-ray images [26], 3D aortic reconstruction during fluoroscopy-guided interventions [27], and deformation recovery during the skirting process in wool handling [28]. Furthermore, recent studies have extended the core concept of EDG to learning-based 3D reconstruction methods, moving beyond traditional optimization techniques [29], [30]. In addition to this, alternative methods such as As-Rigid-As-Possible (ARAP) [31] method maintain local rigidity during deformation using energy minimization, while Thin Plate Splines (TPS) [32] method offer strong interpolation in non-rigid transformations using a combination of affine transformations and a radial basis function. These methods, however, lack the customizability and modularity supported by the traditional optimization-based EDG.

III. Standard EDG Method

EDG typically represents deformations using a deformation graph. The EDG method suggests three steps for the non-rigid registration problem: downsampling, optimization, and transformation (see Fig. 1(c)). In downsampling, the N number of points in the template mesh ($\mathcal{M}_0$), each named a vertex and denoted by v_i, $i \in \{1, ..., N\}$, will be reduced to the n number of points in the deformation graph. These points are named nodes, each denoted by g_j, $j \in \{1, ..., n\}$. The nodes and vertices of the deformation graph have been depicted in Fig. 1(c). EDG at a particular time ($\mathcal{G}$) is parameterized as a set of initial node location vectors ($g_j \in \mathbb{R}^3$, $j \in \{1, ..., n\}$), translation vectors ($t_j \in \mathbb{R}^3$, $j \in \{1, ..., n\}$), and affine transformation matrices ($A_j \in \mathbb{R}^{3\times3}$, $j \in \{1, ..., n\}$). This graph can be used to retrieve the deformed full scale mesh ($\mathcal{M}$) at any time using the parameters that define $\mathcal{G}$ (i.e. $\mathcal{G} = f(g_j, A_j, t_j)$). These parameters are used to define the deformation of each node itself and the surrounding space of the node, hence the name "embedded deformation". The transformation of retrieving vertices in the mesh (v_i) using nodes in the graphs (g_j) can be defined as:

$$\hat{v}_i = \sum_{j \in \mathcal{N}_g(v_i)} \omega_j(v_i)[A_j(v_i - g_j) + g_j + t_j]. \quad (1)$$

Here, vertex v_i is assumed to be influenced by a m number of nodes in the neighbourhood $\mathcal{N}_g(v_i)$. These neighbors contribute to the deformation of v_i, shifting it to a target position denoted by $\hat{v}_i$. In the context of mesh-based geometry, shaping deformation is essentially the transformation of vertices within the deformation graph. The term $\omega_j(v_i)$ represents a weight that reflects the influence of a given node g_j to the vertex v_i. This term can be defined as:

$$\omega_j(v_i) = \frac{(1 - (v_i - g_j)/d_{max})^2}{\sum_{k \in \mathcal{N}_g(v_i)} (1 - (v_i - g_k)/d_{max})^2} \quad (2)$$

where d_{max} is the distance from v_i to the $(m+1)^{th}$ nearest neighboring node in the deformation graph.

During a deformation of the object, the defined transformation parameters (A_j, t_j) are solved as an optimization problem to obtain the deformed $\mathcal{G}$. In the EDG method, this optimization problem has an objective function with three components, denoted E_{rot}, E_{reg}, and E_{con}. Each represents a particular geometrical constraint that can be imposed when the nodes in the deformation graph are moving. Therefore, each term is expected to be minimized. The three constraints terms are defined in (3-5), respectively.

$$E_{rot} = \sum_{j=1}^{m} ||A_j^T A_j - I||_2^2 \quad (3)$$

Here, E_{rot} in (3) preserves detailing by specifying that affine transformations A_j strictly be rotations satisfying the conditions of an orthogonal rotation matrix SO(3). Thus, the local features will deform as rigidly as possible.

$$E_{reg} = \sum_{j=1}^{m} \sum_{k \in \mathcal{N}_g(g_j)} \alpha_{jk} ||A_j(g_k - g_j) + g_j + t_j - (g_k + t_k)||_2^2 \quad (4)$$

E_{reg} in (4) is considered a regularization of deformations that force the affine transformations between adjacent nodes to agree with one another. This maintains the smoothness of the surface after deformation by avoiding the deformed shape from being fragmented and distorted. Here, α_{jk} is the weight reflecting the motion effect of the neighbouring nodes to each other. $\mathcal{N}_g(g_j)$ is the set of neighbors connected to node j.

$$E_{con} = \sum_{i=1}^{n_c} ||\hat{v}_i - v_i^C||_2^2 \quad (5)$$

E_{con} in (5) is suggested to minimize the distance between v_i^C and $\hat{v}_i$, where v_i^C is the measured/actual deformed location of the sparse control point v_i, and $\hat{v}_i$ is its transformed position computed via (1). The number of control points is given by n_c.

Finally, the EDG objective function that combines the constraint terms defined above is presented in (6) as an optimization problem.

$$\min_{A_1, t_1, ..., A_n, t_n} w_{rot}E_{rot} + w_{reg}E_{reg} + w_{con}E_{con} \quad (6)$$

Here, w_{rot}, w_{reg} and w_{con} represent the weights of each component, where in EDG they were select as 1, 10, and 10^2, respectively. This function has to be minimized to obtain the optimal affine transformations and translations of the deformed mesh nodes. This can be solved as a least squares optimization problem. Once the optimization converges, $\mathcal{G}$ can be used to compute the high-resolution vertices in $\mathcal{M}$.

IV. Proposed EDeformNet Modeling Strategy

EDeformNet uses a specialized approach that dynamically adapts the net template based on the known length of the submerged portion of the purse seine net. This ensures that the 3D reconstruction reflects the underwater section of the net, eliminating unnecessary and unreliable data. This enhances reconstruction accuracy in realistic fishing scenarios. The size of the submerged portion is measured in real-time, based on the length of the released float line (see Fig. 1(b)). Once the template is configured, a suitable EDG can be adopted to perform the non-rigid registration from the template to the deformed state. However, for the specific case of purse seine modeling, the proposed scheme suggests a modified EDG objective function whereby two additional constraint terms can be added to improve the results. These include a distance consistency constraint – to maintain vertex distances, and a surface normal constraint – to preserve deformations by preserving control point normals. In field studies, the orientation of the sensor in the fishing net provides the normal direction.

A. Distance consistency constraint

Inherently, the EDG method allows compression and tension to fit the model to the ground truth via control points. However, with the sparse control point correspondence, this behavior loses its true details. The distance constraint introduced in this paper considers the vertex distances between the initial template and the deformed model. This will force the optimization algorithm to converge to a solution where the template has minimum stretch or compression during registration. Equation (7) introduces this constraint as:

$$E_{dist} = \sum_{j=1}^{N} \sum_{k \in \mathcal{N}_v(v_j)} (d_{jk})^2 \tag{7}$$

where

$$d_{jk} = \frac{||(\hat{v}_j) - (\hat{v}_k)||_2 - ||v_j - v_k||_2}{||v_j - v_k||_2}. \tag{8}$$

Here, E_{dist} will consider the change in distance between each two vertices adjacently connected in the triangulated mesh. The neighborhood of these adjacent vertices for a particular vertex j is defined as $\mathcal{N}_v(v_j)$. This can vary between 2-6 depending on the location of the vertex. The change in distance between vertex j and k is defined as d_{jk} in (8).

B. Surface normal constraint

While adjusting the model to match the control points and maintaining distance consistency, some areas of the model may become under-constrained, especially where control points have less influence. To handle this, we introduce a surface normal constraint based on the direction of the surface normals at the control points. This makes the problem better defined (well-posed) and ensures that the model deforms correctly. This constraint is represented by E_{surfn}, and is defined as:.

$$E_{surfn} = \sum_{i=1}^{n_c} ||\hat{n}_i - n_i^C||_2^2 \tag{9}$$

where n_i^C is the surface normal of the i^{th} control point.

The corresponding normal $\hat{n}_i$ of the transformed vertex $\hat{v}_i$ is computed as:

$$\hat{n}_i = \frac{\sum_{\hat{n}_{ik} \in \mathcal{N}_v(\hat{v}_i)} \hat{n}_{ik}}{||\sum_{\hat{n}_{ik} \in \mathcal{N}_v(\hat{v}_i)} \hat{n}_{ik}||_2}. \tag{10}$$

Here, $\hat{n}_{ik}$ represents the normal vector for a particular facet k around the i^{th} control point. In the mesh, the number of facets surrounding the control point can be changed based on the location of the control point. $\hat{n}_{ik}$ was computed using the cross product of the respective edge vectors of the facet formed by the three vertices representing that facet. These three vertices are defined as $j, k, l \in N_v(\hat{v}_i)$. This computation is defined as:

$$\hat{n}_{ik} = \frac{(\hat{v}_j - \hat{v}_k) \times (\hat{v}_l - \hat{v}_k)}{||(\hat{v}_j - \hat{v}_k).(\hat{v}_l - \hat{v}_k)||_2} \text{ for } j, k, l \in \mathcal{N}_v(\hat{v}_i)). \tag{11}$$

Note that the unit vector of the normal was drawn in (10) and (11) via normalization.

C. Modified EDG optimization

After integrating the two additional constraints defined in Section IV-A and Section IV-B into EDG, the objective function for the proposed EDeformNet optimization can be defined as:

$$\begin{aligned} \min_{A_1, t_1, \ldots, A_n, t_n} \quad & w_{rot}E_{rot} + w_{reg}E_{reg} + w_{con}E_{con} \\ & + w_{dist}E_{dist} + w_{surfn}E_{surfn}. \end{aligned} \tag{12}$$

Here, w_{dist} and w_{surfn} represent the respective weights for these two constraints. In our experiments, they were set as 10^2 and 10^4, respectively, by testing a range of values. The least-squares optimization problem (12) is solved using the Levenberg–Marquardt (LM) algorithm.

V. Performance analysis

The proposed scheme was tested against static and dynamic fishing net deployment sequences with comprehensive ablation studies and compared with alternative deformation mapping schemes. The impact of the number of control points is also presented in this section.

A. Ground truth generation

Collecting real geometries to use in the underlying application was quite challenging, as it is currently not feasible to derive the exact shape of a large submerged fishing net in a field experiment. Therefore, we have resorted to employing dynamic simulations created with the 3D computer graphics

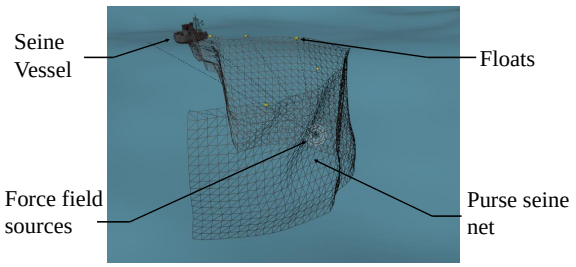

Fig. 2. Illustration of the Blender simulation setup (net size $100m \times 25m$)

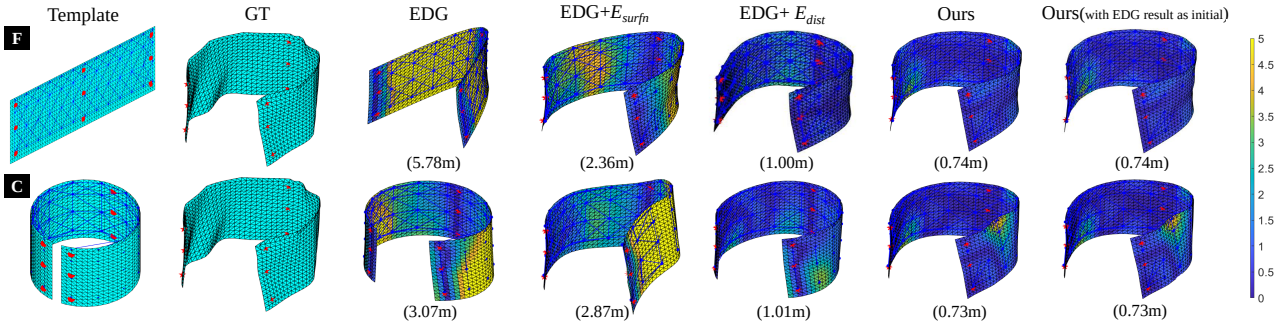

Fig. 3. Comparative 3D reconstruction results for the five methods. (1) EDG. (2) EDG+E_{surfn}. (3) EDG+E_{dist}. (4) EDeformNet. (5) Using the result attained from standard EDG as initial A_i, t_i for EDeformNet. RMSE is shown in brackets under each figure. (F - flat template, C - cylindrical template, GT - ground truth).

and animation software Blender [33] to use as the ground truth. The fishing net model was created using the cloth physics simulator, with deformation behaviors, ocean physics and force currents incorporated into realistic simulations in which the net unfolds while being released in a circular path, whilst controlled via an armature. Fig. 2 depicts the core simulation features at a random time step.

B. Ablation study for a single time-step

In the initial comparative analysis, a fully deployed cylindrical vertical net was allowed to deform under the influence of the simulated ocean currents. For the reconstruction analysis, 9 control points were placed uniformly on the surface of the net, with the number of vertices (N), nodes (n) and node neighbours (m) set at 1134, 52 and 6, respectively. Fishing net shape at a random time frame was then chosen as ground truth, and five experiments were conducted with two templates: a flat (F) and a cylindrical (C) mesh. The five experiments use: (1) standard EDG objective function, EDG objective functions modified by (2) only adding distance constraint and (3) only adding surface normal constraint respectively, (4) EDeformNet (Ours), which includes both constraints, and (5) using the result attained from standard EDG as the initial A_i, t_i for EDeformNet. The results obtained from these experiments are illustrated in Fig. 3.

The reconstruction error was defined as the mean of the Euclidean distances between the vertices of the ground truth mesh (v_i^{gt}, $i \in \{1, ..., N\}$) and the vertices of the reconstructed mesh generated by the EDG ($\hat{v_i}$, $i \in \{1, ..., N\}$). This is defined for each frame as the Root Mean Squared Error (RMSE) and computed as:

$$RMSE = \frac{1}{N}\sum_{i=1}^{N} ||v_i^{gt} - \hat{v_i}||_2. \tag{13}$$

The performance of each test case was evaluated on the basis of the RMSE and is shown in Fig. 3. These results show that regardless of the template type, integrating both constraint terms has led to a lower RMSE when compared to EDG. When employing the F template, EDG with E_{surfn} was able to better preserve the shape of the net in relation to the EDG. Adding E_{dist} instead further improves the result. Only adding one constraint (E_{surfn}/E_{dist}) has shown comparatively larger errors for C template. In terms of templates, the C template is closer to the ground truth and therefore

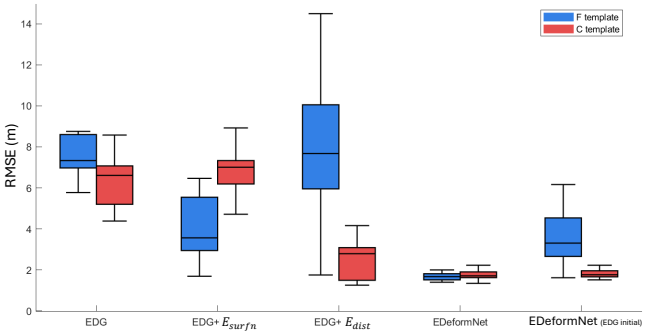

Fig. 4. 3D reconstruction performance comparison of five methods (EDG, EDG+E_{surfn}, EDG+E_{dist}, EDeformNet, and EDeformNet with EDG result as initial) for F - flat template, C - cylindrical template with 30 ground truth samples

exhibits better performance for both EDG and EDeformNet. However, the reduction in RMSE for EDeformNet when using the C template was rather limited (1.35%), whereas EDG´s reduction is very notable (46.89%). It seems evident that EDeformNet is better able to deform a more universal template such as the rectangular flat F shape into arbitrary target geometries. For the C template, incorporating the EDG solution as an initial of EDeformNet improved the results marginally (0.05%) while for the F template, the performance degraded from 0.01%.

In order to further confirm these findings, the five reconstruction methods were conducted on 30 randomly selected ground truth frames (in which the net was also fully deployed). Here, RMSE for each sample in each method was taken as the mean of 5 runs with random Gaussian noise added to the values of the control point correspondences. Then, each set was repeated with the two templates. The result, illustrated in Fig. 4, depicts a similar trend where the C template reveals a marginal improvement of less than 4.1% over the F template. This modest difference demonstrates that even though the F template may be less similar to the physical shape of the net, once the additional constraints are integrated, it can adapt effectively and achieve accuracies nearly on par with the C template. Moreover, using a poor standard EDG result (F template) as the starting point of EDeformNet was ineffective. Overall, the results confirm that the F template can be a viable and, in some cases, a superior starting point — particularly when the target shape diverges from an ideal cylinder, or when a more flexible, general-purpose initial guess is desirable.

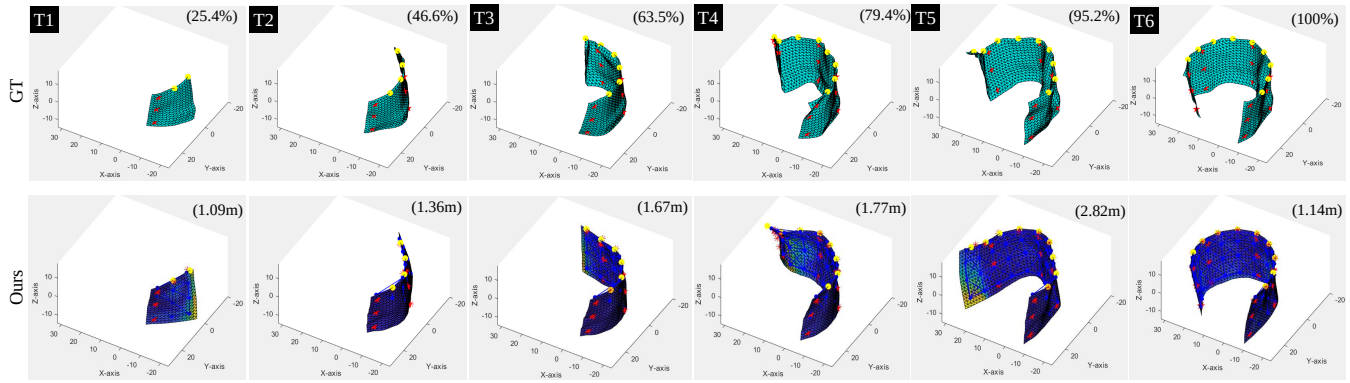

Fig. 5. Ground Truths (GTs) and 3D reconstruction (Ours) results at six time steps in the sequence (control points shown in red, floats shown in yellow, RP shown in brackets on GT, RMSE shown in brackets on Ours); RP - Released portion, as a percentage of the full net.

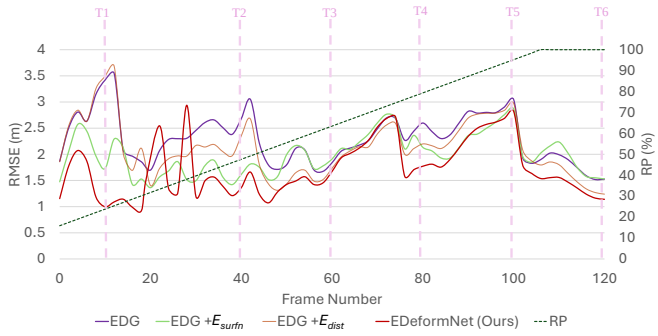

Fig. 6. RMSE evolution for EDG, EDG+E_{surfn}, EDG+E_{dist}, and EDeformNet (Ours) over the reconstruction of a dynamic temporal sequence scene where the fishing net is progressively released in the water (RP - Released portion, as a percentage of the full net).

C. Ablation study for a temporal sequence

In order to evaluate the performance of EDeformNet during temporal sequences, a blender simulation of a net dropping scenario was reconstructed using EDeformNet and other modes of the objective function. 15 uniformly distributed control points that provide both location and surface normal data (deep-sea acoustic sensors/net correspondences) and 9 control points uniformly distributed in the top layer of the net (GPS sensors/float correspondences) that provide only location data were considered. The flat mesh (F) in Fig. 3 was used as the template. At each time step, the template was cropped based on the released length of the net. This is defined as *preprocessing template* in Fig. 1(b). Fig. 5 shows six time steps (T1-T6) extracted from the temporal sequence with their respective 3D reconstructions using EDeformNet. Fig. 5 also denotes the RMSE value for each reconstruction.

To evaluate the performance of EDeformNet in the net drop scenario, RMSE was calculated for each frame. In this study, a scenario with 120 Blender frames was considered. Distributing these frames across the entire deployment process is considered to provide a comprehensive insight about the entire purse seining timeline operation. The full variation of RMSE is illustrated in Fig. 6. In addition, the RMSE calculated for EDG and EDG with each separate constraint is also plotted in the same graph. Here, each method was evaluated five times, introducing random Gaussian noise to the control point correspondences in each trial. The mean of these trials was then computed and plotted to ensure a more robust and reliable comparison of the RMSE values. For the most part, it can be seen that the RMSE is the lowest for the proposed EDeformNet. Especially in the initial frames where only a few sensors are active, EDeformNet performs significantly better. In general, RMSEs in all modes fluctuate periodically. The reason is that between two sensor layers, the 3D reconstruction can deviate slightly from the ground truth (see T5 in Fig. 5 for instance). However, this is corrected when the next control point is released into the sea (see T6 in Fig. 5). EDeformNet shows a 25.8% improvement in cumulative RMSE compared to EDG, with other similar circling purse seine simulations showing a similar trend and values in the comparison (see more examples in the accompanying video).

D. Control point architecture analysis

This simulation experiment consists of a fully extended deforming net over 500 frames (see Fig. 7(a)). The simulation replicates the deformation behaviour captured by Fig. 1(a1). The performance of the 3D reconstruction was evaluated under control points uniformly spaced over the net with 9, 15 and 27 sensors, respectively. In all cases, the F-template in Fig. 3 was used. Fig. 7(b) illustrates a random instance pulled out from the sequence, while Fig. 7(c-e) shows the reconstruction comparison between EDG and EDeformNet for each control point arrangement. It is evident from the results that the proposed EDeformNet method outperforms EDG in every sensor arrangement with 29%, 46%, and 85% cumulative RMSE improvements, respectively. Especially for the sparse arrangement with 9 control points, a rather realistic scenario, EDeformNet showed a remarkable improvement.

E. Performance comparison with alternative deformation methods

In this experiment, the RMSE of EDeformNet was compared against two widely adopted non-rigid mesh deformation methods—TPS and ARAP (MATLAB implementations sourced from [34] and [35]) —across 120 frames of a purse seine net deployment. To assess robustness and accuracy,

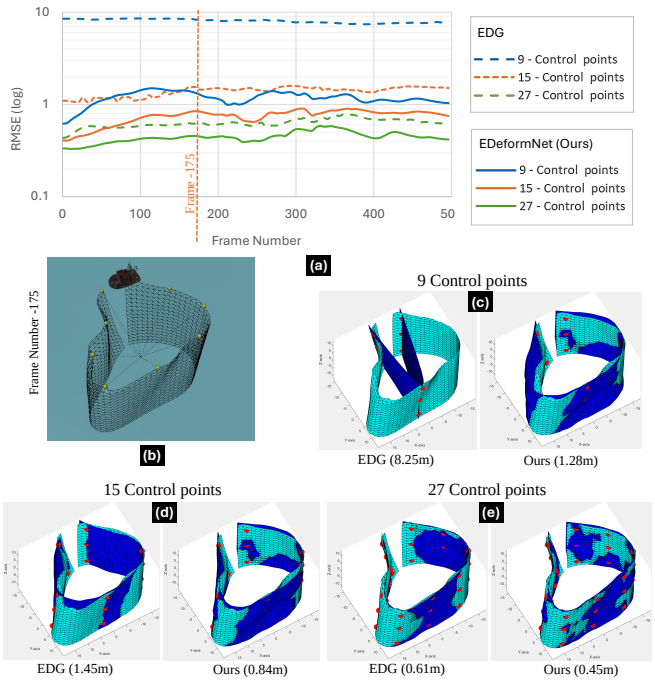

Fig. 7. Control points number performance comparison for EDG and EDeformNet: (a) RMSE for temporal sequence of a deforming fully deployed net over 500 frames. (b) Blender deformation pattern pulled out at the 175th frame. (c,d,e) 3D reconstruction for 9, 15, 27 equally distributed control point arrangements respectively (RMSEs in brackets and control points are shown in red on GTs).

three different temporal simulation scenarios were examined, each corresponding to distinct current patterns. Fig. 8 illustrates the 120^{th} frame of each scenario, depicting the fully deployed net in each case. The comparative results are presented in Fig. 9 (Refer to the accompanying video for further visualizations). The solid, dashed, and dotted lines in the figure represent the mean error for each method, while the shaded regions indicate the standard deviation for 5 experimental runs while introducing random Gaussian noise into the sensor readings within a ± 3.0 m error bound. This bound was selected based on the accuracy specifications of commercially available underwater acoustic sensor units.

In the first scenario, EDeformNet consistently achieved the lowest error across all frames, exhibiting minimal fluctuations and a notably narrow uncertainty range. This outcome implies that EDeformNet is more resilient to noise in the control point measurements. By contrast, TPS and ARAP started with higher initial errors, with ARAP showing the greatest variability, particularly during the early frames characterized by rapid deformations. As the sequence progressed, all methods converged to lower error levels, although EDeformNet continued to deliver superior accuracy and stability. These findings suggest that EDeformNet is the most reliable option for modeling purse seine net deformations, adeptly handling varying degrees of sensor noise while preserving accuracy. In the second and third scenarios, TPS and ARAP performed more comparably to EDeformNet, yet EDeformNet still maintained consistently higher accuracy. On average, EDeformNet shows 27.5% and 29.2% improvements when

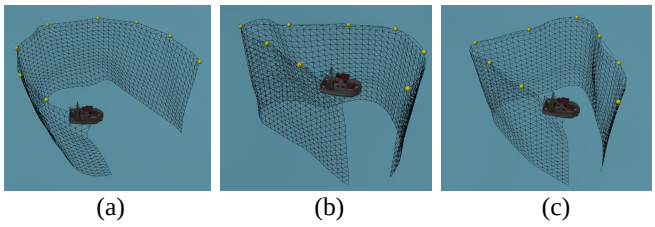

Fig. 8. Net deployment recorded at the 120^{th} frame for the three temporal sequence scenarios (a)-(c) used for the performance comparison.

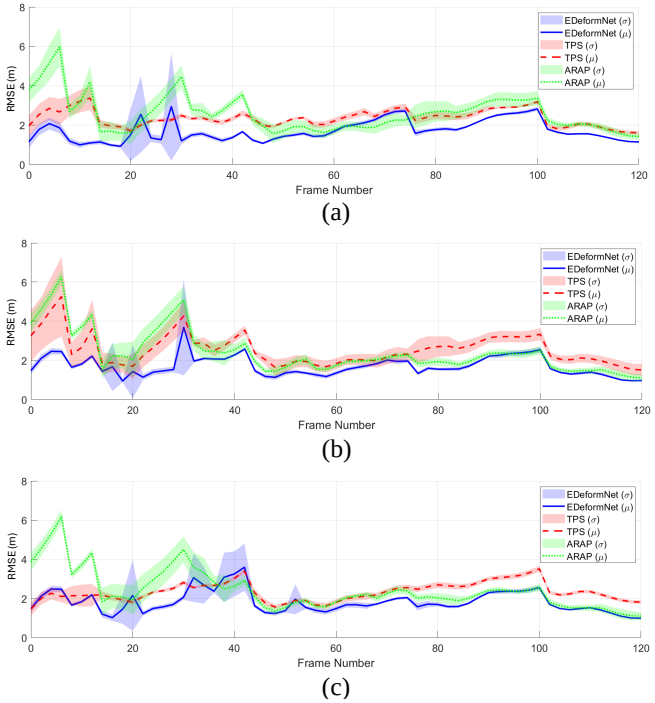

Fig. 9. Performance comparison between EDeformNet, ARAP [34] and TPS [35] for the net dropping scenario. (a)-(c) represent the three temporal sequence scenarios (μ - Mean of the RMSE, σ - Standard deviation of the RMSE for 5 runs per each test).

compared to TPS and ARAP errors, respectively. Overall, these results highlight EDeformNet's robustness and reliability for simulating purse seine net deformations under diverse environmental conditions.

With regards to computational requirements, it was observed that on a high performance laptop computer (Intel Core i7-12650H CPU @ 2.70 GHz, 24 GB RAM, NVIDIA RTX 3060 GPU), the average processing time per frame was approximately 0.2s for TPS, 0.4s for ARAP, and 2.5s for EDeformNet. While EDeformNet demonstrates superior accuracy from incorporating additional constraints, this comes with a moderate increase in computational time. However, this processing time remains well within acceptable limits in purse seining, where a full fishing net deployment operation would typically take around 10 minutes. Given that one frame at every 5.0 seconds is sufficient to extract 120 meaningful insights from the 3D reconstruction process, the computational overhead of EDeformNet appears well within operational constraints, and therefore a viable solution for real-world deployments.

VI. Conclusions and Future Direction

This paper has introduced EDeformNet, a pioneering approach for real-time 3D reconstruction of purse seine fishing nets, extending EDG for this purpose, coupled with sparse positional measurements from trackers attached to the net. EDeformNet is proven to be a notable improvement over the performance of standard EDG by adding suitable constraints to the objective function tailored to a fishing net scenario. Blender-based simulations were developed to mimic net deployment, and extensive analyses were conducted to evaluate the performance of EDeformNet with different templates and control point arrangements. Results demonstrate that the novel scheme can better preserve the typical deformations and movements of a net present during a fishing operation, leading to a minimum of 25% improvement over standard EDG in temporal reconstruction accuracy. It also shows 27.5% and 29.2% improvements compared to TPS and ARAP methods. EDeformNet presents a promising avenue for advancing the capabilities of fisheries monitoring and management, providing a novel tool to help fishing boat captains in their daily jobs. The work is currently being implemented on a fishing vessel for field validation.

References

[1] P. Pravin and B. Meenakumari, "Purse seining in india - a review," *Indian Journal of Fisheries*, vol. 63, 2016.

[2] *MCS Practitioners Introductory Guide: Purse Seine Fishing*, 2021.

[3] H. Steven, R., W. Peter, G., J. Claudio, Castillo, D. Jemery, H. Paul, A., H. William, J., M. Jed, M. Arni, S. Robert, D., S. P. Joe, S. Inna, and P. Graham, M., "The western and central pacific tuna fishery: 2022 overview and status of stocks," 2022.

[4] I. Wijegunawardana, J. V. Miro, I. Quincoces, L. Zhao, and S. Huang, "Adaptive vessel navigation for purse seine fishing net deployment," in *Proceedings of Martech 2025, Technology for Aquaculture and fisheries*. Martech, 2025, p. 33.

[5] H. M. Toonen and S. R. Bush, "The digital frontiers of fisheries governance: fish attraction devices, drones and satellites," *Journal of environmental policy & planning*, vol. 22, no. 1, pp. 125–137, 2020.

[6] Q. Hou, C. Zhou, R. Wan, J. Zhang, and F. Xue, "Application of feature point matching technology to identify images of free-swimming tuna schools in a purse seine fishery," *Journal of Marine Science and Engineering*, vol. 9, no. 12, p. 1357, 2021.

[7] S. A. Hosseini, C.-W. Lee, H.-S. Kim, J. Lee, and G.-H. Lee, "The sinking performance of the tuna purse seine gear with large-meshed panels using numerical method," *Fisheries Science*, vol. 77, pp. 503–520, 2011.

[8] C.-W. Lee, J.-H. Lee, B.-J. Cha, H.-Y. Kim, and J.-H. Lee, "Physical modeling for underwater flexible systems dynamic simulation," *Ocean Engineering*, vol. 32, no. 3, pp. 331–347, 2005.

[9] E. Tretschk, N. Kairanda, M. BR, R. Dabral, A. Kortylewski, B. Egger, M. Habermann, P. Fua, C. Theobalt, and V. Golyanik, "State of the art in dense monocular non-rigid 3d reconstruction," in *Computer Graphics Forum*, vol. 42, no. 2. Wiley Online Library, 2023, pp. 485–520.

[10] M. Zollhöfer, P. Stotko, A. Görlitz, C. Theobalt, M. Nießner, R. Klein, and A. Kolb, "State of the art on 3d reconstruction with rgb-d cameras," in *Computer graphics forum*, vol. 37, no. 2. Wiley Online Library, 2018, pp. 625–652.

[11] R. W. Sumner, J. Schmid, and M. Pauly, "Embedded deformation for shape manipulation," in *ACM siggraph 2007 papers*, 2007, pp. 80–es.

[12] B. Deng, Y. Yao, R. M. Dyke, and J. Zhang, "A survey of non-rigid 3d registration," in *Computer Graphics Forum*, vol. 41, no. 2. Wiley Online Library, 2022, pp. 559–589.

[13] T. Kitamura, N. Iwamoto, H. Kawasaki, and D. Thomas, "A two-step approach for interactive animatable avatars," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 14496 LNCS, p. 491 – 509, 2024.

[14] A. Schmidt, O. Mohareri, S. DiMaio, M. C. Yip, and S. E. Salcudean, "Tracking and mapping in medical computer vision: A review," *Medical Image Analysis*, p. 103131, 2024.

[15] R. Yunus, J. E. Lenssen, M. Niemeyer, Y. Liao, C. Rupprecht, C. Theobalt, G. Pons-Moll, J.-B. Huang, V. Golyanik, and E. Ilg, "Recent trends in 3d reconstruction of general non-rigid scenes," in *Computer Graphics Forum*. Wiley Online Library, 2024, p. e15062.

[16] Y. Wang, X. Yan, M. Jiang, and J. Zheng, "Research on non-rigid structure from motion: A literature review," *Journal of Fiber Bioengineering and Informatics*, vol. 8, no. 4, p. 751 – 760, 2015.

[17] D. Fuentes-Jimenez, D. Pizarro, D. Casillas-Pérez, T. Collins, and A. Bartoli, "Deep shape-from-template: Single-image quasi-isometric deformable registration and reconstruction," *Image and Vision Computing*, vol. 127, 2022.

[18] Y. Liu, X. Peng, W. Zhou, B. Liu, and A. Gerndt, "Template-based 3d reconstruction of non-rigid deformable object from monocular video," *3D Research*, vol. 9, pp. 1–12, 2018.

[19] H. Dai, N. Pears, and W. Smith, "Non-rigid 3d shape registration using an adaptive template," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018.

[20] K. Li, J. Yang, Y.-K. Lai, and D. Guo, "Robust non-rigid registration with reweighted position and transformation sparsity," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 6, pp. 2255–2269, 2019.

[21] T. Yu, K. Guo, F. Xu, Y. Dong, Z. Su, J. Zhao, J. Li, Q. Dai, and Y. Liu, "Bodyfusion: Real-time capture of human motion and surface geometry using a single depth camera," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 910–919.

[22] K. Fujiwara, K. Nishino, J. Takamatsu, B. Zheng, and K. Ikeuchi, "Locally rigid globally non-rigid surface registration," in *2011 International Conference on Computer Vision*, 2011, pp. 1527–1534.

[23] R. Huang, J. Zhao, F. Duan, X. Li, C. Liu, X. Deng, Z. Pan, Z. Wu, and M. Zhou, "Automatic craniofacial registration based on radial curves," *Computers Graphics*, vol. 82, pp. 264–274, 2019.

[24] H. Jiang and L. Zhang, "Interactive animating virtual characters with the human body," in *Advances in Multimedia Information Processing–PCM 2015: 16th Pacific-Rim Conference on Multimedia, Gwangju, South Korea, September 16-18, 2015, Proceedings, Part I 16*. Springer, 2015, pp. 611–620.

[25] Z. Li, Y. Ji, W. Yang, J. Ye, and J. Yu, "Robust 3d human motion reconstruction via dynamic template construction," in *2017 International Conference on 3D Vision (3DV)*. IEEE, 2017, pp. 496–505.

[26] K. Pan, S. Zhang, L. Zhao, S. Huang, Y. Zhang, H. Wang, and Q. Luo, "3d reconstruction of tibia and fibula using one general model and two x-ray images," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 4732–4738.

[27] Y. Zhang, R. Falque, L. Zhao, Y. Chen, S. Huang, and H. Li, "Structure-to-shape aortic 3-d deformation reconstruction for endovascular interventions," *IEEE Transactions on Robotics*, 2023.

[28] A. Alempijevic, T. Patten, D. Banuelos, and R. Guenot-Falque, "Skirting line annotation via deformation modelling," in *Australasian Conference on Robotics and Automation*, 2021.

[29] A. Bozic, P. Palafox, M. Zollhofer, J. Thies, A. Dai, and M. Nießner, "Neural deformation graphs for globally-consistent non-rigid reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1450–1459.

[30] W. Feng, H. Cai, J. Hou, B. Deng, and J. Zhang, "Differentiable deformation graph-based neural non-rigid registration," *Communications in Mathematics and Statistics*, vol. 11, no. 1, pp. 151–167, 2023.

[31] O. Sorkine and M. Alexa, "As-rigid-as-possible surface modeling," in *Symposium on Geometry processing*, vol. 4. Citeseer, 2007, pp. 109–116.

[32] W. Qin, Y. Hu, Y. Sun, and B. Yin, "An automatic multi-sample 3d face registration method based on thin plate spline and deformable model," in *2012 IEEE International Conference on Multimedia and Expo Workshops*. IEEE, 2012, pp. 453–458.

[33] *Blender Manual*, Blender Foundation, 2023, accessed: Sep. 16, 2024. [Online]. Available: https://docs.blender.org/manual/en/4.2/index.html

[34] Y. Yang, "3d thin plate spline warping function," MATLAB Central File Exchange, 2025, retrieved February 19, 2025. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/37576-3d-thin-plate-spline-warping-function

[35] A. Jacobson, "gptoolbox," GitHub, 2025, retrieved February 19, 2025. [Online]. Available: https://github.com/alecjacobson/gptoolbox