Some Research Questions for SLAM in Deformable Environments

Shoudong Huang[1], Yongbo Chen[1], Liang Zhao[1], Yanhao Zhang[1], and Mengya Xu[1]

***Abstract*— SLAM in deformable environments is a very challenging research topic. Some research works have been presented by different research groups in the past few years. However, there are still some challenging research questions remaining unanswered. This paper discusses some of these research questions focusing on the case when point features are used to describe the deformable environments. The SLAM problems are formulated as extensions of point feature based SLAM in static environments, including both optimisation based offline SLAM and filter based online SLAM. To illustrate the problems and questions more clearly, some concepts and results using simple 2D examples are presented. The MATLAB source codes of the results are made publicly available (https://github.com/cyb1212/DeformableSLAM2D.git) to help the readers understand the problems more clearly.**

I. INTRODUCTION

Significant progress has been made over the decades in the study of SLAM in static environments [1]. However, when the environment has deformations such as when a surgical robot is navigating in an internal body environment, SLAM needs to build a time-varying 3D map of the soft tissues and estimate the location of the sensor relative to the tissues. This poses a very challenging problem since the sensor is moving whilst the environment is deforming.

SLAM in deformable environments can find very important applications in many different areas such as (1) minimally invasive robotic surgery; (2) animal body shape reconstruction; (3) human body motion tracking (e.g. for sports performance analysis); and (4) motion capture in virtual reality and computer games.

In the last few years, a few research groups have initiated some research on this challenging topic. When an RGB-D sensor is used for the observations, a common approach is to deform the prior or built map based on the observation directly, using different ways to represent the deformation. For example, DynamicFusion algorithm [2] and SurfelWarp [3] apply dual quaternions [4] to represent the warp field, and provide real-time reconstruction and tracking systems for non-rigid scenes, one using volumetric data structure and the other using a surfel based representation of the geometry. VolumeDeform [5] uses the as-rigid-as-possible volume regularization of the space embedding to deform the surface [6]. Fusion4D [7] uses embedded deformation [8] to deform the map. For surgical cases, MIS-SLAM [9] is a GPU-based real-time stereo vision SLAM algorithm in deformable inside-body environment utilising the pose estimation from the well-known ORB-SLAM system [10].

In some deformable environments with the spatial limitation, like gastrointestinal tract and urinary tract, the endoscope, which is commonly a monocular camera, is the main sensor to perform inspection tasks. However, SLAM in deformable environments using a monocular camera is more challenging, and the camera poses in the global coordinate frame are more difficult to solve without introducing additional constraints such as camera motion model or the feature motion model, because of the lack of edges in the SLAM graph and the unavailable depth information. A very recent work, DefSLAM, proposed in [11], presents a complete framework fusing energy-minimisation pose estimation and isometric Non-Rigid Structure-from-Motion (NRSfM) techniques. Recent important research work on NRSfM includes [12], [13], [14], [15], [16], and [17]. As a general tool to recover depth for multiple monocular images, the NRSfM method is an important building block for realising the deformable SLAM with monocular sensors. The NRSfM problem is still regarded as an open challenging computer vision problem suffering one or several limitations amongst solution ambiguities, inaccuracy, ill-posedness, inability to handle changing data, lack of robustness for noises, high computational cost, and inability or poor performance in discontinuous environment. In [18], a sum-of-Gaussians filter is proposed to jointly retrieve camera auto-calibration, camera pose, and the 3D reconstruction of a non-rigid object. Very recently, some scholars use the learning algorithms to deal with the camera poses estimation and NRSfM with some promising results, such as Endo-Depth-and-Motion [19], N-NRSfM approach [20], Neural Deformation Graphs [21], and so on.

Although some good progress has been made in the area of SLAM in deformable environments. There are still many fundamental questions that have not been clearly answered yet. For example, what is the observability of the problem under different scenarios? Can we obtain a consistent result on the uncertainties of the estimate?

This paper discusses some of the important questions that require clear answers and further investigations from a point feature based SLAM point of view. We hope this discussion is useful for the researchers in the robotics and computer vision communities who are interested in working on this challenging research topic.

II. PROBLEM FORMULATION

We focus on the case when point features are used to represent the map.

*This work was supported by Australian Research Council (ARC) Discovery project DP200100982.

[1] Centre for Autonomous Systems, University of Technology Sydney, NSW 2008, Australia. shoudong.huang@uts.edu.au

A. Information available

Consider a scenario of a robot (or a mobile sensor such as a camera) moves around in an environment that contains point features (e.g. some points on a piece of cloth, or on a human/animal body, or on an inside body organ). The robot pose at time step i is denoted as $P_i = (\boldsymbol{x}_i, \boldsymbol{R}_i)$ where $\boldsymbol{x}_i$ is the position and $\boldsymbol{R}_i$ is the orientation[1]. The point features can change their positions over time because the "environment" (e.g. human body) can move and the "shape" of the environment is deformable. The position of the j-th feature at time step i is denoted as $\boldsymbol{f}_i^j$.

In general, the following information is available for the SLAM problem in deformable environments.

- Observations: the robot can observe some of the features at each time step, for example, at time step i, the robot can observe feature $\boldsymbol{f}_i^j$ from pose P_i;
- Dynamic models of the features: some knowledge on the possible motion of the features should be known, this includes both the global rigid motion and the local deformation of the features;
- Odometry (optional): relative pose information between two consecutive robot poses (such as P_i and P_{i+1}).

While the observation information and odometry information are both commonly used in the SLAM literature, the dynamic model of the features only appears in SLAM for dynamic or deformable environments. It should be pointed out clearly that for SLAM in static environments, the "dynamic model of the feature" does exist and the information is very strong, which is "all the features are stationary with zero uncertainty". On the other hand, for SLAM in dynamic or deformable environments, the "dynamic model of the feature" is "the feature is moving following a certain model but with some/large uncertainty". Thus the information is a lot weaker. This makes the SLAM problem in deformable environments very difficult and challenging.

One important question for SLAM in deformable environments is how to describe this information clearly.

Research question 1: How to describe the dynamic models of the features correctly and clearly? Or, how to clearly describe the possible global rigid motion and the possible local deformation of the features/environment, together with the associated uncertainties?

B. One possible assumption on the available information

Assume the observation of feature $\boldsymbol{f}_i^j$ from pose P_i is $\boldsymbol{z}_i^j$ given by

$$\boldsymbol{z}_i^j = \boldsymbol{h}_i^j(\boldsymbol{x}_i, \boldsymbol{R}_i, \boldsymbol{f}_i^j) + \boldsymbol{z}_{ij}^e, \tag{1}$$

where $\boldsymbol{h}_i^j$ is the observation function depending on the sensor used, $\boldsymbol{z}_{ij}^e \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{ij})$ is the zero mean Gaussian noise in this measurement.

For the dynamic models of the features, we assume that they can be separated into two parts.

[1]In this paper, we use the same notation to represent a pose which can be either 2D or 3D depending on the scenarios. Similarly, we use the same notation for both 2D and 3D point features.

Smooth constraints: One part shows the relation between the feature position at time i ($\boldsymbol{f}_i^j$) and the feature position at time $i+1$ ($\boldsymbol{f}_{i+1}^j$),

$$\boldsymbol{r}_{i,i+1}^j = \boldsymbol{m}_i^j(\boldsymbol{f}_i^j, \boldsymbol{f}_{i+1}^j) + \boldsymbol{r}_{ij}^e, \tag{2}$$

where $\boldsymbol{r}_{i,i+1}^j$ is the measurement (knowledge) of the constraint between j-th feature at i-th time step and at $i+1$-th time step, $\boldsymbol{r}_{ij}^e \sim \mathcal{N}(\mathbf{0}, \widehat{\boldsymbol{\Sigma}}_{ij})$ is the zero mean Gaussian noise in this measurement, $\boldsymbol{m}_i^j$ is a function depending on the motion of the feature, which should cover both the global rigid motion and the local deformation.

Structure constraints: The other part is the relation between two features at the same time step. This is necessary since the features are related to each other and cannot move independently. We express this relation as follows.

$$\boldsymbol{q}_i^{j,k} = \boldsymbol{s}_i^{j,k}(\boldsymbol{f}_i^j, \boldsymbol{f}_i^k) + \boldsymbol{q}_{ijk}^e, \tag{3}$$

where $\boldsymbol{q}_i^{j,k}$ is the constraint between features $\boldsymbol{f}_i^j$ and $\boldsymbol{f}_i^k$ at time step i, and $\boldsymbol{q}_{ijk}^e \sim \mathcal{N}(\mathbf{0}, \bar{\boldsymbol{\Sigma}}_{ijk})$ is the zero mean Gaussian noise in this measurement. $\boldsymbol{s}_i^{j,k}$ is a function which depends on the structure property of the environment. One example is the distance between the two points. This constraint commonly describes the motion relationship between every two nearby points $\boldsymbol{f}_i^j$ and $\boldsymbol{f}_i^k$. We use $\mathcal{E}_i$ to denote the set of all the pairs of features that have connection at time step i, that is, $(j,k) \in \mathcal{E}_i$.

The (optional) odometry between pose P_i and pose P_{i+1} is in the form of relative pose $(\boldsymbol{x}_{i,i+1}, \boldsymbol{R}_{i,i+1})$ given by

$$\begin{aligned} \boldsymbol{x}_{i,i+1} &= \boldsymbol{R}_i^\top(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i) + \boldsymbol{x}_i^e, \\ \boldsymbol{R}_{i,i+1} &= \boldsymbol{R}_i^\top \boldsymbol{R}_{i+1} \boldsymbol{R}_i^e, \end{aligned} \tag{4}$$

where $\boldsymbol{x}_i^e$ and $\boldsymbol{R}_i^e \in SO(3)$ are odometry noises. We assume $\boldsymbol{x}_i^e \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{x}})$ is a zero mean Gaussian noise with covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{x}}$. Similar to [22][23], we assume $\boldsymbol{R}^e$ follows an isotropic Langevin distribution.

Once the available information is clearly described, the SLAM problem can be formulated as: *given the observations, the dynamic models of the features, and the (optional) odometry, estimate the feature positions and the robot poses at all the different time steps.*

C. Optimisation based offline SLAM

In optimisation based offline SLAM, all the available information is used to estimate all the robot poses and the feature positions at all the different time steps. In another word, we estimate multiple maps (one map at each time step) together with the robot poses at each time step.

Assuming the robot moves N steps from time step 0 to time step N, and there are M point features in the environment, the set of the states to be estimated (in the world coordinate frame) is

$$X^w = \{\boldsymbol{f}_0^1, \cdots, \boldsymbol{f}_0^M, P_1, \boldsymbol{f}_1^1, \cdots, \boldsymbol{f}_1^M, \cdots, P_N, \boldsymbol{f}_N^1 \cdots, \boldsymbol{f}_N^M\} \tag{5}$$

assuming the robot pose P_0 defines the world coordinate frame. That is, $\boldsymbol{x}_0 = \mathbf{0}, \boldsymbol{R}_0 = I$. Thus P_0 is not in the set of the states to be estimated.

Now we can formulate SLAM in deformable environment as a non-linear least squares optimisation problem:

$$
\begin{aligned}
\min_{X^w} \Bigg(& \sum_i ||\boldsymbol{x}_i^{i+1} - \boldsymbol{R}_i^\top(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i)||^2_{\boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1}} \\
& + \sum_i d_{SO}\left(\boldsymbol{R}_i^{i+1}, \boldsymbol{R}_i^\top \boldsymbol{R}_j\right)^2 \\
& + \sum_i \sum_j ||\boldsymbol{z}_i^j - \boldsymbol{h}_i^j(\boldsymbol{x}_i, \boldsymbol{R}_i, \boldsymbol{f}_i^j)||^2_{\boldsymbol{\Sigma}_{ij}^{-1}} \\
& + \sum_i \sum_{(j,k)\in\mathcal{E}_i} ||\boldsymbol{q}_i^{j,k} - \boldsymbol{s}_i^{j,k}(\boldsymbol{f}_i^j, \boldsymbol{f}_i^k)||^2_{\bar{\boldsymbol{\Sigma}}_{ijk}^{-1}} \\
& + \sum_i \sum_j ||\boldsymbol{r}_{i,i+1}^j - \boldsymbol{m}_i^j(\boldsymbol{f}_i^j, \boldsymbol{f}_{i+1}^j)||^2_{\widehat{\boldsymbol{\Sigma}}_{ij}^{-1}} \Bigg)
\end{aligned}
\tag{6}
$$

where $d_{SO}(\star, \bullet)$ represents a distance function on the Lie group $SO(2)$ or $SO(3)$. If odometry information is not available, then the first two terms in (6) will be removed.

Fig. 1 shows a graph structure of the problem where $M = 3$ features exist and robot observes all the features in each of the 3 time steps ($N = 2$). This can be regarded as an extension of graph-based SLAM with static features [1].

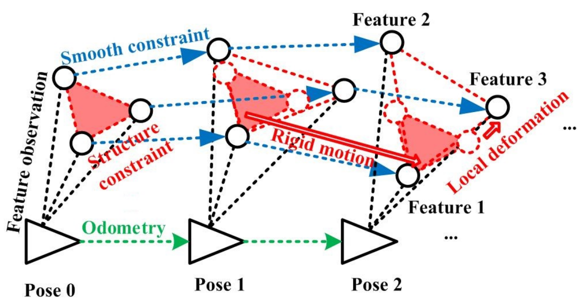


Fig. 1. The graph structure of the problem of SLAM in deformable environments. The black dot lines show the robot to feature observations; the red dot lines show the structure constraints; the green arrows show the odometry measurements; the blue arrows show the smooth constraints. The rigid body motion and the local deformation are also illustrated in the figure. It is an extension of graph-based SLAM for static environments.

D. Filter based online SLAM

In filter based online SLAM, at time step i, the available information up to time i is used to estimate the robot pose and the feature positions at time step i. The feature positions (map) and the robot pose at time i form the state of the whole dynamic system, that is

$$X_i^w = \{P_i, \boldsymbol{f}_i^1, \cdots, \boldsymbol{f}_i^M\}. \tag{7}$$

The process model of the whole dynamic system consists of the odometry (4) and the smooth constraint (2). The observation model of the whole dynamic system consists of the robot to feature observation (1) and the structure constraint (3).

Extended Kalman filter (EKF) or other filtering techniques could be used to solve this state estimation problem for a dynamic system, provided that a good linearisation point can always be obtained.

E. Linear case

Now we use a simple 2D linear case as an example to illustrate the results of the SLAM problems mentioned above.

For the linear case, we assume the robot rotation matrix is always an identity matrix. Thus the robot rotation is no longer a variable and the robot pose contains the position only. The feature observation measurement then becomes:

$$\boldsymbol{z}_i^j = \boldsymbol{f}_i^j - \boldsymbol{x}_i + \boldsymbol{z}_{ij}^e, \quad \boldsymbol{z}_{ij}^e \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{ij}). \tag{8}$$

The smooth measurement can be assumed to be:

$$\boldsymbol{r}_{i,i+1}^j = \boldsymbol{f}_{i+1}^j - \boldsymbol{f}_i^j + \boldsymbol{r}_{ij}^e, \quad \boldsymbol{r}_{ij}^e \sim \mathcal{N}(\boldsymbol{0}, \widehat{\boldsymbol{\Sigma}}_{ij}). \tag{9}$$

The structure measurement can be assumed to be:

$$\boldsymbol{q}_i^{j,k} = \boldsymbol{f}_i^k - \boldsymbol{f}_i^j + \boldsymbol{q}_{ijk}^e, \quad \boldsymbol{q}_{ijk}^e \sim \mathcal{N}(\boldsymbol{0}, \bar{\boldsymbol{\Sigma}}_{ijk}). \tag{10}$$

The odometry measurement will be:

$$\boldsymbol{x}_{i,i+1} = \boldsymbol{x}_{i+1} - \boldsymbol{x}_i + \boldsymbol{x}_i^e, \quad \boldsymbol{x}_i^e \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{x}}). \tag{11}$$

For this simple linear case, the optimisation problem (6) becomes a linear least squares (LS) problem which has a closed form solution. In addition, Kalman filter (KF) can be used to solve the online SLAM problem.

Fig. 2 shows a 2D example of the estimation result of LS. The uncertainty ellipses from the obtained covariance matrices are also shown in the figure. They are all circles because the covariance matrices for all the measurement noises are all proportional to an identity matrix and the estimation problem is linear. Similarly, Fig. 3 shows an example of the estimation result from KF. We call them "world centric" because the coordinate frame is decided by the first robot pose.

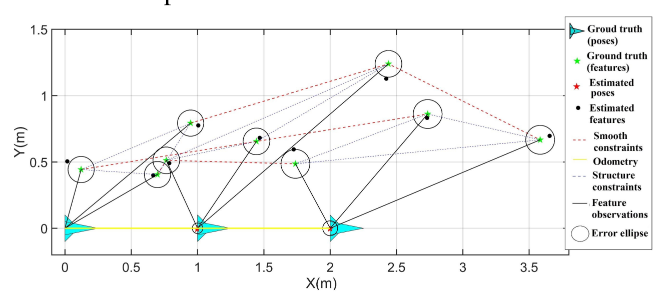


Fig. 2. Estimated result based on world centric formulation using LS.

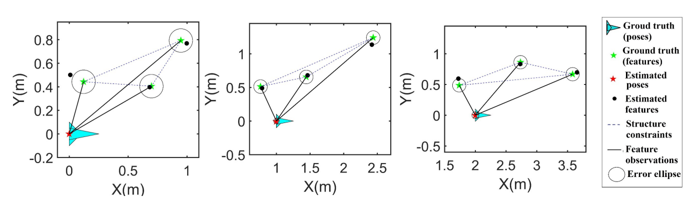


Fig. 3. Estimated result based on world centric formulation using KF at every step.

F. Some discussions on related problems

The constraints (2) and (3) are some simple examples representing the possible feature motion and the deformation structure. The key idea is to express the constraints as functions of the variables (the features at different times).

The constraint in (2) is called "smooth constraint" in this paper. It is similar to the energy-minimisation objective function in [11], this constraint should cover both the global rigid motion and the local deformation. The smooth constraint commonly describes the smooth and low-rank trajectory of the features deforming process [24]. In the NRSfM work [14], there is no such constraint and there is no odometry information, thus the camera poses cannot be estimated and the 3D reconstruction is for the shape in the local frame only.

The constraint in (3) is called "structure constraint" in this paper. This constraint is commonly related to the local deformation property of the deforming object. Depending on the different objects/environments, the commonly used constraints include rigid, inextensible (relaxation of isometric), isometric (e.g. paper, flag, rug, and so on), conformal (e.g. balloon), equiareal, and generic [14]. The regularization term in [8], and the as rigid as possible constraint [6] also belong to this category. The constraint in (3) should be expanded to cover two or more points in multiple time steps to properly describe some of the above mentioned constraints.

The above formulation of SLAM in deformable environments is similar to SLAM with dynamic features. If there is no structure constraint (3), then the problem becomes SLAM in dynamic environments because the motions of the features are independent with each other.

In the point feature based SLAM problem in classical static environments, the smooth constraint and structure constraint both exist with zero uncertainty (no noise), they are implicitly expressed in the "stationary" assumption.

III. OBSERVABILITY AND CONSISTENCY

One key question to ask is whether the information available is enough to make the problem of SLAM in deformable environments solvable.

Research question 2: Under what conditions, the problem of SLAM in deformable environments is solvable? When the problem is solvable, what is the achievable estimate accuracy?

It is easy to see the answer to this question is closely related to **Research question 1** and depends on (i) whether the odometry information is available or not; (2) how many features can be observed at each time step and the observation model; (3) the details of the dynamic models of the features. When the odometry, observation, and the dynamic models of the features are given by (1), (2), (3) and (4), the answer to **Research question 2** could be obtained by analysing the corresponding information matrix.

A. Fisher information matrix

For the nonlinear least squares problem (6), the Fisher information matrix can be computed by

$$\text{FIM} = J^\top P^{-1} J \tag{12}$$

where P is constructed by the covariance matrices $\boldsymbol{\Sigma}_{\boldsymbol{x}}, \boldsymbol{\Sigma}_{ij}, \widehat{\boldsymbol{\Sigma}}_{ij}, \bar{\boldsymbol{\Sigma}}_{ijk}$ and J is the Jacobian matrix.

Whether the Fisher information matrix (Jacobian) is full rank (meaning the SLAM problem is observable [25]) or not depends on the connectivity of the graph structure (e.g. Fig. 1), which is related to the number of observations in each time step, and the number of structure constraints, etc.

B. Unobservable cases

There are clearly some cases when some of the features or robot poses at certain time steps are not observable or with very weak observability. For example, if there is no dynamic model for one particular feature and it is not observed at time step k, then the feature position at time step k is not observable. Assuming that the dynamic models for all the features as in (2) and (3) exist, if one feature is not observed for multiple consecutive steps $k, k+1, \cdots, k+m$, then the observability of that feature during these time steps will be very poor. That is, the uncertainty of that feature at these time steps will be very large.

A linear case example: Consider the 2D linear case with 3 features, if feature 3 is not observed from pose 1 to pose 3, then the uncertainties of feature 3 at time steps $1, 2, 3$ are very large, as shown in Fig. 4.

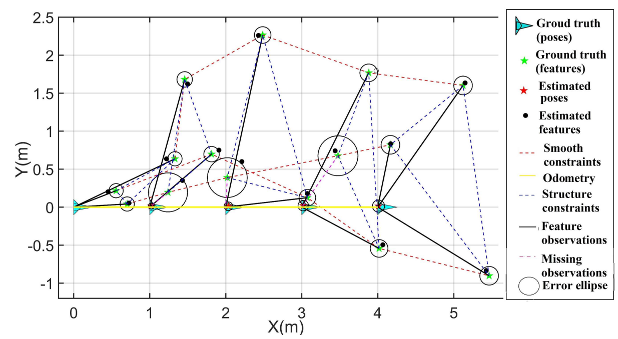


Fig. 4. The result using LS with some missing feature observations.

C. Consistency

Estimate consistency has been an important research topic for EKF based SLAM algorithms [26][27]. Inconsistent estimate (over-confident estimate) can happen in traditional EKF SLAM for static environments especially when the robot orientation error is large. For SLAM in deformable environments, since the features are not stationary, the uncertainty involved is larger than that of SLAM in static environments. Thus the consistency is also a very important research topic.

Research question 3: For a SLAM algorithm in deformable environments, can we calculate the uncertainty of the estimate together with the estimate values? Can we make sure the estimated uncertainty is consistent with the true error?

D. Uncertainty estimate

In some of the research on SLAM in deformable environments, the proposed algorithms only provide an estimate of the map (and robot pose) but do not provide the corresponding uncertainty estimate [11].

When the noises are assumed to be Gaussian and the problem is formulated as a nonlinear least squares problem such as in (6), the Fisher information matrix can be computed by (12), thus the covariance matrix of the estimate could be obtained by computing the inverse of the Fisher information matrix. When an EKF is used in solving the SLAM problem, the corresponding covariance matrix can be obtained directly from the EKF process.

E. Consistency check

To check the consistency of the estimate, simulation with ground truth is needed. One way to check the estimate consistency is to compare the actual estimation error with the 2σ bounds obtained from the covariance matrix. Another method is to perform Monte Carlo simulations and compute Normalised Estimation Error Squared (NEES). For a sample of errors $e_n, n = 1, \cdots, p$, if the dimension of e_n is d, and the covariance matrix is C_n, then the NEES is given by

$$\text{NEES} = \frac{1}{pd}\sum_{n=1}^{p} e_n^{\top} C_n^{-1} e_n. \qquad (13)$$

F. Linear case example

Consider the 2D linear case with 3 poses and 3 deforming features, for LS and KF methods, we respectively run the Monte Carlo simulations 1000 times and get the estimated results. They are mostly within the 2σ bound, as shown in Fig. 5 and Fig. 6, thus the algorithms can be claimed as consistent. We also run a SLAM simulation with 100 frames. The robot and feature position estimation error are shown in Fig. 7. It also verifies our statement of the consistency.

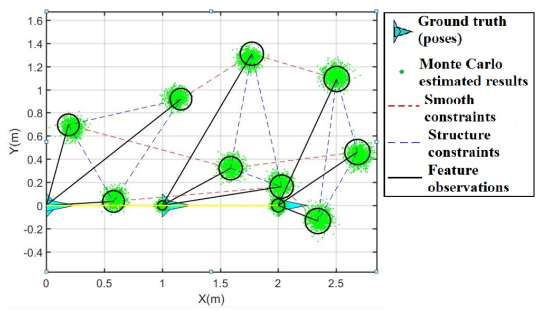


Fig. 5. Consistency verification of the LS result.

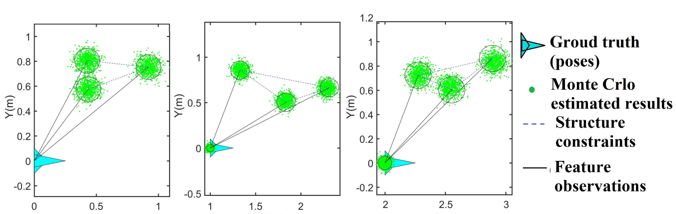


Fig. 6. Consistency verification of the KF result.

For the Monte Carlo simulations with $p = 50$, the two-side 95% probability region of NEES is $[1.4844, 2.5912]$ for e_n with dimension of 2. For the linear case, the NEES results with 100 poses and 3 deforming features using LS and KF methods are respectively shown in Fig. 8 and Fig. 9. The figures show that the NEES results are well bounded by the two-side 95% probability regions and only 3-4 out of the 100 points fall outside this region, which is acceptable and verifies the consistency of the obtained results.

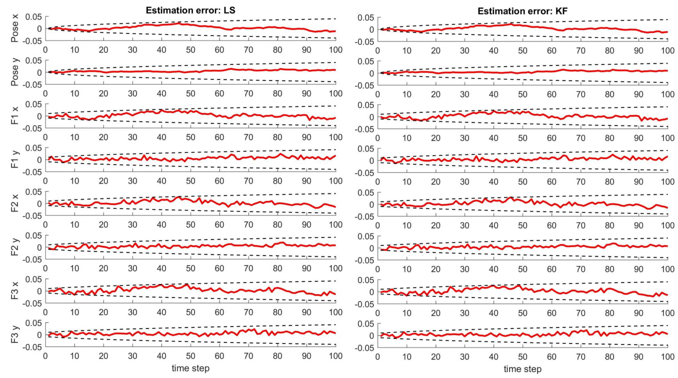


Fig. 7. Robot and feature position estimation error with 2σ bound for LS and KF result. The LS and KF results are from the same noisy measurements.

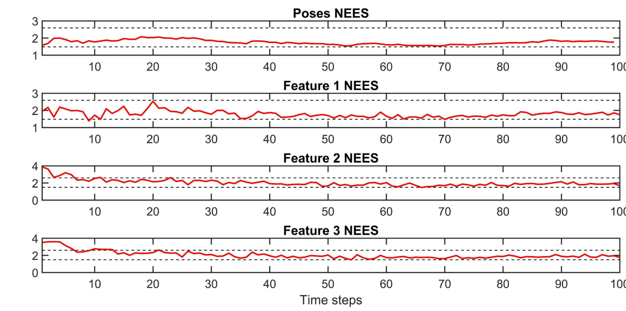


Fig. 8. Robot and feature position NEES result using LS.

IV. DIFFERENT SLAM FORMULATIONS

Typically in a SLAM algorithm, we choose the first robot pose to define the global coordinate frame as in (5). However, this "world centric" SLAM problem formulation can result in inconsistent estimate when traditional EKF is used in the SLAM problem in static environments. On the other hand, the "robot centric" EKF SLAM can improve the estimate consistency [28]. In fact, since all the information available in SLAM is relative information, we have the freedom to choose any coordinate frame to define the robot poses and feature positions in the SLAM problem formulations.

A. World centric

The optimisation based offline SLAM and the filter based online SLAM problem formulations provided in Section II-C and Section II-D are both world centric, where the first robot pose defines the origin of the world coordinate frame.

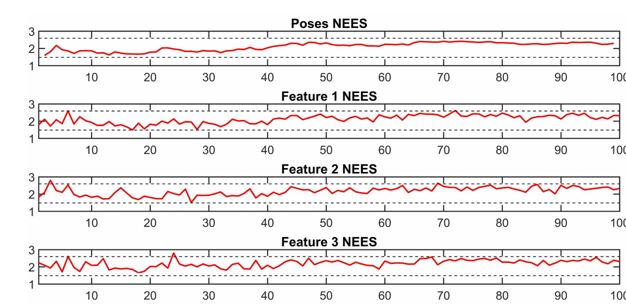


Fig. 9. Robot and feature position NEES result using KF.

B. Robot centric

In the robot centric formulation, the last robot pose is defined as the origin of the coordinate frame. This frame keeps changing when the robot is moving.

For the optimisation based offline SLAM, the set of states to be estimated (in the final robot coordinate frame) is

$$X^r = \{\boldsymbol{f}_0^1, \cdots, \boldsymbol{f}_0^M, P_0, \cdots, \boldsymbol{f}_{N-1}^M, P_{N-1}, \boldsymbol{f}_N^1, \cdots, \boldsymbol{f}_N^M\} \tag{14}$$

where the robot pose P_N defines the robot coordinate frame. That is, $\boldsymbol{x}_N = \boldsymbol{0}, \boldsymbol{R}_N = I$.

C. Map centric

In the map centric formulation, 2 features (for 2D) or 3 features (for 3D) are used to define the coordinate frame [29] [30]. The details on the coordinate transformation can be found in the Appendix in [30].

For SLAM in static environments, the coordinate frame defined by the features does not change. But for SLAM in deformable environments, since the features are moving, the coordinate frame is also changing over time.

For the optimisation based offline SLAM in 2D case, the set of the states to be estimated (in the map coordinate frame defined by $\boldsymbol{f}_0^1$ and $\boldsymbol{f}_0^2$) is

$$X^m = \{P_0, \boldsymbol{f}_0^2(x), \boldsymbol{f}_0^3, \cdots, \boldsymbol{f}_0^M, \cdots, P_N, \boldsymbol{f}_N^1, \cdots, \boldsymbol{f}_N^M\} \tag{15}$$

where $\boldsymbol{f}_0^2(x)$ means the x coordinate of feature $\boldsymbol{f}_0^2$. In this case, $\boldsymbol{f}_0^1 = \boldsymbol{0}$ and $\boldsymbol{f}_0^2(y) = 0$ are not in the set of the states to be estimated while all the poses $P_0, \cdots, P_N$ are in the set.

D. Linear case

For the 2D linear case, the state vectors of the LS optimisation based SLAM for world centric, robot centric and map centric (assuming the first feature at time step 0 is used to define the map coordinate frame) are respectively

$$X^w = \{\boldsymbol{f}_0^1, \cdots, \boldsymbol{f}_0^M, \boldsymbol{x}_1, \boldsymbol{f}_1^1, \cdots, \boldsymbol{x}_N, \boldsymbol{f}_N^1, \cdots, \boldsymbol{f}_N^M\} \tag{16}$$

$$X^r = \{\boldsymbol{f}_0^1, \cdots, \boldsymbol{f}_0^M, \boldsymbol{x}_0, \cdots, \boldsymbol{x}_{N-1}, \boldsymbol{f}_N^1, \cdots, \boldsymbol{f}_N^M\} \tag{17}$$

$$X^m = \{\boldsymbol{x}_0, \boldsymbol{f}_0^2, \cdots, \boldsymbol{f}_0^M, \cdots, \boldsymbol{x}_N, \boldsymbol{f}_N^1, \cdots, \boldsymbol{f}_N^M\} \tag{18}$$

Please note that in the 2D linear case, one feature $\boldsymbol{f}_0^1$ can define the map coordinate frame.

For the world centric formulation, example results using LS and KF are shown in Fig. 2 and Fig. 3.

The estimated results based on the robot centric formulation using the LS and KF methods are presented in Fig. 10 and Fig. 11 (using the same dataset for Fig. 2 and Fig. 3).

The estimated results based on the map centric formulation using the LS and KF methods are presented in Fig. 12 and Fig. 13 (using the same dataset as that for Fig. 2 and Fig. 3).

Based on the LS and KF results for world centric, robot centric and map centric using the same measurement data are presented in Fig. 2, Fig. 3, Fig. 10 to Fig. 13. Through numerical checking, all the LS results are the same (can be transferred from one to another). The LS results are also the same as the KF results (for the latest time step when all the information is used).

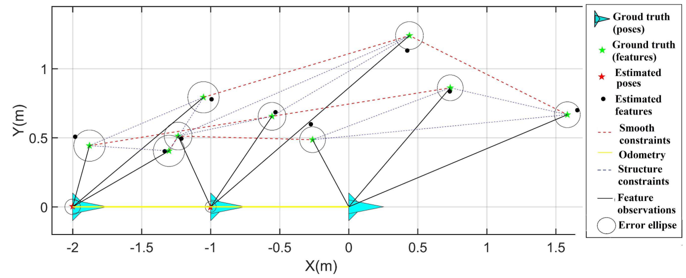


Fig. 10. Estimated result based on robot centric formulation using LS at last step (robot pose 2 is the origin).

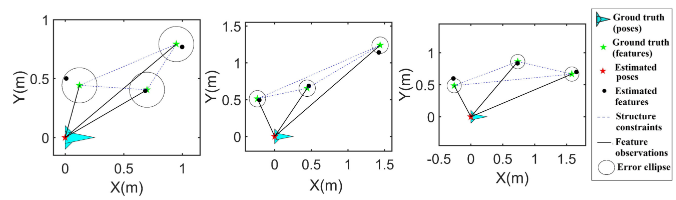


Fig. 11. Estimated result based on robot centric formulation using KF at every step.

For the nonlinear cases, in optimisation based SLAM, the different SLAM formulations are expected to produce exactly the same results, meaning the estimates (and their corresponding covariance matrices) can be transferred from one to another using coordinate transformations. However, for EKF based SLAM, the results are expected to be significantly different.

Research question 4: Are there any advantages using robot centric or map centric problem formulations for SLAM in deformable environments? Can these problem formulations result in better estimate consistency especially for filter based online SLAM? Is it possible to develop invariant EKF SLAM algorithm similar to [27]?

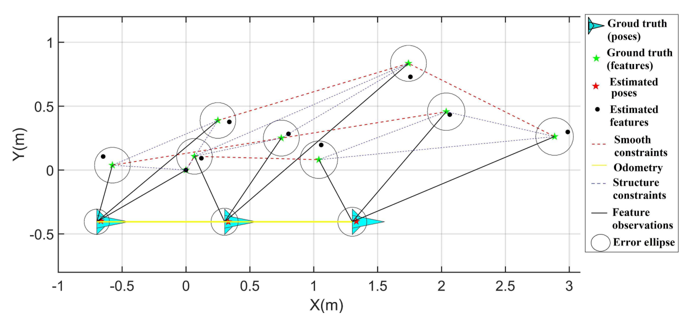


Fig. 12. Estimated result based on map centric formulation using LS (the feature 1 at time step 0 is used to define the coordinate frame).

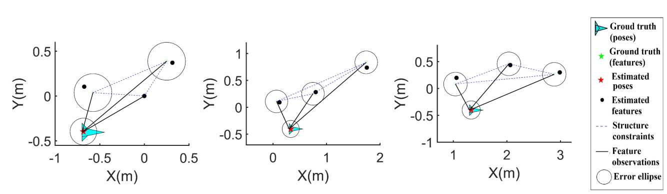


Fig. 13. Estimated result based on map centric formulation using KF at every step.

V. ACTIVE SLAM

Similar to active SLAM in static environments [31], active SLAM in deformable environments means to plan the robot trajectory such that the quality of the SLAM estimate can be improved. In another word, the robot odometry is not pre-determined as in (4), instead, the robot decides its control action online based on the different situations.

A. Active SLAM problem

In active SLAM, one of the most important performance criteria is the quality of SLAM estimate. This can be decided by a metric of the covariance matrix in the obtained SLAM result, for example, the trace/determinant of the covariance matrix.

For deformable environment, the robot trajectory is very critical for achieving high quality SLAM estimate. Intuitively, the robot wants to observe as many moving features as possible at every time steps such that it can gain more information through the observations to features.

In active SLAM, there are some other performance requirements such as coverage or exploration within minimal time [32]. Keeping a good balance between performing other tasks and minimising the uncertainty in SLAM is an important challenge.

B. Example: 2D nonlinear case

A simple example is given to show the advantages of active SLAM over SLAM with a predetermined path. As shown in Fig. 14, the three features are deforming and at the same time moving along a circle-like path. Fig. 14(a) shows the result of SLAM with a pre-determined circular robot path. Since the features are not following the circular path exactly, sometimes the robot cannot observe some features (the total number of unseen features is 47) and the uncertainty of the SLAM estimate is large. On the other hand, as shown in Fig. 14(b), the robot in active SLAM follows the features very closely and can observe all the three features at almost all the steps (there is only one time with one feature unobserved). The uncertainty of the SLAM estimate is small.

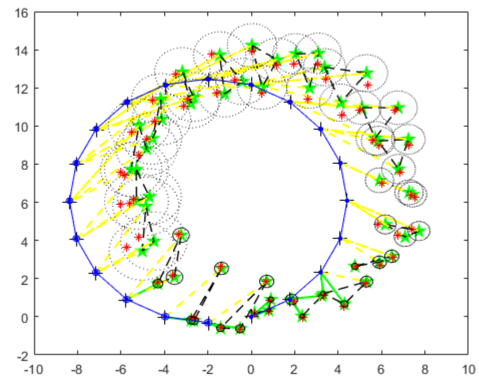
Research question 5: How can we plan the robot trajectory such that better estimation results can be achieved for SLAM in deformable environments? Can we plan the robot trajectory to optimise the observability of the SLAM problem and improve the consistency of the SLAM results?

VI. SOME OTHER RESEARCH QUESTIONS

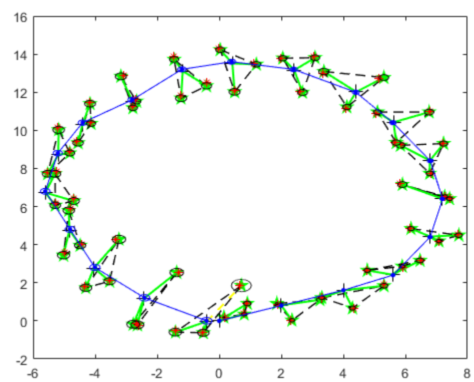
In this paper, we have mainly focused on the back-end of SLAM problem in deformable environments with point features. There are many other important research questions that have not been discussed yet. We will briefly mention some of them in this section.

A. Data association

Data association is a very important aspect in SLAM. The process of extracting the point features, tracking them, and getting rid of outliers are all very important and challenging topics for SLAM in deformable environments. A method of simultaneous estimating the correspondences and non-rigid structure is proposed in [33].



(a) SLAM with a predetermined circular robot path. Number of unseen features is 47. The uncertainty of the estimate of some features is very large.



(b) Active SLAM. Number of unseen features is 1. The uncertainty of the feature estimate is very small.

Fig. 14. Comparing active SLAM with SLAM using a predetermined robot path. Blue cross are robot positions and green star are feature positions. The three features are deforming and moving along a near circular path.

B. Point features are not available

In the case when point features are not available or very difficult to be detected and matched, non-rigid scan matching techniques have been proposed to deal with the SLAM problem in deformable environments [34]. In this case, one challenge is to distinguish whether the scan difference is from the change of poses or from the deformation of the surface. Clearly finding the overlapping area between two scans correctly is also very difficult.

C. Learn the deformation model

A key requirement for solving SLAM in deformable SLAM is to understand the model of the deformation, which is part of the dynamics of features. Finite element analysis has been proposed to characterise the mechanical model of the deformation [18][35]. Machine learning approaches can also be used to learn the model of deformation [36].

VII. CONCLUSIONS

In this paper, we present a few research questions for SLAM in deformable environments when point features are used to represent the map. The research questions are related to problem formulation, observability, consistency, different map representations, and active SLAM. The results of simple 2D examples are used to illustrate the questions and some of the answers. It is demonstrated that for the 2D linear cases, both the different versions of Kalman filter based SLAM algorithms and the different versions of linear least squares SLAM algorithms provide exactly the same optimal results. The MATLAB codes of the results presented in this paper are made publicly available.

The research questions for 3D nonlinear cases are much more complicated and require further investigation. These research questions together with some other challenging questions associated with SLAM in deformable environments are the topics of our future research work.

REFERENCES

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J.J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," IEEE Transactions on Robotics, vol. 32, no. 6, pp.1309-1332, Dec. 2016.

[2] R.A. Newcombe, D. Fox, and S.M. Seitz, "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Jul. 2015, pp. 343-352.

[3] W. Gao and R. Tedrake, "SurfelWarp: efficient non-volumetric single view dynamic reconstruction," Robotics: Science and System, 2018.

[4] L. Kavan, S. Collins, J. Žára, and C. O'Sullivan, "Geometric skinning with approximate dual quaternion blending," ACM Transactions on Graphics, vol. 27, no. 4, pp. 1-23, Oct. 2008.

[5] M. Innmann, M. Zollhöfer, M. Nießner, C. Theobalt, and M. Stamminger, "Volumedeform: Real-time volumetric non-rigid reconstruction," European Conference on Computer Vision, Springer, Cham, Oct. 2016, pp. 362-379.

[6] O. Sorkine and M. Alexa, "As-rigid-as-possible surface modeling," Symposium on Geometry processing, 4, pp. 109-116, Jul. 2007.

[7] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S.R. Fanello, A. Kowdle, S.O. Escolano, C. Rhemann, D. Kim, J. Taylor, and P. Kohli, "Fusion4d: Real-time performance capture of challenging scenes," ACM Transactions on Graphics, vol. 35, no. 4, pp. 1-13, Jul. 2016.

[8] R.W. Sumner, J. Schmid, and M. Pauly, "Embedded deformation for shape manipulation," ACM SIGGRAPH, 2007, pp. 80-es.

[9] J. Song, J. Wang, L. Zhao, S. Huang, and G. Dissanayake, "MIS-SLAM: Real-time large-scale dense deformable SLAM system in minimal invasive surgery based on heterogeneous computing," IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 4068-4075, 2018.

[10] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147-1163, 2017.

[11] J. Lamarca, S. Parashar, A. Bartoli, and J.M.M. Montiel, "DefSLAM: Tracking and mapping of deforming scenes from monocular sequences," IEEE Transactions on Robotics, vol. 37, no. 1, pp. 291-303, Sep. 2020.

[12] P. Ji, H. Li, Y. Dai, and I. Reid, " Maximizing rigidity revisited: A convex programming approach for generic 3D shape reconstruction from multiple perspective views," IEEE International Conference on Computer Vision, Venice, 2017, pp. 929-937.

[13] A. Chhatkuli, D. Pizarro, T. Collins, and A. Bartoli, "Inextensible non-rigid structure-from-motion by second-order cone programming," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 10, pp. 2428-2441, Oct. 2018.

[14] S. Parashar, D. Pizarro, and A. Bartoli, "Local deformable 3D reconstruction with Cartan's connections," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 12, Jun. 2019.

[15] A. Agudo, J. M. M. Montiel, L. Agapito, and B. Calvo, "Online dense non-rigid 3D shape and camera motion recovery," British Machine Vision Conference, Sep. 2014.

[16] S. Parashar, M. Salzmann, and P. Fua, "Local non-rigid structure-from-motion from locally diffeomorphic mappings," Computer Vision and Pattern Recognition, Seattle, USA, Jun. 2020, pp. 2059-2067.

[17] S. Kumar, Y. Dai, and H. Li, "Superpixel soup: monocular dense 3D reconstruction of a complex dynamic scene," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 5, pp. 1705-1717, May 2021.

[18] A. Agudo, "Total estimation from RGB video: on-line camera self-calibration, non-rigid shape and motion," IEEE International Conference on Pattern Recognition, Jan. 2020, pp. 8140-8147.

[19] D. Recasens, J. Lamarca, J. M. Facil, J. M. M. Montiel, and J. Civera, "Endo-Depth-and-Motion: Reconstruction and tracking in endoscopic videos using depth networks and photometric constraints," IEEE Robotics and Automation Letters, 2021.

[20] V. Sidhu, E. Tretschk, V. Golyanik, A. Agudo,and C. Theobalt, "Neural dense non-rigid structure from motion with latent space constraints," European Conference on Computer Vision, Aug. 2020, pp. 204-222.

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

[22] L. Carlone, R. Tron, K. Daniilidis and F. Dellaert, "Initialization techniques for 3D SLAM: a survey on rotation estimation and its use in pose graph optimization," IEEE International Conference on Robotics and Automation, 2015, pp. 4597-4604.

[23] Y. Chen, S. Huang, L. Zhao and G. Dissanayake, "Cramér-Rao bounds and optimal design metrics for pose-graph SLAM," IEEE Transactions on Robotics, vol. 37, no. 2, pp. 627-641, Jan. 2020.

[24] Y. Dai, H. Li, and M. He, "A simple prior-free method for non-rigid structure-from-motion factorization," International Journal of Computer Vision, vol. 107, no. 2, pp. 101-122, Apr. 2014.

[25] Y. Bar-Shalom, X.R. Li and T. Kirubarajan, Estimation with applications to tracking and navigation: theory algorithms and software. John Wiley & Sons. 2004.

[26] S. Huang and G. Dissanayake, "Convergence and consistency analysis for extended Kalman filter based SLAM," IEEE Transactions on Robotics, vol. 23, no. 5, pp. 1036-1049, Oct. 2007.

[27] T. Zhang, K. Wu, J. Song, S. Huang, and G. Dissanayake, "Convergence and consistency analysis for a 3-D Invariant-EKF SLAM," IEEE Robotics and Automation Letters, vol. 2, no. 2, pp.733-740, Jan. 2017.

[28] J. A. Castellanos, J. Neira, and J. D. Tards. "Limits to the consistency of EKF-based SLAM," IFAC Proceedings Volumes, vol. 37, no. 8, pp. 716-721, Jul. 2004.

[29] Z. Wang, S. Huang, and G. Dissanayake, "D-SLAM: A decoupled solution to simultaneous localization and mapping," The International Journal of Robotics Research, vol. 26, no. 2, pp. 187-204, Feb. 2007.

[30] L. Zhao, S. Huang, and G. Dissanayake, "Linear SLAM: Linearising the SLAM problems using submap joining," Automatica, vol. 100, pp. 231-246, Feb. 2019.

[31] C. Leung, S. Huang, and G. Dissanayake, "Active SLAM using model predictive control and attractor based exploration," IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct. 2006, pp. 5026-5031.

[32] Y. Chen, S. Huang, and R. Fitch, "Active SLAM for mobile robots with area coverage and obstacle avoidance," IEEE/ASME Transactions on Mechatronics, vol. 25, no. 3, pp. 1182-1192, Jun. 2020.

[33] Y. Dai and H. Li, "Simultaneous correspondences estimation and non-rigid structure reconstruction," International Conference on Digital Image Computing: Techniques and Applications, Gold Coast, QLD, 2016, pp. 1-7.

[34] S. Kumar, L. Van Gool, C.E. de Oliveira, A. Cherian, Y. Dai and H. Li, Dense Non-Rigid structure from motion: A manifold viewpoint. arXiv preprint arXiv:2006.09197, 2020.

[35] A. Agudo, F. Moreno-Noguer, B. Calvo, and J.M.M. Montiel, "Sequential non-rigid structure from motion using physical priors," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 5, pp. 979-994, Aug. 2015.

[36] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "3d convolutional neural networks for efficient and robust hand pose estimation from single depth images," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1991-2000.