José Lamarca Peiró

Monocular slam for deformable scenarios.

Director/es

Dr. D. José María Martínez Montiel

ISSN 2254-7606

Prensas de la Universidad
Universidad Zaragoza

Servicio de Publicaciones

ISSN 2254-7606

Universidad Zaragoza

1542

Tesis Doctoral

MONOCULAR SLAM FOR DEFORMABLE SCENARIOS.

Autor

José Lamarca Peiró

Director/es

Dr. D. José María Martínez Montiel

UNIVERSIDAD DE ZARAGOZA
Escuela de Doctorado

2021

Tesis Doctoral

Monocular SLAM for Deformable Scenarios

Autor

José Lamarca Peiro

Director/es

José María Martínez Montiel

Escuela de Ingeniería y Arquitectura
2021

Monocular SLAM for Deformable Scenarios

José Lamarca Peiro

Advisor: José María Martinez Montiel

Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza

This dissertation is submitted for the degree of
Doctor of Philosophy

October 2021

A Laura,
mis padres,
mi hermana,
y mis abuelos,

Acknowledgements

These four years have been probably the most challenging and exciting of my life and I would like to acknowledge all the people that have supported me during these four years. Especially to my beloved wife Laura who has put up with me all these years, thinking out loud about weird ideas, all my stayings far from home, and all the writing of my successful and unsuccessful articles and thesis. I would also like to thank my sister, parents and grandparents, who have accepted with unconditional love, all the hours stolen from their time for debugging, deadlines and conferences.

Many people have made the road to this point much easier during these years. But the person who has contributed more intensively has been my great advisor Monti. One page would not be enough to transmit all the gratitude that I have for him. Long hours of writing, discussions, encouragement, concerns and a bunch of patience that are not included in the Professor contract and I deeply appreciate it.

I would like to mention Javier Civera who advised me in my Bachelor and Master thesis introducing me to the SLAM and recommending me this incredible adventure. Also Kirill Safronov and Sarah Gillet who supervised me in my first R&D internship after my Master. Adrien Bartoli who embraced me in his Lab in Clermont-Ferrand. And Jakob Engel who offered me the opportunity of doing an internship in Facebook. A special mention for Jing Dong and Yipu Zhao who supervised my work in the company.

There are a lot more people that I would like to acknowledge for their support. All the people I have collaborated with and have left a part of themselves in my work. Shaifali who introduced me to the incredible world of Riemannian geometry, Mingo whose incredible understanding of SLAM has supposed a breaking point in my research, Morlana who started almost at the same time as me in the fuzzy world of Non-Rigid, Juanjo with whom I have collaborated very closely during this last lockdown year, Recasens with whom we could bring together photogrammetry and deep learning, and Chema, my dear lab mate that has pushed me during this four years and will keep pushing me beyond. I just want to transmit to you my deepest thanks. I would not like to forget all the friends, PhD students and professors who I have not mentioned yet and with whom I have shared ideas and experience: Lorenzo, Clara, Berta, Carlos, Richard, Jon, Iñigo, Jesus, Seong, Alejandro, Melani, Bastien, etc.

All my entire formation from the beginning of my degree to the end of my thesis would not have been possible without the support of the incredible Spanish public education. I have to thank for the grants received during these years. Especially to the public MINECO scholarship BES-2016-078678 for my predoctoral contract. I would like to acknowledge the projects in which I have collaborated during my research: DPI2017-91104-EXP: SLAM visual deformable para endocopia ,PGC2018-096367-B-I00: SLAM visual mediante aprendizaje profundo, EndoMapper / Real-time mapping from endoscopic video (H2020 G.A. no. 863146), GRUPO DGA and DPI2015-67275-P: READY-to-TRANSFER VISUAL SLAM. I could access my dreamed education and I hope that many excellent students could still do it in the upcoming years.

Finally, thanks to the external reviewers Clara and Yongbo and to the PhD committee Shoudong, Lourdes, Chepe, Daniel and Francesc for your time.

"The Sleep of Reason Produces Monsters." – Francisco Goya

Abstract

The problem of localizing the position of a sensor in an uncertain map which is estimated simultaneously is known as Simultaneous Localization and Mapping –SLAM–. It is a challenging problem comparable to the egg and chicken paradigm. To locate the sensor we need to know the map, but to build the map, we require the position of the sensor. When using a visual sensor, e.g. a camera, it is coined as Visual SLAM or VSLAM. Visual sensors for SLAM are divided between those which provide depth information (e.g. RGB-D cameras or stereo rigs) and those which do not (e.g. monocular cameras or event cameras). In this thesis, we have focused our research on SLAM with monocular cameras.

Due to the lack of depth perception, monocular SLAM is inherently harder compared to the SLAM with depth sensors. State-of-the-art manuscripts in monocular VSLAM systems have widely assumed that the scene remains rigid during the entire sequence, which is a feasible assumption for industrial and human environments. The rigidity assumption constrains the problem and allows to build a reliable map after processing several images. In the last years, the interest in SLAM has arrived at medical areas. SLAM algorithms could help to orientate the surgeon or to locate the position of a robot. However, in contrast, to the industrial or usual human scenarios, in in-body sequences, everything can deform eventually and the rigidity assumption rends invalid in practice, and therefore the state-of-the-art monocular SLAM algorithms do too. Thus, we aim to extend the boundaries of SLAM algorithms and to conceive the first monocular SLAM system able to cope with the deformation of the scene.

The state-of-the-art SLAM methods in literature compute the position of the camera and the map of the scene in two concurrent threads: the tracking and the mapping. The tracking processes every single frame to locate the sensor continuously. In contrast, the mapping is in charge of building a map of the scene. We have adopted this structure and conceive both the deformable tracking and the deformable mapping now able to cope with the deformation.

Our first contribution is the deformable tracking. The deformable tracking uses the structure of the map to recover the camera pose from a single view. Simultaneously, as the map is

deforming during the sequence, it also recovers the deformation of the map for every frame. We have proposed two families of deformable tracking. In the first kind of deformable tracking, we assume that all the points are embedded in a surface referred to as the template. We can recover the deformation of the surface thanks to a global deformation model that allows us to estimate the most likely deformation of the object. With our second kind of deformable tracking, we prove that it is possible to recover the deformation of the map without a global deformation model, representing the map as individual surfels. Our experimental results showed both methods outperform both in robustness and accuracy to the previous rigid methods recovering the deformation of the map.

Our second contribution is the conception of the deformable mapping. It is the back-end of the SLAM algorithm and processes a batch of frames to both recover the structure of the map for each of these frames and to grow the map by assembling the partial observations of the same. Both deformable tracking and mapping running in parallel and together assemble the first deformable monocular SLAM: *DefSLAM*. An extended evaluation of our method proved, in both laboratory-controlled sequences and medical sequences, that our method successfully processes sequences where current monocular SLAM systems fail.

As our third contribution are two methods to exploit the photometric information in deformable monocular SLAM. On the one hand, *SD-DefSLAM*, which exploits the semi-direct matching to obtain a much more reliable tracking of the map points in the new frames. As a consequence, it was proved to be more robust and stable in medical sequences. On the other hand, we propose the *Direct and Sparse Deformable Tracking* in which we use a direct photometric error to track the deformation of a map modelled as a set of unconnected 3D surfels. We can recover the deformation of multiple disconnected surfaces, non-isometric deformations and surfaces with changing topology.

Resumen

El problema de localizar la posición de un sensor en un mapa incierto que se estima simultáneamente se conoce como Localización y Mapeo Simultáneo –SLAM–. Es un problema desafiante comparable al paradigma del huevo y la gallina. Para ubicar el sensor necesitamos conocer el mapa, pero para construir el mapa, necesitamos la posición del sensor. Cuando se utiliza un sensor visual, por ejemplo, una cámara, se denomina Visual SLAM o VSLAM. Los sensores visuales para SLAM se dividen entre los que proporcionan información de profundidad (por ejemplo, cámaras RGB-D o equipos estéreo) y los que no (por ejemplo, cámaras monoculares o cámaras de eventos). En esta tesis hemos centrado nuestra investigación en SLAM con cámaras monoculares.

Debido a la falta de percepción de profundidad, el SLAM monocular es intrínsecamente más duro en comparación con el SLAM con sensores de profundidad. Los trabajos estado del arte en VSLAM monocular han asumido normalmente que la escena permanece rígida durante toda la secuencia, lo que es una suposición factible para entornos industriales y urbanos. El supuesto de rigidez aporta las restricciones suficientes al problema y permite construir un mapa fiable tras procesar varias imágenes. En los últimos años, el interés por el SLAM ha llegado a las áreas médicas. Los algoritmos SLAM podrían ayudar a orientar al cirujano o localizar la posición de un robot. Sin embargo, a diferencia de los escenarios industriales o urbanos, en secuencias dentro del cuerpo, todo puede deformarse eventualmente y la suposición de rigidez acaba siendo inválida en la práctica, y por extensión, también los algoritmos de SLAM monoculares. Por lo tanto, nuestro objetivo es ampliar los límites de los algoritmos de SLAM y concebir el primer sistema SLAM monocular capaz de hacer frente a la deformación de la escena.

Los sistemas de SLAM actuales calculan la posición de la cámara y la estructura del mapa en dos subprocesos concurrentes: la localización y el mapeo. La localización se encarga de procesar cada imagen para ubicar el sensor de forma continua, en cambio el mapeo se encarga de construir el mapa de la escena. Nosotros hemos adoptado esta estructura y concebimos tanto la localización deformable como el mapeo deformable ahora capaces de recuperar la

escena incluso con deformación.

Nuestra primera contribución es la localización deformable. La localización deformable utiliza la estructura del mapa para recuperar la pose de la cámara con una única imagen. Simultáneamente, a medida que el mapa se deforma durante la secuencia, también recupera la deformación del mapa para cada fotograma. Hemos propuesto dos familias de localización deformable. En el primer algoritmo de localización deformable, asumimos que todos los puntos están embebidos en una superficie denominada plantilla. Podemos recuperar la deformación de la superficie gracias a un modelo de deformación global que permite estimar la deformación más probable del objeto. Con nuestro segundo algoritmo de localización deformable, demostramos que es posible recuperar la deformación del mapa sin un modelo de deformación global, representando el mapa como surfels individuales. Nuestros resultados experimentales mostraron que, recuperando la deformación del mapa, ambos métodos superan tanto en robustez como en precisión a los métodos rígidos.

Nuestra segunda contribución es la concepción del mapeo deformable. Es el back-end del algoritmo SLAM y procesa un lote de imagenes para recuperar la estructura del mapa para todas las imagenes y hacer crecer el mapa ensamblando las observaciones parciales del mismo. Tanto la localización deformable como el mapeo que se ejecutan en paralelo y juntos ensamblan el primer SLAM monocular deformable: *DefSLAM*. Una evaluación ampliada de nuestro método demostró, tanto en secuencias controladas por laboratorio como en secuencias médicas, que nuestro método procesa con éxito secuencias en las que falla el sistema monocular SLAM actual.

Nuestra tercera contribución son dos métodos para explotar la información fotométrica en SLAM monocular deformable. Por un lado, *SD-DefSLAM* que aprovecha el emparejamiento semi-directo para obtener un emparejamiento mucho más fiable de los puntos del mapa en las nuevas imágenes, como consecuencia, se demostró que es más robusto y estable en secuencias médicas. Por otro lado, proponemos un método de *Localización Deformable Directa y Dispersa* en el que usamos un error fotométrico directo para rastrear la deformación de un mapa modelado como un conjunto de surfels 3D desconectados. Podemos recuperar la deformación de múltiples superficies desconectadas, deformaciones no isométricas o superficies con una topología cambiante.

Table of contents

Chapter 1

Introduction

1.1 Problem statement

In the cave allegory, exposed by Plato in "The Republic" in 400 B.C., humans were only able to observe shadows of the reality in a wall. With a computer, this allegory gets literal since only with the input of a sensor, e.g. a camera, the computer has only partial and incomplete observations of the scene that surrounds it. From this information, the computer must be able to recover as much information about the environment and itself as possible. There is a huge range of different information to extract with a computer, e.g. we may want to know what are the objects in a room, or maybe if it is going to rain by looking at the clouds. In this dissertation, I am going to focus on the understanding of *”where the sensor is located”* and *”how the observed environment is”* and what is the more important contribution of this thesis *”how the environment evolves”*.

Solving *”where the sensor is located”* and *”how the observed environment is”* simultaneously and sequentially is widely known as SLAM, the acronym for *Simultaneous Localization and Mapping* problem. It is the perfect example of the chicken-and-egg paradigm: we need the structure of the scene to locate our sensor and we need the position of the sensors to map the scene from its partial observations. If the given sensor works through with visual perception (e.g. standard monocular cameras, stereo pairs, or event cameras to name the more frequent), the problem is so-called Visual SLAM. From now on, the structure of the scene will be referred to as *map*.

As an example of a SLAM algorithm, given the input of an RGB-D camera that includes a depth image apart from the normal RGB image, we initialize the map as the point cloud extracted from the first image. Given this map, we can use it to locate the sensor in the following images by fitting the prediction of what we should be seeing and the measurements of what we are actually seeing. After processing some images, we can refine and grow

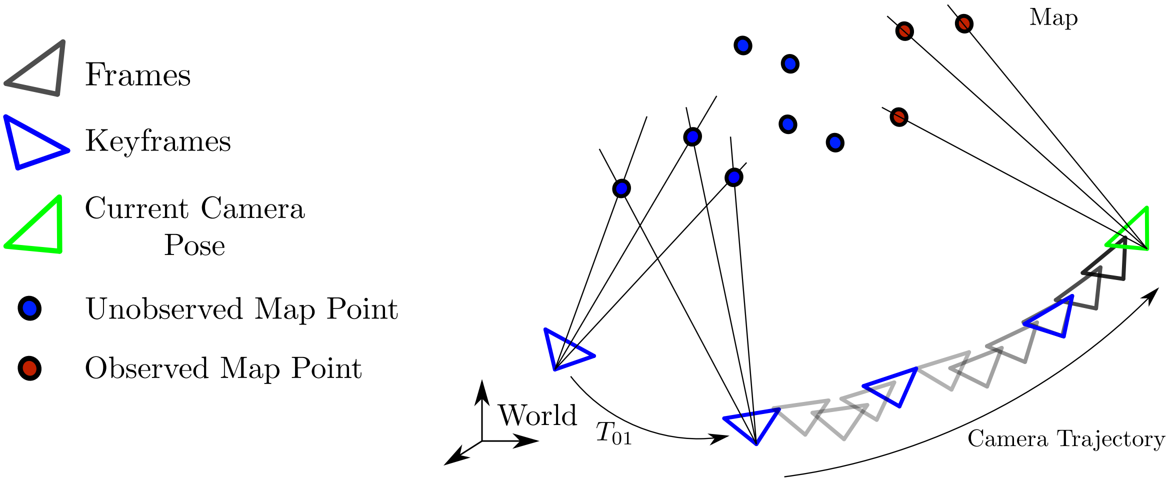

Fig. 1.1 Monocular SLAM scheme. The map is initialized from scratch with the first two frames. The tracking processes the frame to estimate the current camera pose. The mapping processes some frames, so-called keyframes, to create a map. The map accuracy gets better as we introduce keyframes with more parallax. Eventually, the camera abandons the initial zone exploring new areas. We distinguish between those map points observed by the camera (in red) and those unobserved (in blue).

our intial map with the new RGB-D images since we know the pose of the camera, like assembling a puzzle. With a refined and bigger map, we can keep performing these two steps to track our sensor and build the entire map of the scene from partial observations. The step of localizing the camera is known as *tracking*, meanwhile, the step of building the map is known as *mapping*.

If we cannot observe the depth, like it is the case for monocular cameras, the algorithm becomes more challenging. We need the position of at least two cameras to initialize the map that will be used to track the camera (See Fig.1.1). Assuming that the first camera is the reference camera, the algorithm estimates an initial solution of the pose of the second camera from scratch. Then, we use the pose of both cameras to roughly recover the map points by triangulation. This initial map is used to track the sensor position in the next images, so-called frames. Eventually, we process a batch of some of these frames, so-called keyframes, to improve the initial map. As a thumb rule, the more parallax the keyframes have, the better the reconstruction is. Finally, with enough parallax, we recover a map that we can use to reliably track the sensor. Needless to say, the lack of depth information in the monocular case entangles substantially the problem. However, trying to solve the SLAM problem with monocular cameras is not in vain. The power consumption is much smaller, it is easier to implement in hardware and the miniaturization is almost unlimited.

Due to its advantages and challenging nature, monocular SLAM problem has drawn the attention of many good researchers, and incredible advances were made in this area in the last decades. Almost all the state-of-the-art monocular SLAM algorithms use the scheme presented in the last paragraph with a tracking and mapping concurrent threads. Notice, however, that the current versions of the described components can be used only with the underlying assumption of a rigid world. That means that the scene must remain rigid during the entire sequence, which is a valid assumption for human scenarios like buildings or the distribution of a room or a factory. Thanks to the rigidity assumption, multiview geometry provides enough equations to overconstrain the problem and to compute the map from matches among keyframes, and also the camera pose from matches between the frame and the map. That is why in the rigid case, we deal with a well-posed problem.

The astonishing advances of the last years both in computer vision applications and SLAM brought the interest in many different areas like medical imaging or computer-assisted surgery. In in-body scenes where surgeons can barely orient themselves after a long training, the computers have been shown to be extremely helpful. However, in contrast to human scenarios, one of the main characteristics of the in-body sequences is that everything can deform eventually. This is an important limitation for the current monocular SLAM systems since the rigidity assumption is not longer valid for this kind of scenario. Thus, multiview geometry does not provide enough equations rendering the problem ill-posed. As a consequence, instead of basing our SLAM algorithms in multiview geometry, we use Non-Rigid Structure-from-Motion –NRSfM– to provide the theoretical fundamentals for our algorithms. And this is the goal of this thesis: *Conceive a monocular SLAM system able to work in scenarios where the structure of the scene can deform built on top of NRSfM techniques*.

We aim to expand the boundaries of monocular SLAM to enable the processing of deformable scenes. As we mentioned before, we have two main functions in a SLAM algorithm: the tracking, which takes care of recover the pose of the camera given a known map, and the mapping, in charge of creating that map. Through this manuscript, we present a tracking and a mapping modules able to process deformation. We coined those modules as *deformable tracking* and *deformable mapping*.

The tracking is the front-end of the SLAM algorithms. It takes as input the stream of images and processes the current image to estimate the state of the system. In rigid SLAM, this state is the camera pose since the state of the map is constant. For the deformable case, the state for each frame is the camera pose and the deformation of the map. The camera pose is modelled with six DoF –Degrees of Freedom–, three for the translation of the camera and three DoF for the rotation. When it comes to the map, we need to choose a representation of

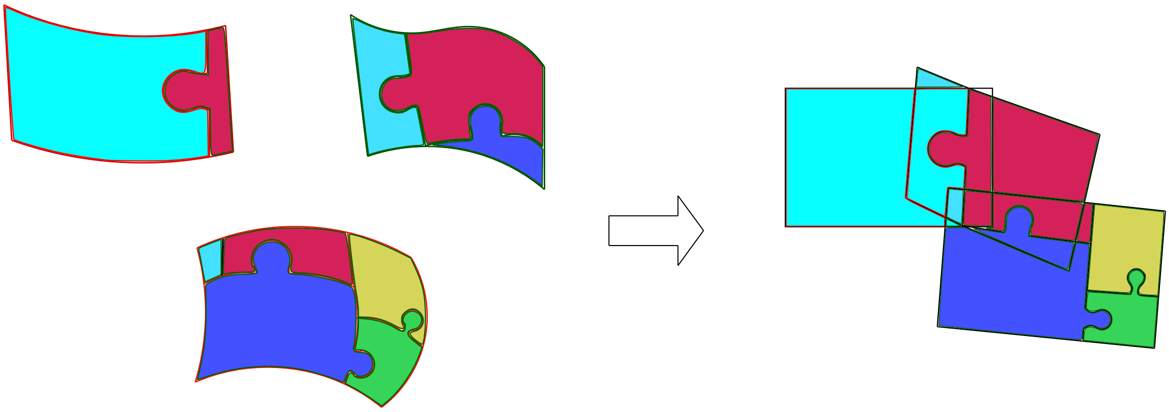

Fig. 1.2 Non-Rigid Surface alignment. It deals with the alignment of parts of the same surface with different deformations. In deformable SLAM, it is a key part of the deformable mapping and deals with the alignment of different partial maps observed into a single global map

the selected scene. It can be reconstructed as a volumetric map or a surface. There are few objects that are actually a surface, however, the surface of the object is usually the observable part of the scene. Thus, we represent the map as a surface. In the Chapters 2, 3 and 4 we represent the map as a mesh, where the estimated map state is the position of the nodes. The mesh embeds all the map points and is used to incorporate a deformation model with standard equations from computer graphics and continuum mechanics. In Chapter 5, we model the surface as disconnected surfels where the map state is the position and orientation of these surfels.

In addition to recovering the camera pose and the deformation of the map, the tracking thread performs the automatic data association between frames. There are two main ways to perform the data association between frames or between the frame and the map. The first one is the feature-based method and consists in matching interest points through an extractor and a descriptor. The second consists in optimizing the photometric error, i.e. the difference between the intensity level of the observation and the prediction. In the deformable tracking proposed in Chapter 2 and 3, we use a feature-based tracking with ORB matching. We find this technique fast enough to develop a real-time SLAM and very reliable for well-textured scenes. In Chapter 4 and Chapter 5, we show the potential of the photometric error compared with the features methods yielding to more robust and accurate methods of tracking.

The *deformable mapping* estimates the structure of the map for the deformable tracking. There are two inherent tasks for the deformable mapping. The first task is to reconstruct a deforming scene from several monocular views. As aforementioned, when we relax the rigidity assumption the problem becomes underconstrained. In our deformable mapping proposed in Chapter 3, we build this part on top of the NRSfM methods that reconstruct the structure of the map from a stream of monocular images. We achieve to reconstruct the map observed in several keyframes by assuming that the map is a smooth surface, locally isometric and infinitesimally planar.

The second task of the deformable mapping is to grow the map when visiting new zones of the scene. Once we have obtained the structure of the scene observed for each keyframe, we need to align partial observations of the map. In contrast to the rigid case, we need to align partial observations of the map with a different deformation for each keyframe (See Fig. 1.2). The deformable mapping proposed in Chapter 3 estimates incrementally the map of the scene surface for each keyframe processed and then performs a *non-rigid surface alignment* of the observed partial maps in the different keyframes into a bigger map.

Thanks to the union of the deformable tracking and deformable mapping, we conceive the first monocular SLAM system able to perform exploration in deformable environments: *DefSLAM*.

In the next Sections, we contextualize our work *wrt.* the state-of-the-art SLAM systems and NRSfM methods for monocular sequences. In the last section, we state the contributions to the field made within this thesis.

1.2 Visual SLAM

SLAM comprises the reconstruction of a map and the localization of a sensor for each new measurement. The main goal of SLAM is to bring the localization of the sensor in scenarios where there is no a prior map available and it needs to be built. The more important characteristic of a SLAM algorithm is the assumption of sequentiality. SLAM algorithms do not process an unconnected set of images, they process videos, where each image is close to the previous one in time and can be related more easily. This is a natural defense against wrong data association and perceptual aliasing in similar scenes. Another important characteristic inherited from this assumption is the efficiency of the system. A solution for the last image will be always close enough to the solution for the new image.

The first real-time monocular SLAM systems (Chiuso et al., 2002, Davison, 2003), were based in EKF filtering. They evolved to more complete and better engineered EKF systems (Davison et al., 2007, Civera et al., 2008, Eade and Drummond, 2006). In MonoSLAM (Davison, 2003, Davison et al., 2007), the algorithm matched a set of specific features between images, so-called keypoints, to recover the map geometry as a point cloud that represents the 3D location of the keypoints. Every new frame was processed by the filter to estimate the map point locations and the camera pose. The depth of the scene is unknown for a single view. The structure of the map only becomes observable by using several images with a certain separation, so-called parallax. The main drawback of these seminal methods is that they process every single frame for building the map from viewpoints with very small parallax, with a computational cost quadratic in the map size what severely limits its scalability. Additionally, the lack of relinearization degrades its performance in the long run.

Klein and Murray (2007) presented the Parallel Tracking And Mapping –PTAM– structure, that solves the low parallax problem by splitting of the mapping and camera tracking. The camera tracking is estimated at frame-rate, meanwhile, the map is estimated in parallel only for a few selected frames, coined as keyframes, with higher parallax. This kind of structure is especially useful to decouple the camera tracking, that can run in real-time, and the mapping built on top of more accurate techniques like bundle adjustment (Triggs et al., 1999) instead of filtering (Strasdat et al., 2012). Due to its advantages, the PTAM structure has become dominant in the last years and it is used by the current state-of-the-art VSLAM system (Engel et al., 2014, 2017, Mur-Artal et al., 2015, Campos et al., 2021).

The early SLAM methods were feature-based methods (Davison, 2003, Klein and Murray, 2007). The feature-based matching algorithms have two steps detection and description. In the detection, the aforementioned keypoints are selected. In the second step, each one of these keypoints is associated with a repeatable code, so it can be found in other images. Ideally, both these points and their descriptor are as-invariant-as-possible so they can be tracked from different points of view. Among all the possible options of features that can be used for matching for SLAM, the selected one should be fast enough so it does not increase too much the runtime of the algorithm. This excludes the analytic methods like SIFT, SURF or AKAZE (Alcantarilla et al., 2011). Binary extractors and descriptors are used preferentially. ORB-SLAM (Mur-Artal et al., 2015) and its extended versions Campos et al. (2021) use a FAST-ORB tandem (Rublee et al., 2011) for all the stages of the algorithm what makes it extremely efficient.

There is another family of SLAM systems which use the photometric error for the matching, they are the so-called direct methods. Instead of matching keypoints between images based on the descriptor similarity, they minimize the photometric error (gray level) between the estimated projection of the point and the current image. The dense direct method par excellence is DTAM (Newcombe et al., 2011). They use variational optimization to minimize the gray difference between images to recover the map and use it to track the camera. The main drawback of this method is the high computational cost. Thanks to the variational formulation, the computation can be parallelized in the GPU and it is able to run in real-time. Semi-dense approaches reduce notably the overall cost of the problem by taking into consideration only the pixels of the image with high gradient (Engel et al., 2014).

In (Concha and Civera, 2015), a dense reconstruction is obtained with low computational cost by assuming that the low gradient parts in the images are planes. One of the most remarkable direct method in terms of accuracy is DSO (Engel et al., 2017) with a direct sparse photometric bundle adjustment. It recovers the structure of the scene for a sparse set of points by optimizing directly the photometric error. This work was improved later in DSM (Zubizarreta et al., 2020) with a revisiting policy over the previously observed keyframes. A midpoint is the semi-direct methods like SVO (Forster et al., 2014), where the photometric error is initially used to estimate the matches. Once estimated the matches become fixed and the computation is performed as a feature-based method.

The first SLAM systems were able to deal with small scenes, but they were prompted to fail or have a poor performance for larger scenes. All the SLAM systems have a tendency to accumulate drift in the non-observable variables of the system. In the case of the monocular SLAM they accumulate drift in translation, rotation and scale (See Fig.1.3). In the works by Strasdat et al. (2010, 2011), they focus on getting a SLAM able to work on a large scale. In the first work (Strasdat et al., 2010), the drift is corrected by introducing a loop closure that performs a Sim(3) pose graph optimization. In the next work (Strasdat et al., 2011), a covisibility window is used to center the computation of the algorithm only in a local map instead of the entire map. When the map becomes bigger, global optimization can become ineffective. Some works have focused on how to perform an efficient update of the map through incremental smoothing and mapping (iSAM) (Kaess and Dellaert, 2009), or through novel data structures like Bayesian trees (Kaess et al., 2010).

One essential part to overcome possible tracking failures is the relocalization module. This module is meant to solve the kidnapped robot problem. The algorithm must be able to detect if the sensor observing is in the currently available map and where it is located. There are two parts in this algorithm. The first consists of image retrieval by searching similar images to the new frame among the current map keyframes. It can be done with a bag-of-word (Gálvez-López and Tardos, 2012) that describes the image through a vocabulary of the descriptors or with newer techniques based in deep learning (Arandjelovic et al., 2016). The image retrieval can get extremely hard under illumination changes like day-and-night or season changes sequences, e.g. between summer and winter. The second part of the relocalization consists of, once selected a candidate image, matching between the local map of that image and the query one and recover the pose of the camera, for example with the ePnP algorithm (Lepetit et al., 2009).

All the current state-of-the-art SLAM algorithms are based on the assumption that the map was static from the beginning of the sequence to the end. This assumption can be true for some sequences in which the map contains rigid parts, but when there are some non-static

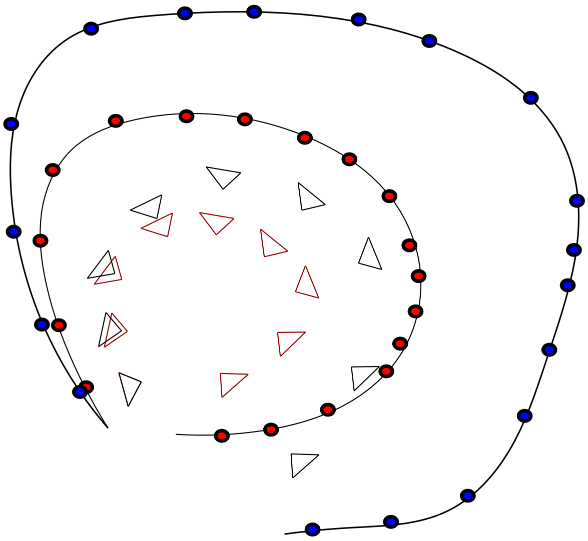

Fig. 1.3 Drift in SLAM. Triangles represent the cameras, dots the map points. Ground truth in red, estimation in blue. The scale error accumulates proportionally to the travelled trajectory. The rotation and translation also accumulate drift in the estimation as they are relative parameters between cameras.

parts the tracking and mapping tend to fail. Although there are mechanisms to weigh less these parts in the optimization, they always affect the quality of the reconstruction to a greater or lesser extent. In the work presented by Bescos et al. (2018), it is proposed to remove these dynamic parts through segmentation to subsequently apply conventional rigid techniques.

There are just a few brave SLAM works that have tackled the challenge of reconstructing the entire scene even if it is deforming. The first SLAM method able to reconstruct a deforming map was done by Newcombe et al. (2015). DynamicFusion was the first RGB-D deformable SLAM developed. They define the dynamics of map with an as-rigid-as-possible physical model (Sorkine and Alexa, 2007). The map is deformed to fit with the current frame depth measurements and the camera is estimated *wrt.* the map shape-at-rest, that they coined the canonical map. Innmann et al. (2016) proposed a similar approach, but using in addition the RGB channel to improve the performance. MaskFusion (Runz et al., 2018) uses a rigid SLAM method to track the camera and DynamicFusion to reconstruct the dynamic parts of the scene segmented and classified by using semantic information. Another important work in this line is MISSLAM (Song et al., 2018), Song et al. (2018) extended the DynamicFusion to work with stereo cameras and it was able to reconstruct maps in in-body scenarios by using the embedded deformation model. DynamicFusion was the work that inspired this thesis: we aim to replicate similar results but with an important limitation, we target to use monocular sequences.

In our first work (Lamarca and Montiel, 2018a), we propose a deformable tracking able to cope with deformations given a known map. This work had a pipeline with two stages: a rigid SLAM to reconstruct the shape-at-rest of the map, and a second stage in which we use the map to track both the camera and the deformation of the map. In DefSLAM Lamarca et al. (2020), we presented the deformable mapping to complement the tracking and conceived the first full monocular Deformable SLAM pipeline. Later in Rodríguez et al. (2020), we propose to change the feature-based method used in DefSLAM by a photometric method, obtaining a more robust and stable matching. Finally, we introduce Lamarca et al. (2021) where we develop a deformable tracking able to cope with local deformations without assuming a global model, being able to track discontinuous surfaces and non-isometric global deformations.

1.3 Non-Rigid Structure-from-Motion

The reconstruction of non-rigid scenes and the location of the poses of the camera from monocular sequences is an underconstrained problem with a large literature in the last decades. To develop the monocular deformable SLAM, we use as starting point the current Non-Rigid monocular techniques. For the deformable tracking, we use the Shape-from-Template –SfT– or Template-based techniques. For the mapping, we based our method on the NRSfM algorithms. For the sake of simplicity, we have split this section in the related literature for the deformable tracking and for the deformable mapping.

1.3.1 Deformable Tracking

The deformable tracking aims to recover the deformation of the map and the camera pose. Given a known camera pose and the initial configuration of the map, the first ambiguity emerges when the scene can be deformed. If the structure of the scene can be deformed, multiple configurations can generate the same image. The simplest example is an elastic body that stretches and moves away, or shrinks and comes closer to the camera. We coined this ambiguity as the *growing map ambiguity*. To solve the ambiguity, we need a deformation model that constrains the map deformation. The family of algorithms that tries to solve this problem are called template-based or Shape-from-Template techniques. The template is the shape-at-rest of the map, also called canonical map. The distinctive characteristics between these algorithms are how to define the original shape (usually a surface) and its deformation model.

One of the widest deformation model used is isometry. When a surface is isometric, the geodesic distance (distance through the surface) between two points is preserved. At an infinitesimal level, an isometric surface preserves the length of the directional derivatives and the angle between them. Bartoli et al. (2012) proposed a closed-form solution assuming isometry. It uses as input a 2D warp between the image, observing the initial shape, and the target image. They can infer the 3D shape for the target image using the 2D warp. Other closed-form methods that assume isometry (Moreno-Noguer et al., 2009, Salzmann et al., 2008) use directly the correspondences between images meaning to solve very large linear systems. Some closed-forms solutions represent the scene as a volumetric deformable object. Parashar et al. (2015) solve the problem by using 3D splines as a continuous approach. In contrast, Collins and Bartoli (2015) use a discrete approach and define the object as a tetrahedral mesh. Latter works have also explored the well-posedness and the uniqueness by measuring the curves that are contained in the surface Gallardo et al. (2020). Two relaxations of the isometry assumption are the conformality, which only preserves the angles (Bartoli et al., 2015) and equireality (Casillas-Perez et al., 2019) that preserves the area between both directional derivative vectors.

The optimization-based methods work directly with the correspondences (Ngo et al., 2016, Salzmann and Fua, 2011, Yu et al., 2015). They jointly optimize the reprojection error and the deformation energy of the template. As it happens in rigid, optimization-based techniques are excellent to exploit the continuity of a sequential image stream. They usually recover the observed shape directly from correspondences between the shape-at-rest and the new image. If the changes in shape are not very extreme, they take the solution in the last image as the seed to estimate the shape in the new image. As aforementioned, the problem needs to be constrained. An important number of methods are based on inextensibility, or in other words, limiting the Euclidean distance between nodes. There is an example of inextensibility in Fig. 1.4. Other additional sources of information can be used such as shading or textural clues (White and Forsyth, 2006, Moreno-Noguer et al., 2010). In Ngo et al. (2016), a physical model based on inextensibility and a Laplacian penalty is used to penalize both the stretching and the bending of a surface and recover its shape in real-time.

Many template-based methods assumed static camera (Ngo et al., 2016, Salzmann and Fua, 2011, Bartoli et al., 2015). However, in SLAM, we are interested in estimating an explicit camera pose, then the camera pose becomes an unknown of the problem. If the map can move freely, the space of solutions includes any coupled rigid motions of the camera and the map that produce the same view. That is what we have coined as *the floating map ambiguity*. There is very small literature about the estimation of the camera pose and the deformation of the object in the monocular case at the same time. One of the most common

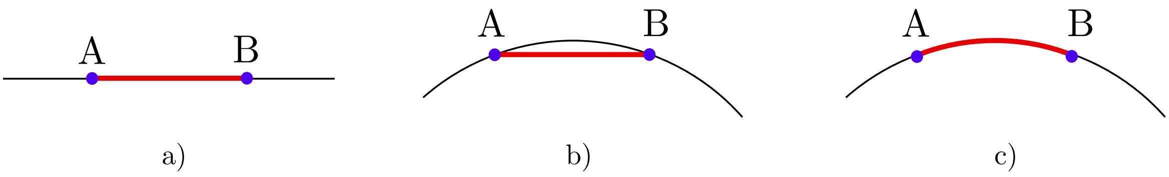

Fig. 1.4 a) Shape-at-rest of the surface b) and c) deformed surfaces. b) shows the Euclidean distance between the point A and B. In contrast, c) shows the geodesic distance between A and B. In an isometric surface, the geometric distance between two points never changes. An inextensible surface is defined using the Euclidean distance. As it can be seen in b), with an isometric deformation the Euclidean distance with deformation is always smaller than the original one.

solutions is to segment some rigid parts and use them to estimate the camera pose. It can be a fair solution for scenes where most observed parts are static, but this is not always the case and it is not the case for our in-body sequences.

Moreno-Noguer and Porta (2011) introduced the factor graph formalism for simultaneous camera localization and shape recovery. Both map and deformation were solved simultaneously by a least-square optimization and the method was able to detect and remove outliers. They used an orthographic camera model that does not allow large displacement of the camera *wrt.* the reconstructed body. It was one of the first work bringing closer the monocular SLAM and the non-rigid methods and can be considered the first non-rigid camera tracking. Another way to constrain the problem is through physical priors. A physical model is used as prior to reconstruct the shape. Agudo et al. (2012) use 3D finite element models –FEM– to formulate the deformations of the scene and the camera pose. In Agudo et al. (2015), an initial reconstruction is made in static to initialize. Then, a FEM model is used to estimate the most plausible solution by estimating the minimal forces required for the observed deformation. In the work by Agudo et al. (2015), the model is defined through FEM and there are rigid boundary conditions, yielding to a shape with limited deformation what constrains the movement of the body and the camera.

We use the idea of a shape-at-rest and a deformation model to build our deformable tracking, the front-end of the deformable SLAM conceived to recover the camera pose and the deformation of the map at frame-rate (Lamarca and Montiel, 2018a). The closed-form template-based techniques can suit perfectly for a non-rigid relocalization module, but they are very sensitive to the matching. In contrast, the optimization-based techniques are clear candidates as camera and map deformation tracking. Indeed, we present two optimization-based deformable tracking methods in this thesis (Chapters 2 and 5). The first one uses a triangular mesh and a global deformation model to obtain the deformation of the map for every single frame. In the second one, we minimize directly the photometric error and a local deformation model to recover the deformation of the map.

1.3.2 Deformable Mapping

If we assume perspective camera, unknown shape-at-rest and camera pose, a third ambiguity arises, we call it the *common scale ambiguity*. With a perspective and moving camera, the scales of two different points in a deformable scene are unrelated. For example, if we have a real elephant very far and a toy elephant very close, they would have the same projection in the camera. If they move conveniently while imaged by a moving camera, they can generate the same projection along their own trajectories. More constraints are needed to reconstruct the global scene, usually quite restrictive. The methods that tackle this problem are called Non-Rigid Structure-from-Motion –NRSfM–.

The first attempt of reconstructing deforming scenes came with the seminal work Bregler et al. (2000) twenty years ago. This method uses an orthographic camera model, assuming that the changes in the observed depth are negligible. The geniality of Bregler et al. (2000) is the introduction of the concept of bases. The basic idea is that a deforming shape can be defined as a combination of shapes, or in other words, a low-dimensional shape space. In the case of the work presented by Bregler et al. (2000), a linear combination of a certain number of shapes.

In the last twenty years many orthographic methods have been developed based on this idea, and they are known as statistic methods or low-dimensional NRSfM. The main distinction between methods is the representation of the basis. Initially, all the methods followed the idea of recovering a structure (Torresani et al., 2004, Xiao et al., 2004, Dai et al., 2014, Torresani et al., 2008). Bartoli et al. (2008) propose a bundle adjustment based on the idea of basis to recover the minimal number of shapes. In Akhter et al. (2011), they propose a change of paradigm and they use Discrete Cosine Transform (DCT) bases to define the trajectory of the points. They prove the duality between using a shape basis space and a temporal trajectory basis space. Although the results were impressive, almost all the images had to be processed and the system failed under certain camera movements. Gotardo and Martinez (2011b) expanded this idea by explicitly combining the DCT and shape space in a complementary rank-3 space. Finally, Gotardo and Martinez (2011a) proposed non-linear combinations of the basis instead of purely linear combinations what improved the results against the former proposals. Garg et al. (2013) proposed an optimization that regularizes the structure of the scene to be smooth and the number of basis.

As aforementioned, all these methods can work thanks to the assumption of orthographic camera. This assumption is quite acceptable for many situations where the depth of the camera is almost constant, but it is prompted to fail in close-up exploratory sequences. To process these sequences, the perspective assumption is needed. However, this assumption is not enough to constrain the problem and we need extra priors. The most extended is the local isometry (used in the Shape-from-Template methods) plus some bending regularizer. A first solution considering a regularly sampled surface mesh model was presented in Salzmann et al. (2008). Chhatkuli et al. (2014b) formulated the isometry through differential equations, assuming that the observed surfaces are approximated as infinitesimally planar. Parashar et al. (2017) formalized these assumptions in the context of Riemannian geometry, introducing a method able to recover the unit normals of the surface and manage unobserved features. Vicente and Agapito (2012) implemented stretching as soft-constraints in an energy minimization framework.

One fundamental challenge of the perspective methods is how to define the connection between the map points. Almost all the mentioned methods work by assuming that the points are contained in a smooth surface. Some of the physical models assume that the non-rigid object is a piecewise partition, i.e. a collection of pre-defined patches that move independently as rigid objects. Varol et al. (2009) were the first in using this strategy, followed by imposing a 3D global consistency in overlapping points. A relaxation to the piece-wise rigid constraint was given by Fayad et al. (2010), assuming each patch deforming with a quadratic physical model, thus, accounting for linear and bending deformations. All these methods required an initial patch segmentation and the number of overlapping points, to this end Russell et al. (2011) optimize the number of patches and overlap through an energy-based optimization. In contrast, Taylor et al. (2010) constructs a triangular mesh, connecting all the points, and considering each triangle as being locally rigid, being able to deal with topological changes.

Beyond the incredible merit of reconstructing a deforming surface, one of the main disadvantages of these methods is the inability of extending the map. The NRSfM methods usually focus on a small object covered by the field of view of the camera. In Chapter 3, we conceive the first deformable method using as a base the work proposed by Parashar et al. (2017). The key feature of our module is that apart from reconstructing the shape of the objects for each keyframe, it is able to extend the map by assembling these parts together by non-rigid surface alignment, also known in literature as *Non-Rigid puzzle* (Litany et al., 2016). By extending the map, DefSLAM is the first system able to perform exploration in new deforming zones (See Fig.1.5). In addition, the computational cost of the NRSfM methods is usually bigger than the cost for the template-based algorithms. That is the reason why we run the deformable tracking and mapping in parallel, achieving real-time performances.

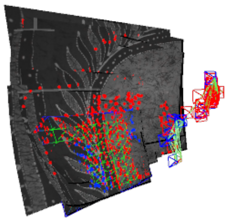

Fig. 1.5 Deformable SLAM exploration. The deformable tracking recovers the deformation and the camera pose for the current frame (in green). The deformable mapping uses the matches given by the tracking to estimate the shape of the map for the current keyframes (blue and red frames) and assemble it with the previous map. Map points in red.

1.4 Contributions

We have already mentioned the different topics covered in this thesis in the previous sections. Regarding the document structure, we opted for including the four more relevant publications as the four main chapters of the thesis. We aim to benefit from the careful editing process of top tier venues that result in self-contained and easy to read chapters, attempts to rewrite them would easily have resulted in poorer chapters. On the other hand, this approach implies repetitions specially in the introductions and related work, some changes in the notation and sub-optimal cross-referencing between chapters. In any case, we find that the advantages clearly outweigh the inconveniences.

The starting point of our research was the monocular ORBSLAM (Mur-Artal et al., 2015) sequential processing composed of two main threads: the tracking and the mapping. The tracking estimates tracking camera pose assuming a known map. The mapping builds the map from the matches of sparse features among the keyframes. The available geometrical information after processing every frame/keyframe is exploited to robustify and speed up the processing of the incoming new frame/keyframe, constraining the search for matches of discrete features described by an ORB binary descriptor. Once the discrete matches are available, a non-linear optimization refines the available geometry estimation which converges fast to the new optimum yielding a better estimation to the geometry.

Our first contribution, described in chapter 2, is to conceive a deformable tracking thread able to estimate the camera pose and the map deformation assuming that a set of well spread ORB features in the current frame have been matched with a shape at rest that plays the role of the deformable map. It is assumed the deformable map is available. We propose a regular triangular mesh whose control variables are the 3D coordinates of the mesh nodes. One of the issues was how to relate the matched features with the nodes, we opted for the barycentric coordinates. We state the deformable tracking as the optimization of a target function that combines deformation energy with reprojection error. The resulting system operates in real time and it was published as a demo session and the corresponding manuscript as a 2018 ECCV workshop:

- Lamarca, J., & Montiel, J. M. M. Camera tracking for SLAM in deformable maps. Proceedings of the European Conference on Computer Vision (ECCV) Workshops 2018. pg. 730–737.
- ECCV Demo Session Thursday, September 13, 4 PM - 6 PM. Camera Tracking for SLAM in Deformable Maps. J. Lamarca, J.M.M. Montiel

Our second contribution is described in Chapter 3 and it is the central contribution of the thesis: conceiving DefSLAM, the first ever visual monocular SLAM operating in deformable scenes. Mainly, we devised the mapping thread for deforming scenes from a calibrated monocular perspective sequence, then we integrated it with the deformable tracking thread of chapter 2 running in parallel. The mapping is built on top of the IsoNRSfM proposed in Parashar et al. (2017), we devised how to compute the ORB matches sequentially, the keyframe creation policy and the expansion of the map when new regions are visited. For the experimental evaluation, we created the Mandala dataset. This dataset contains stereo sequences with increasing levels of deformation, i.e. faster and bigger deformations of a mandala kerchief. The stereo images allow to compute the ground truth for the estimated deforming maps. DefSLAM was validated on the Mandala dataset and on a selection of sequences of the Hamlyn dataset. This research was developed in a cooperation with Prof. Bartoli and Dr. Parashar from Université Clermont Auvergne and it was the result of a research visit on 2019. The results have been published as:

- Lamarca, J., Parashar, S., Bartoli, A., & Montiel, J. M. M. DefSLAM: Tracking and mapping of deforming scenes from monocular sequences. IEEE Transactions on robotics, 37(1), 291-303.

The code is available under GPLv3.1 license at:

- Lamarca, J., Parashar, S., Bartoli, A., & Montiel, J. M. M. DefSLAM https://github.com/UZ-SLAMLab/DefSLAM

DefSLAM algorithm heavily relies on the ORB matches. The better the matches are, the better the performance is. Unfortunatelly, ORB points produce weak matches. The poor repeatability of the FAST detector struggles to initialize points, and once a map point is initialized, it goes on struggling to match the point in the subsequent frames, resulting in short intermittent point tracks. In Chapter 4, we propose a semi-direct matching based on the photometric Lucas-Kanade tracking able to produce long and continuous tracks that boosts the performance of DefSLAM. To further increase the robustness we also add relocalization and a surgical tool removal for surgical images. The system that includes all these matching advances, called SD-DefSLAM, has been published as:

- Rodríguez, J. J. G.*, Lamarca, J.*, Morlana, J., Tardós, J. D., & Montiel, J. M. M. Sd-defslam: Semi-direct monocular slam for deformable and intracorporeal scenes. In 2021 IEEE International Conference on Robotics and Automation (ICRA) (*Equal contribution)

This research intertwines the deep understanding of J. Lamarca the main author of DefSLAM with the solid background of J. J. Gómez-Rodríguez in LK tracking. Resulting in a tight integration that boosts the DefSLAM performance, particularly in medical scenes.

Our last contribution, Chapter 5, is a revisiting of the foundations of camera tracking thread for deforming environments. In the previous chapters, we modeled the scene as a continuous surface with a planar topology, what severely limits the applicability in medical scenes where discontinuities or tubular shapes are frequent. We propose to code the map surface as a set of disconnected 3D textures planar patches, called surfels. Encouraged for the nice performance of the semi-direct matches, we target a full photometric error minimization that recomputes the matches at each iteration of the optimization, avoiding the hard data association of the feature based methods and achieving subpixel accuracy. The experiments display how we can achieve longer tracks and recover more accurately the map deformation and the camera pose than the feature based DefSLAM- and the semi-direct SD-DefSLAM deformable tracking. The results have been submitted for publication in:

- J. Lamarca, J. J. Gómez-Rodríguez, J. D. Tardós, J.M.M. Montiel Direct and Sparse Deformable Tracking. Submitted to IEEE Robotics and Automation Letters with ICRA 2022 option.

In addition to the directly related with the thesis publications, there was a collaboration out-of-the-scope of the thesis that was not included:

- D. Recasens, J. Lamarca, J.M. Fácil, 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.

This paper presented a monocular rigid direct odometry system developed by J. Lamarca that used a CNN to predict the depth for the keyframes made in collaboration with the Dr. J.M. Fácil, expert in single-view estimation. To make it able to process in-body sequences we use the network Monodepth 2 (Godard et al., 2019) trained in the Hamlyn dataset by David Recasens during his Master Thesis.

The thesis results were disseminated in the "IEEE RAS Winter School on SLAM in Deformable Environments" [1] held in Australia 5-9 July 2021. DefSLAM and SD-DefSLAM where the core of the next sessions:

- Lecture, Deformable SLAM, J. Lamarca
- Tutorial, Deformable SLAM development and applications J. Lamarca
- Supervision of 3 student projects based on DefSLAM. Two of them were awarded a price: Laura Oliva Maza, Antonella Wilby, Alex McClung, Shi Zhou, Scarlett Liu, who obtain first prize of the Winter School and Yury Brodskiy, Hemanth Kanner, Olaya Alvarez Tunon, Luiza Ribeiro Marnet, Reuben Docea, who obtain the third prize.

In addition the candidate has co-supervised the next end-of-grade projects:

- Repetible pairing on visual SLAM in medical endoscopy environment. Lozano Puñet, Rodrigo. Co-supervised by Montiel, J.M.M. and Lamarca, J. Trabajo Fin de Grado. Universidad de Zaragoza 2020.
- Numerical estimation of differential properties of the deformation field in deformable SLAM with FlowNet2. Royo, Diego, Co-supervised by Montiel, J.M.M. and Lamarca, J. Trabajo Fin de Grado. Universidad de Zaragoza 2020
- Evaluation and processing of medical scenes with non-rigid VSLAM system. Morlana Ledesma, Javier, Co-supervised by Montiel, J.M.M. and Lamarca, J. Trabajo Fin de Máster. Universidad de Zaragoza 2019

[1] https://www.uts.edu.au/research-and-teaching/our-research/centre-autonomous-systems/events/ieee-ras-winter-school-slam-deformable-environments

Chapter 2

Camera Tracking for SLAM in Deformable Maps

The current SLAM algorithms cannot work without assuming rigidity. In this chapter, we present the first real-time tracking thread for monocular VSLAM systems that manages deformable scenes. It is based on top of the Shape-from-Template (SfT) methods to code the scene deformation model. Our proposal is a sequential method that manages efficiently large templates, i.e. deformable maps estimating at the same time the camera pose and deformation. It also can be relocated in case of tracking loss. We have created a new dataset to evaluate our system. Our results show the robustness of the method in deformable environments while running in real time with errors under 3% in depth estimation.

2.1 Introduction

Recovering 3D scenes from monocular RGB-only images is a significantly challenging problem in Computer Vision. Under the rigidity assumption, Structure-from-Motion (SfM) methods provide the theoretical basis for the solution in static environments. Nonetheless, this assumption renders invalid for deforming scenes as most medical imaging scenarios. In the case of the non-rigid scenes the theoretical foundations are not yet well defined.

We can distinguish two types of algorithms that manage non rigid 3D reconstruction: Non-Rigid Structure-from-Motion (NRSfM), which are mostly batch processes, and Shape-from-Template (SfT), which work frame-to-frame. The main difference between these methods is that NRSfM learns the deformation model from the observations while SfT assumes a previously defined deformation model to estimate the deformation for each image.

Rigid methods like Visual SLAM (Simultaneous Localisation and Mapping) have made headway to work sequentially with scenes bigger than the camera field of view (Mur-Artal et al., 2015, Concha and Civera, 2015, Klein and Murray, 2007, Engel et al., 2017). Meanwhile, non-rigid methods are mostly focused on reconstructing structures which are entirely imaged and tracked, for example, surfaces (Chhatkuli et al., 2014a, Ngo et al., 2016, Salzmann and Fua, 2011), faces (Bregler et al., 2000, Torresani et al., 2008, Bartoli et al., 2008, Paladini et al., 2010), or articulated objects (Russell et al., 2011, Lee et al., 2014).

We conceive the first real-time tracking thread integrated in a SLAM system that can locate the camera and estimate the deformation of the surface based on top of a SfT algorithm following Salzmann and Fua (2011), Ngo et al. (2016), Perriollat et al. (2011), Bartoli et al. (2015). Our method includes automatic data association and PnP+RANSAC relocalisation algorithm. We code the deformable map as a template which consists of a mesh with a deformation model. Our template is represented as a 3D surface triangular mesh with spatial and temporal regularisers that are rotation and translation invariant. We have selected a meshh to represent our map because it is suitable for implementing physical models. In addition, we can relate the observations with the template with barycentric coordinates.

We evaluate our algorithm with experimental validation over real data both for camera location and scene deformation. This is the first work that focuses on recovering the deformable 3D just from partial images. Thus, we have created a new dataset to experiment with partially-imaged template for sake of future comparison.

2.2 Problem formulation

2.2.1 Template definition

We code the deformable structure of the scene as a known template $\mathcal{T} \subset \mathbb{R}^3$. The template is modelled as a surface mesh composed of planar triangular facets $\mathcal{F}$ that connect a set of nodes $\mathcal{V}$. The facet f is defined in the frame i by its three nodes $V^i_{f_j} = \{V^i_{f,h}\}\ h = 1\dots3$. The mesh is measured through observable points $\mathcal{X}$ which lie inside the facets. To code a point $X_j \in \mathcal{X}$ in frame i wrt. its facet f_j nodes, we use a piecewise linear interpolation through the barycentric coordinates $\mathbf{b}_j = [b_{j,1}, b_{j,2}, b_{j,3}]^\top$ by means of the function $\varphi : [\mathbb{R}^3, \mathbb{R}^{3x3}] \to \mathbb{R}^3$:

$$\mathbf{X}^i_j = \varphi(\mathbf{b}_j, \mathbf{V}^i_{f_j}) = \sum_{h=1}^{3} b_{j,h} \mathbf{V}^i_{f_j,h} \tag{2.1}$$

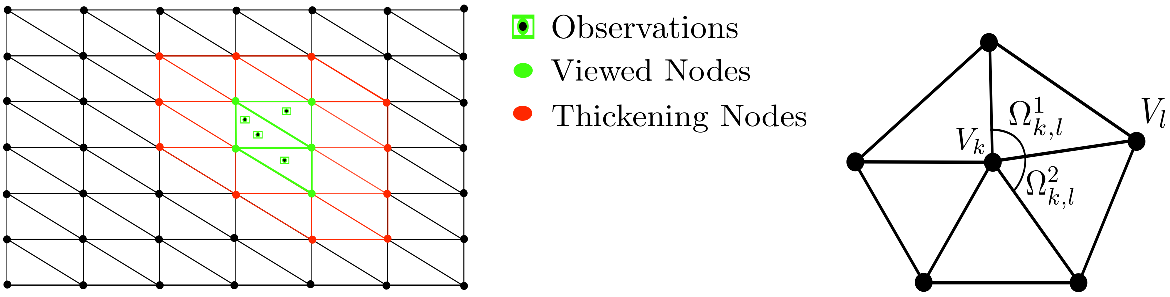

Fig. 2.1 Left: Two step region definition for the case of three observations inside two unconnected facets. $d_{\mathcal{K}=1}$ for the thickening $\mathcal{K}_i$. Right: Ring of neighbours $\mathcal{N}_k$ of the node K.

The camera is assumed projective, the observable point $\mathbf{X}_j^i \in \mathcal{T}$ defined in $\mathbb{R}^3$ is viewed in the frame i with the camera located in the pose $\mathbf{T}_i$ through the projective function $\pi : [SE(3), \mathbb{R}^3] \rightarrow \mathbb{R}^2$.

$$\pi\left(\mathbf{T}_i, \mathbf{X}_j^i\right) = \begin{bmatrix} f_u \frac{x_j^i}{z_j^i} + c_u \\ f_v \frac{y_j^i}{z_j^i} + c_v \end{bmatrix} \tag{2.2}$$

$$\begin{bmatrix} x_j^i & y_j^i & z_j^i \end{bmatrix}^T = \mathbf{R}^i \mathbf{X}_j^i + \mathbf{t}^i \tag{2.3}$$

Where $\mathbf{R}^i \in SO(3)$ and $\mathbf{t}^i \in \mathbb{R}^3$ are respectively the rotation and the translation of the transformation $\mathbf{T}_i$ and $\{f_u, f_v, c_u, c_v\}$ are the focal lengths and the principal point that define the projective calibration for the camera. The algorithm works under the assumption of previously knowing the template. This is a common assumption of template methods. We efectivelly compute it by means of a rigid VSLAM algorithm Mur-Artal et al. (2015). We initialise the template from a 3D reconstruction of the shape surface at rest. We use Poisson surface reconstruction as it is proposed in Kazhdan et al. (2006) to construct the template triangular mesh from the sparse point cloud. Once the template is generated, only cloud points which lie close to a facet are retained and then projected into the mesh facets where their barycentric coordinates are computed.

2.3 Optimisation

We recover the camera pose and the deformation only in the template region detected by the camera. We define the *observation region*, $\mathcal{O}_i$, as the template nodes belonging to a facet with one or more matched observations in the current image i. We dilate the $\mathcal{O}_i$ region with a layer that we call *thickening layer*, $\mathcal{K}_i$ whose thickness is $d_{\mathcal{K}}$. We call the template region estimated in the local step *local map*, $\mathcal{L}_i$. It is defined as $\mathcal{L}_i = \mathcal{O}_i \cup \mathcal{K}_i$ (See Fig. 2.1).

We propose the next optimisation to recover both the camera pose T_i and the position of the local map nodes $V_k^i \in \mathcal{L}_i$, in frame i:

$$\begin{aligned} \underset{T_i, V_k^i \in L_i}{argmin} \quad & \frac{1}{N_\bullet} \sum_j \rho \left(\left\| \pi_i \left(T_i, \varphi(\mathbf{b}_j, V_{f_j}^i) \right) - x_j^i \right\|^2 \right) \\ & + \frac{\lambda_d}{N_\bullet} \sum_k \sum_{l \in \mathcal{N}_k} \left(\frac{\left\| V_k^i - V_l^i \right\| - \left\| V_k^0 - V_l^0 \right\|}{\left\| V_k^0 - V_l^0 \right\|} \right)^2 \\ & + \frac{\lambda_L}{N_\bullet} \sum_k \left(\|\delta_k^i\| - \|\delta_k^0\| \right)^2 \sum_{l \in \mathcal{N}_k} \frac{1}{\|V_k - V_l\|^2} \\ & + \frac{\lambda_T}{SN_\bullet} \sum_k \left\| V_k^i - V_k^{i-1} \right\| \end{aligned} \tag{2.4}$$

The weights of the regularisers $\lambda_L, \lambda_d, \lambda_t$ are defined with respect to a unit weight for the data term. Additionally, we consider different normalisation factors to correct the final weight assigned to each term. We consider a correction depending on the number of addends, denoted as $N_\bullet$, in the summation of the corresponding regularising term and a scale correction for the temporal term.

The nodes not included in the optimisation, whose position is fixed, $V_k^i \in \{\mathcal{T} \setminus \mathcal{L}_i\}$, are linked with those optimised, hence they are acting as boundary conditions. As a consequence most of the rigid motion between the camera and the template is included in the camera motion estimate $\mathbf{T}_i$.

The regularisers code our deformation model, they are inspired in continuum mechanics where bodies deform generating internal energies due to normal strain and shear strain. The first term is the Cauchy or engineering strain:

$$\sum_k \sum_{l \in \mathcal{N}_k} \left(\frac{\left\| V_k^i - V_l^i \right\| - \left\| V_k^0 - V_l^0 \right\|}{\left\| V_k^0 - V_l^0 \right\|} \right)^2 \tag{2.5}$$

It penalises the normal strain energy. Per each node V_k^i we consider a summation over the ring of its neighbours N_k. Per each neighbour the deformation energy is computed as proportional to the squared ratio between the distance increment and the distance at rest. Unlike other isometry or inextensibility regularisers, (Ngo et al., 2016, Gallardo et al., 2016), it is a dimensionless magnitude, invariant with respect to the facet size. Per each node V_k^i we consider its ring of neighbours $\mathcal{N}_k$ in the computation.

The second regulariser is the bending energy:

$$\sum_{k}\left(\|\delta_k^i\|-\|\delta_k^0\|\right)^2\sum_{l\in\mathcal{N}_k}\frac{1}{\left\|V_k^i-V_l^i\right\|^2} \tag{2.6}$$

It penalises the shear strain energy. It is coded as the squared ratio between the deflection change and the mean edge length in its ring of neighbours $\mathcal{N}_k$. We use the ratio in order to get dimensionless magnitude invariant to the facet size. The deflection δ_k^i also represents the mean curvature, it is computed by means of the discrete Laplace-Beltrami operator:

$$\delta_k^i = V_k^i - \frac{1}{\sum_{l\in\mathcal{N}_j}\omega_l}\sum_{l\in\mathcal{N}_j}\omega_l V_l^i \tag{2.7}$$

in order to cope with irregular and obtuse meshes, ω_l is defined by the so-called mean-value coordinates Floater (2003):

$$\omega_l = \frac{\tan(\Omega_{k,l}^1/2)+\tan(\Omega_{k,l}^2/2)}{\left\|V_k^0-V_l^0\right\|} \tag{2.8}$$

The $\Omega_{k,l}^1$ and $\Omega_{k,l}^2$ angles are defined in Figure 2.1.

The last term codes a temporal smoothing between the nodes in $\mathcal{L}_i$. This term is dimensionless with the average length of the arcs in the mesh represented by S. We joinly optimise the reprojection error and the deformation energy with the Levenberg–Marquardt algorithm implemented in the library g2o Kümmerle et al. (2011).

2.4 SLAM Pipeline

To compose the entire tracking thread, we integrate the optimisation in a pipeline with automatic data association working with ORB points, and a DBoW keyframe database (Gálvez-López and Tardós, 2012) that allows relocalisation in case of losing the tracking.

Our optimisation method uses as input the observations of the template points in the current frame. Specifically, multiscale FAST corner to detect the observations, and the ORB descriptor (Rublee et al., 2011) to identify the matches. We apply the classical in VSLAM *active matching*, that sequentially process the image stream. First, the ORB points are detected in the current image. Next, with a camera motion model, it is predicted the camera pose as a function of the past camera poses. Then, the last template estimate and the barycentric coordinates are used to predict where the template points would be imaged. Around the template point prediction it is defined a search region. Among the ORB points inside the search region, the one with the closest ORB descriptor is selected as the observation. We apply a threshold on the ORB similarity to definitively accept a match. The ORB descriptor of the template point is taken from the template initialisation. The similarity is estimated as the Hamming distance between the ORB descriptors. To reduce the number of false negatives, we cluster the matches according to their geometrical innovation, difference between the predicted template point in the image and the detected one. Only the three main clusters of matches are retained.

For relocalisation, we use a relaxed rigid PnP + RANSAC algorithm. We estimate the initial solution with a PnP and refine it with a deformable optimization. We tested the original rigid PnP in five thousand images that contain deformation and we got a recall of 26% successful relocalisation. With the proposed relaxed method up to a 49%. The precision in the relocalisation is close to the 100%.

2.5 Experiments

Comparison with state of the art SfT. We benchmark our proposal with the standard *Kinect paper dataset*, to compare the performance of our deformation model with respect to state-of-the-art template-based algorithms. Kinect paper dataset is composed of 193 frames, each frame contains around 1300 observations coming form SIFT points. The matches for the observations are also provided. The ground truth for the matched points are computed from a Kinect RGB-D sensor. The benchmark considers a template that can be fit within the camera field of view. To make an homogeneous comparison we fixed the camera and leave the boundaries of the mesh free. In table 2.1 we show the mean RMS error along the sequence compared with respect to some popular methods Chhatkuli et al. (2014a) [1], Bartoli et al. (2015)[2], Özgür and Bartoli (2017)[3], Salzmann and Fua (2011)[4], Östlund et al. (2012)[5], Brunet et al. (2010)[6]. Ours gets 4.86 mm at 13 ms per frame, what is comparable with the similar state-of-the-art algorithms Salzmann and Fua (2011), Östlund et al. (2012) and with a full data association stage.

Experimental validation. To analyse the performance of our system, we have created the *Kinect mandala dataset*. In this dataset, a mandala blanket is hanged and deformed manually

Table 2.1 RMSE averaged over all the frames in the sequence.

	[1]	[2]	[3]	[4]	[5]	[6]	Ours
Mean RMSE (mm)	3.97	4.56	3.78	7.47	4.82	3.86	4.86
Runtime per Frame (ms)	2	0.7	7	5	30	116	13

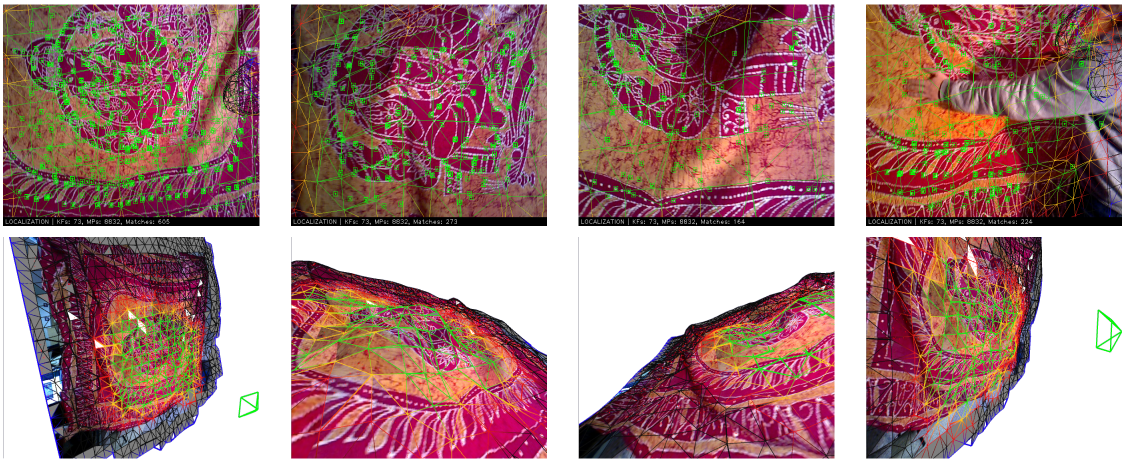
LOCALIZATION | KFs: 73, MPs: 8832, Matches: 605
LOCALIZATION | KFs: 73, MPs: 8832, Matches: 273
LOCALIZATION | KFs: 73, MPs: 8832, Matches: 164
LOCALIZATION | KFs: 73, MPs: 8832, Matches: 224

Fig. 2.2 From left to right: frames #1347, #2089, #9454, #10739, corresponding to the shape at rest and different deformations. Top: 2D image Bottom: 3D reconstruction

from its back surface, meanwhile a hand-held RGB-D camera closely observes the mandala surface mimicking a scanning movement in circles. Due to the limited field of view of the camera and its proximity to the cloth, the whole mandala is never completely imaged. We run the experiments in a Intel® Core™ i7-7700K CPU @ 4.20GHz × 8 with a 32GB of RAM memory.

The sequence is composed of ten thousand frames, there is a first part for initialisation where the cloth remains rigid. After that, the level of hardness of the deformation is progressively increased. The video captures from big displacements in different points of the mandala to wrinkled moments and occlusions.

We evaluate the influence of the thickening layer size, $d_{\mathcal{K}}$. As result of the experiment, we get a system that can run in real-time and have an RMS error of 2.30%, 2.22%, and 2.32% for $d_{\mathcal{K}} = 0, 1$ and 2 respectively. When it comes to runtime, the optimisation algorithm is taking 17, 19 and 20 ms, and the total times per frame are 39, 40 and 41 ms. With $d_{\mathcal{K}=1}$ we get to reduce the error without increasing excessively the time.

2.6 Discussion

We present a new tracking method able to work in deformable environment incorporating SfT techniques to a SLAM pipeline. We have developed a full-fledged SLAM tracking thread that can robustly operate with an average time budged of 39 ms per frame in very general scenarios with an error under 3% in a real scene and with a relocalisation algorithm with a recall of a 46% in deformable environments with a precision close to the 100%. In the next Chapter, we formulate a full deformable SLAM pipeline built on top of the deformable tracking presented.

Chapter 3

DefSLAM: Tracking and Mapping of Deforming Scenes from Monocular Sequences

Monocular SLAM algorithms perform robustly when observing rigid scenes, however, they fail when the observed scene deforms, for example, in medical endoscopy applications. In this Chapter, we present DefSLAM, the first monocular SLAM capable of operating in deforming scenes in real-time. Our approach intertwines Shape-from-Template (SfT) and Non-Rigid Structure-from-Motion (NRSfM) techniques to deal with the exploratory sequences typical of SLAM. A deformable tracking thread recovers the pose of the camera and the deformation of the observed map by means of SfT processing a template that models the scene shape-at-rest at frame rate. A deformable mapping thread runs in parallel with the tracking to update the template by means of an isometric NRSfM processing a batch of full perspective keyframes at keyframe rate. In our experiments, DefSLAM processes close-up sequences of deforming scenes, both in a laboratory controlled experiment and in medical endoscopy sequences, producing accurate 3D models of the scene with respect to the moving camera.

3.1 Introduction

The goal of visual Simultaneous Localization and Mapping (SLAM) algorithms is to locate a visual sensor in an uncertain map which is being estimated simultaneously. The typical use case in SLAM includes exploratory trajectories where the camera images a scene without previous information of the structure observed. Using a monocular sensor, visual SLAM

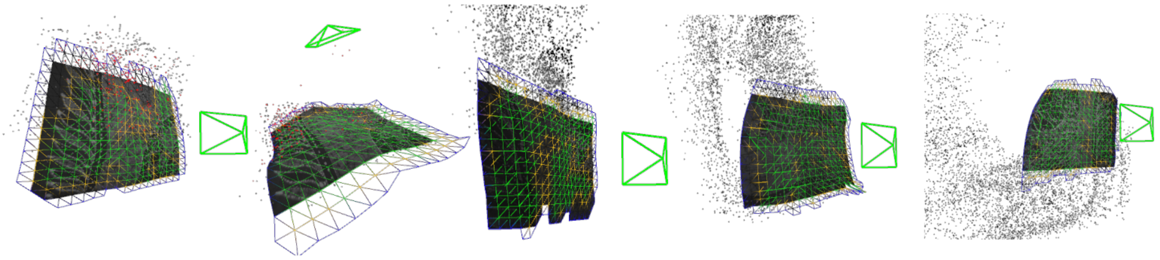

Fig. 3.1 Real-time reconstruction of a deforming scene with DefSLAM. The mandala kerchief deforms while the camera moves. DefSLAM locates the camera shown as a green frustum, while recovering the deformation of the kerchief using a template of the same. The estimated 3D deformable map is expanded when new regions are explored by reestimating new templates. The map is composed of sparse 3D points, in black, and a template as triangular mesh, viewed part in green.

has to process several images rendering enough parallax to recover the map for the new scene region *wrt.* the camera. Once the map is available, the camera can be localized *wrt.* this map from just one image as long as the camera does not move to unexplored areas. The rigidity assumption constrains the problem significantly, and it is intensively exploited by state-of-the-art monocular SLAM systems (Engel et al., 2017, Klein and Murray, 2007, Mur-Artal et al., 2015).

However, the rigidity assumption rends invalid in applications where the deformation is predominant. To this end, we introduce **DefSLAM**, a calibrated monocular and deformable SLAM system which can perform in deforming, i.e. non rigid, environments. A relevant use case is medical endoscopy, where monocular visual SLAM is crucial a tool for augmented reality and autonomous medical robotics.

In the literature, non-rigid monocular scenes have been handled by Non-Rigid Structure-from-Motion methods (Chhatkuli et al., 2014a, 2016, Parashar et al., 2017, Taylor et al., 2010, Vicente and Agapito, 2012) and Shape-from-Template methods (Chhatkuli et al., 2017, Lamarca and Montiel, 2018b, Ngo et al., 2016, Salzmann and Fua, 2011). NRSfM methods are able to recover the evolution of the 3D scenes non-rigid deformations from a set of monocular images, after a computationally demanding batch processing of the images. In contrast, SfT recovers the 3D deformation from a single image, at a low computational cost but needs a template. The template is a 3D textured model describing the shape at rest of the scene. DefSLAM framework combines the advantages of the two classes of non-rigid monocular methods. We propose a parallel algorithm composed of a *deformable tracking* thread as the front-end running SfT at frame rate, and a *deformable mapping* thread as the back-end running NRSfM to compute the SfT template at a slower keyframe rate.

Fig. 3.1 shows DefSLAM processing a sequence processed where the camera is being located *wrt.* a deforming kerchief being mapped simultaneously using images from a monocular sensor from partial observations of different regions of the kerchief. The *deformable tracking* thread recovers the camera pose and the deformation of the map at frame rate. It uses a template for the viewed part of the map to recover the map points deformation by minimizing a combination of reprojection error and deformation energy for each frame. The *deformable mapping* thread initializes and refines map estimates, and extends the map when new regions are visited. It processes just a selection of frames – keyframes – imaging the same region to define the shape-at-rest of the template used by the *deformable tracking* thread to process the subsequent frames.

We validated our DefSLAM algorithm in monocular sequences that include exploratory trajectories observing deforming scenes. We evaluate DefSLAM on new waving mandala kerchief dataset which we created and an in-vivo medical endoscopy Hamlyn dataset (Mountney et al., 2010a). To make some comparison we have resorted to systems with a different configuration than ours. We compare our results with the state-of-the-art rigid monocular ORBSLAM (Mur-Artal et al., 2015) to display the DefSLAM unique capability to SLAM deforming scenes. We also compared with MISSLAM (Song et al., 2018), the closest in the literature offering SLAM accuracy results in medical deformable scenes, despite it is stereo in contrast to our monocular system. These experiments validate the unprecedented ability of DefSLAM to accurately code the structure of the scene in rigid and deformable scenarios, including medical cases.

3.2 Related Work

3.2.1 SLAM

Deformable visual SLAM. The deformable SLAM methods in the literature rely on sensors providing depth information, *i.e.* RGB-D or stereo sensors. DynamicFusion (Newcombe et al., 2015) is a seminal work in deformable VSLAM with an RGB-D camera. It fuses the frame-by-frame depth information into a canonical shape, i.e. a shape at rest, that incrementally maps the entire scene after an exploratory trajectory of partial observations. This canonical shape is deformed to the current keyframe with the as-rigid-as-possible deformation model (Sorkine and Alexa, 2007). In the work proposed by Innmann et al. (2016), the quality of the deformation is improved by including the photometric error in the optimization. Gao and Tedrake (2018) substituted the volumetric representation by surfels to improve the efficiency of the algorithm. These methods recover the whole canonical shape deformation which is usually small. This technique is not scalable to bigger shapes like exploratory scenes in endoscopy. Song et al. (2017) propose to use an embedded deformation model Sumner et al. (2007) instead of the as-rigid-as-possible model because it better preserves the local details under the deformation. In MISSLAM (Song et al., 2018), the system is enhanced with the tracking of a rigid system ORBSLAM (Mur-Artal et al., 2015) to achieve better tracks and more robust deformable SLAM for medical endoscopy exploration. In any case, all these algorithms optimize the whole map each time and thus scale poorly with the size of the map. We aim similar SLAM capabilities in deformable scenes, but in the challenging monocular case. In addition, our approach only optimizes the observed map zone achieving good scalability *wrt.* the size of the map, being able to be run on the CPU.

Rigid visual SLAM. Monocular rigid VSLAM is a mature field. The current state-of-the-art monocular rigid VSLAM methods (Engel et al., 2017, Mur-Artal et al., 2015) provide accurate, robust and fast results in robotic scenes. Some works have attempted to apply rigid methods in in-vivo medical quasi-rigid scenes. Grasa et al. (2014) proposed an EKF-SLAM algorithm, and Mahmoud et al. (2018) get dense maps based on ORBSLAM (Mur-Artal et al., 2015). Marmol et al. (2019) use a rigid SLAM system to locate the camera in arthroscopic images. All of these methods assume that the deformation is negligible and hence that a purely rigid SLAM system is able to survive just by excluding from the map any deformed scene region. We aim to achieve a similar performance, but in scenarios where deformation is predominant, more specifically: real-time operation and capability to handle sequences of close-ups corresponding to exploratory trajectories.

3.2.2 Non-Rigid Monocular Techniques

The methods in the literature which aim to recover the structure of a non-rigid scene from monocular sequences are SfT and NRSfM.

Shape-from-Template. SfT methods recover the deformed shape of an object from a monocular image and the object's textured 3D shape at rest. This textured shape-at-rest of the object is the so-called **template**. These methods associate a deformation model with this template to recover the deformed shape. The main difference between these methods is the definition of the deformation model. We distinguish between analytic and energy-based methods. Among the analytic solutions, we focus on the isometric deformation which assumes that the geodesic distance between points in the surface is preserved. Isometry for SfT has proven to be well-posed and it quickly evolved to stable and real-time solutions (Bartoli et al., 2015, Chhatkuli et al., 2017, Collins and Bartoli, 2010). Energy-based methods (Agudo et al., 2014, Lamarca and Montiel, 2018b, Ngo et al., 2016, Salzmann and Fua, 2011) jointly minimize the shape energy *wrt.* the shape-at-rest and the reprojection error for the image correspondences. These optimization methods are well suited to implement sequential data association with robust kernels to deal with outliers.

Orthographic Non-Rigid Structure-from-Motion. The earliest non-rigid monocular techniques are NRSfM. These methods were formulated using statistical models, first proposed by Bregler et al. (2000). This work gave rise to a family of methods (Dai et al., 2014, Moreno-Noguer and Porta, 2011, Paladini et al., 2009) which used a low dimensional basis model to obtain the configuration of the 3D points from the images of a sequence. They exploited spatial reguralizers (Dai et al., 2014, Garg et al., 2013), temporal regularizers (Akhter et al., 2011) and spatio-temporal regularizers (Agudo and Moreno-Noguer, 2015, Gotardo and Martinez, 2011a,b). These methods may handle small surface deformations or articulated objects, but they usually fail with very large deformations. They use an orthographic camera model which is an approximation only valid when the scene is distant from the camera, this is a strong assumption invalid in many applications.

Perspective Non-Rigid Structure-from-Motion. Real use cases need the more accurate perspective camera model. It is able to model the close-up sequences typical in SLAM, especially in medical endoscopy. The isometry assumption, first proposed in SfT methods, has also produced excellent results in NRSfM (Chhatkuli et al., 2014a, 2016, Parashar et al., 2017, Taylor et al., 2010, Vicente and Agapito, 2012). It brought not only improvements in terms of accuracy, but also the ability to handle perspective cameras. Parashar et al. (2017) proposed a local method, able to handle naturally occlusions and missing data also usual in many applications.

Our approach. We propose the first visual SLAM system capable of working with deforming monocular sequences. We propose a *deformable tracking* thread based on the work presented in the Chapter 2, which uses a pre-computed template to recover the camera pose and the deformation of the scene. We also propose a *deformable mapping* thread which extends the map and estimates the shape-at-rest of the template in new explored zones by means of the isometric NRSfM proposed by Parashar et al. (2017). Our contribution is a new iterative scheme for the optimization in IsoNRSfM (Parashar et al., 2017) that allows to calculate and refine the solutions incrementally at keyframe rate. Both for the deformable mapping and tracking, we only optimize the part of the template observed having a runtime independent of the size of the map in exploratory sequences.

We also propose a sequential active matching that exploits the already available SLAM map to boost the data association performance. Our final contribution is to integrate in the deformable mapping a non-rigid alignment between surfaces to build a global map, extending alignment as proposed in Newcombe et al. (2015), Gao and Tedrake (2018), Song et al. (2018) to the monocular case.

The proposed deformable tracking and mapping algorithms run in parallel, in a similar way to the state-of-the-art rigid SLAM methods Engel et al. (2017), Klein and Murray (2007), Mur-Artal et al. (2015) to achieve real-time performances.

3.3 DefSLAM System Overview

DefSLAM recovers the structure of the scene, its deformation and the camera pose. It is composed of three main components:

- **The map**. The map represents the structure of the scene reconstructed by DefSLAM as a set of 3D map points. The map is deformable and the position of the map points evolves along the sequence. Each map point j is represented by its position $\mathbf{X}_j^t$ for each processed frame t.

  We save some selected frames in the map called keyframes. We refer to the keyframes in which a map point is initialized as anchor keyframes. After each new keyframe processing, one of the anchor keyframes is selected as the reference keyframe. The reference keyframe defines the template used by the deformable tracking to process the new incoming frames.

- **The deformable tracking thread**. This thread is the front-end of the system and runs at frame rate. It uses SfT to estimate the position of the map points $\mathbf{X}_j^t$ and the camera pose $\mathbf{T}_{tw}$ for each frame t. We embed the map points into the template $\mathcal{T}_k$ to compute their position. The shape-at-rest of the template $\mathcal{T}_k$ is the surface $\mathcal{S}_k$ observed in the reference keyframe k.

- **The deformable mapping thread**. This thread is the back-end of the system and runs at keyframe rate. It uses NRSfM to estimate the surface $\mathcal{S}_k$ observed in the keyframe k.

Notation We use calligraphic letters for sets of geometrical entities in the deforming scene, *e.g.* $\mathcal{X}$ for the set of all map points. Bold letters represent matrices and vectors. Scalars are represented in italics. The indexes t represent the frames and $\mathbf{T}_{tw}$ the pose of the frame at instant t. Superindexes represent the temporal instant of the estimation. The index j represents the map points, n the nodes and e the edges of the mesh describing the template surface.

3.4 Deformable Tracking

Deformable tracking recovers the camera pose $\mathbf{T}_{tw}$ and the shape of the template $\mathcal{T}_k^t$ in the frame t by jointly minimizing reprojection error and deformation energy. $\mathcal{T}_k$ is the surface reconstructed in the reference keyframe k. The tracking algorithm is composed of three stages: data association, camera pose estimation and template deformation and new keyframe selection. Next, it is detailed the template structure, the camera model and the three steps of the algorithm.

3.4.1 Template

The template is a surface parametrized with a 3D triangular mesh. It is composed of a set of planar triangular facets $\mathcal{F}$, defined by a set of nodes $\mathcal{V}$, and connected by a set of edges $\mathcal{E}$. The deformation of the map at frame t is defined through the pose of the nodes of the template $\mathcal{T}_k^t$. The facet $f \in \mathcal{F}$ at frame t is defined by the pose of its three nodes $V_{f_j}^t = \{V_{f,h}^t\},\ h = \{1,2,3\}$. The map points observed in the keyframe k are embedded in the facets of the mesh. The position of a map point $\mathbf{X}_j^t \in \mathcal{X}$ in frame t is defined with its barycentric coordinates, $\mathbf{b}_j = [b_{j,1}, b_{j,2}, b_{j,3}]^\top$, *wrt.* the position of the nodes of the face f_j :

$$\mathbf{X}_j^t = \sum_{h=1}^{3} b_{j,h} \mathbf{V}_{f_j,h}^t \text{ s.t. } b_{j,1} + b_{j,2} + b_{j,3} = 1. \tag{3.1}$$

3.4.2 Camera Model

We use the calibrated pinhole model. The projection of the 3D point j, $\mathbf{X}_j^t \in \mathcal{X}_k^t$ in the frame t by a camera located at $\mathbf{T}_{tw}$ is modelled by the projection function $\pi : [\mathrm{SE}(3), \mathbb{R}^3] \to \mathbb{R}^2$:

$$\begin{aligned} \pi\left(\mathbf{T}_{tw}, \mathbf{X}_j^t\right) &= \begin{bmatrix} f_x \frac{X_j^t}{Z_j^t} + C_x \\ f_y \frac{Y_j^t}{Z_j^t} + C_y \end{bmatrix}, \\ \text{where} \quad & \begin{bmatrix} X_j^t & Y_j^t & Z_j^t \end{bmatrix}^\top = \mathbf{R}_{tw} \mathbf{X}_j^t + \mathbf{t}_{tw}. \end{aligned} \tag{3.2}$$

$\mathbf{R}_{tw} \in SO(3)$ and $\mathbf{t}_{tw} \in \mathbb{R}^3$ are respectively the rotation and the translation of the transformation $\mathbf{T}_{tw}$. $\{f_x, f_y, C_x, C_y\}$ are the focal lengths and the principal points from the camera calibration. The set of observation in the image $\mathcal{I}^t$ are the keypoints x^t matched with a map point of $\mathcal{X}^t$. The map point $\mathbf{X}_j^t$ is projected in the normalized retina as $(\hat{x}_j^t, \hat{y}_j^t)$ where $\hat{x}_j^t = \frac{x_j^t - C_x}{f_x}$, $\hat{y}_j^t = \frac{y_j^t - C_y}{f_y}$ and $\left(x_j^t \quad y_j^t\right)^\top = \pi\left(\mathbf{T}_{tw}, \mathbf{X}_j^t\right)$.

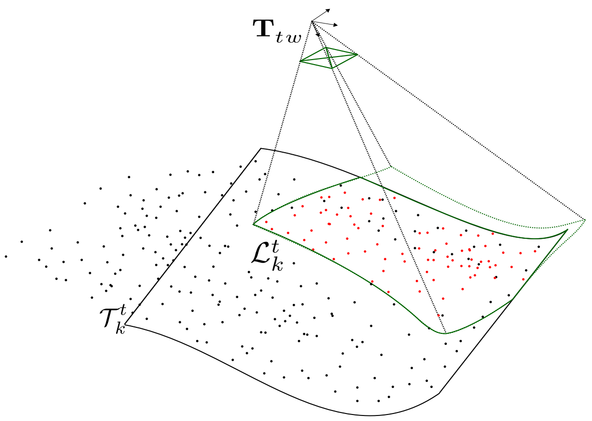

Fig. 3.2 Deformable tracking: estimating camera pose and deformation of the viewed map. $\mathcal{T}_k^t$ is the map shape in the frame t, $\mathcal{L}_k^t$ is the local map shape in the frame t and $\mathbf{T}_{cw}^t$ the camera pose. Black points belong to the global map. Some of them are embedded in the template. Current matched points in red.

3.4.3 Camera Pose and Template Deformation

In SLAM sequences, the camera usually images a zone smaller than the template. For efficiency and scalability, we only optimize the observed zone of the template and its closest vicinity. We refer to this part of the template as the local zone $\mathcal{L}_k^t \subseteq \mathcal{T}_k^t$. Figure 3.2 shows all the components of the deformable tracking: the template $\mathcal{T}_k^t$, the local zone $\mathcal{L}_k^t$ and the camera pose $\mathbf{T}_{tw}$.

To estimate the deformed $\mathcal{L}_k^t$ and $\mathbf{T}_{tw}$, we jointly minimize the reprojection error $\varphi_d(\mathcal{I}^t, \mathbf{T}_{cw}, \mathcal{L}_k^t)$ in the image I^t and the deformation energy $\varphi_e(\mathcal{L}_k^t, \mathcal{T}_k)$ of the template $\mathcal{T}_k$:

$$\underset{L_k^t, T_{tw}}{argmin} \quad \varphi_d(\mathcal{I}^t, \mathbf{T}_{tw}, \mathcal{L}_k^t) + \varphi_e(\mathcal{L}_k^t, \mathcal{T}_k). \tag{3.3}$$

We solve (3.3) using the Levenberg-Marquardt optimization method. The initial guess for $(\mathcal{L}_k^t, \mathbf{T}_{tw})$, is the solution of the previous frame, $(\mathcal{L}_k^{t-1}, \mathbf{T}_{t-1w})$. We fix the pose boundary nodes of $\mathcal{L}_k^t$ during the optimization to constraint the gauge freedoms of the camera pose $\mathbf{T}_{tw}$,

The reprojection error $\varphi_d(\mathcal{I}^t, \mathbf{T}_{tw}, \mathcal{L}_k^t)$ for the set of keypoints x^t in image $\mathcal{I}^t$ is defined as:

$$\varphi_d(\mathcal{I}^t, \mathbf{T}_{tw}, \mathcal{L}_k^t) = \sum_{j \in \boldsymbol{x}^t} \rho\left(\left\|\pi(\mathbf{X}_j^t, \mathbf{T}_{tw}) - \mathbf{x}_j^t\right\|\right). \tag{3.4}$$

The reprojection error is robust against outliers as it is weighted with a Huber robust kernel $\rho(.)$.

We define a deformation energy $\varphi_e(\mathcal{L}_k^t, \mathcal{T}_k)$ *wrt.* $\mathcal{T}_k$ as a combination of a stretching energy $\varphi_s(\mathcal{L}_k^t, \mathcal{T}_k)$, a bending energy $\varphi_b(\mathcal{L}_k^t, \mathcal{T}_k)$ and a reference regularizer $\varphi_r(\mathcal{L}_k^t, \mathcal{T}_k)$:

$$\begin{aligned}\varphi_e(\mathcal{L}_k^t, \mathcal{T}_k) = \lambda_s \varphi_s(\mathcal{L}_k^t, \mathcal{T}_k) + \lambda_b \varphi_b(\mathcal{L}_k^t, \mathcal{T}_k) \\ + \lambda_r \varphi_r(\mathcal{L}_k^t, \mathcal{T}_k).\end{aligned} \tag{3.5}$$

We use λ_s, λ_b and λ_r to weight the influence of each term.

The stretching energy $\varphi_s(\mathcal{L}_k^t, \mathcal{T}_k)$ measures the difference in the length l_e^t of each edge e in the local zone $\mathcal{L}_k^t$ in the frame t with respect to its length l_e^k in the shape-at-rest of $\mathcal{T}_k$:

$$\varphi_s(\mathcal{L}_t^k, \mathcal{T}_k) = \sum_{e \in \mathcal{L}_k^t} \left(\frac{l_e^t - l_e^k}{l_e^k} \right)^2. \tag{3.6}$$

The bending energy $\varphi_b(\mathcal{L}_k^t, \mathcal{T}_k)$ measures the changes in mean curvature δ_n^t in each node n *wrt.* the estimated δ_n^k in the shape-at-rest of $\mathcal{T}_k$. We estimate the mean curvature through the discrete Laplacian operator Floater (2003). We make the bending term dimensionless by dividing it by the mean distance l_e^k of the edges connected with the node $\mathcal{E}_n^k$:

$$\varphi_b(\mathcal{L}_k^t, \mathcal{T}_k) = \sum_{n \in \mathcal{L}_k^t} \sum_{e \in \mathcal{E}_n^k} \left(\frac{\delta_n^t - \delta_n^k}{l_e^k} \right)^2. \tag{3.7}$$

Optimization considering the terms $\varphi_d(\cdot)$, $\varphi_b(\cdot)$ and $\varphi_s(\cdot)$ allows to recover the relative pose of the camera with respect to the template, but the absolute camera pose is not observable. Thanks to the fixation of the $\mathcal{L}_k^t$ boundary nodes pose, the absolute camera pose becomes observable. However, the camera pose sometimes is only weakly observable depending on the boundary nodes geometrical distribution and cardinality. If the template is completely observed by the camera, then there are no boundary points to be fixed and the camera pose becomes fully non-observable.

We add another regularizer, $\varphi_r(\mathcal{L}_k^t, \mathcal{T}_k)$, that we call reference regularizer to keep the template as close as possible to its initial position in its reference keyframe, to alleviate the camera pose weak observability. It is given by:

$$\varphi_r(\mathcal{L}_k^t, \mathcal{T}_k) = \sum_{n \in \mathcal{L}_k^t} \left\| \mathbf{V}_n^t - \mathbf{V}_n^k \right\|. \tag{3.8}$$

Optimization (3.3) also needs the derivatives of the regularizers (3.6)-(3.8), they are detailed in Appendix A.

3.4.4 Data Association

To match the keypoints in the current frame with the map points, we apply an active matching strategy as proposed in Davison (2003). First, the ORB keypoints are detected in the current frame. Next, the camera pose is predicted with a camera motion model as a function of the past camera poses. Then, we use the last estimated shape of template and the barycentric coordinates to predict where the map points will be imaged. Around the map point prediction, we define a search region. We match the map point with the keypoint with the most similar ORB descriptor inside its search region. The similarity is estimated as the Hamming distance between the ORB descriptors, the match is accepted only if it is below a distance threshold. The ORB descriptor of the map point is taken from the keypoint of the keyframe where it was initialized.

3.4.5 New Keyframe Selection

We select a new keyframe as soon as the mapping thread finishes its processing. If the new keyframe covers a new map region, it becomes an anchor keyframe and the reference keyframe and a new template is created. Otherwise, the new keyframe is a regular keyframe, and its most covisible anchor keyframe is selected as the reference keyframe, and its template is refined.

3.5 Deformable Mapping

Deformable mapping recovers the observed map as a surface S_k for the reference keyframe k. This surface contains the map points observed in the keyframe during the tracking. With the new keyframe we refine the map points and create new ones. S_k defines the shape-at-rest of the template $\mathcal{T}_k$ for the deformable tracking for the next frames, as shown in Figure 3.3.

Deformable mapping is performed as follows: first, we compute the warps η_{kk^*} between the anchor keyframes k and the new keyframe k^*. At this stage, the considered anchor keyframes are those where one of the currently observed map points were initialized. Second, we estimate an up-to-scale surface $\overline{\mathcal{S}}_k$ by processing the covisible keyframes with the new keyframe by means of NRSfM. Third, we align $\overline{\mathcal{S}}_k$ with the previous map to recover the scale and the scaled surface $\mathcal{S}_k$. Finally, with this new surface, we create the new template by computing a triangular mesh and embedding the map points in its facets.

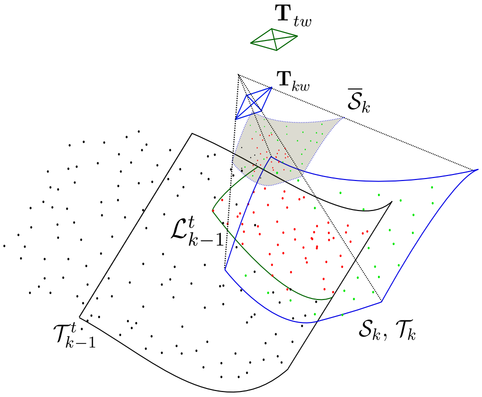

Fig. 3.3 Extension of the map in the deformable mapping. Local area $\mathcal{L}^t_{k-1}$ in green. Matched points in red. In blue, the up-to-scale surface estimated by NRSfM, $\overline{\mathcal{S}}_k$ (dotted line), and template $\mathcal{T}_k$ computed from the scaled surface $\mathcal{S}_k$ of the reference keyframe k.

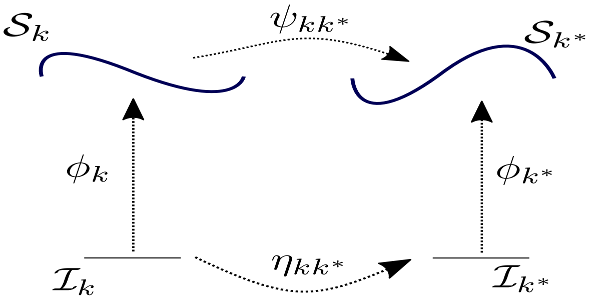

Fig. 3.4 Relation between an anchor keyframe k and one of its covisibles k^*. ϕ_k and ϕ_{k^*} are embeddings of the two keyframe surfaces k and k^*. η_{kk^*} is the warp between k and k^*. ψ_{kk^*} is the deformation field between the surfaces S_k and S_{k^*}

3.5.1 NRSfM

In isometric NRSfM, the surface deformation is modelled locally for each point under the assumption of isometry and infinitesimal planarity. Assuming infinitesimal planarity, any surface is approximated as a plane at an infinitesimal level, while maintaining its curvature at the global level. Isometric NRSfM can handle both rigid and non-rigid scenes. Since we use a local method, it can handle missing data and occlusions inherently. We build on the isometric NRSfM proposed in Parashar et al. (2017). For the sake of completeness, we summarize the formulation.

ϕ_k is the embedding of the scene surface $\mathcal{S}_k$, it is parametrized using the retina normalized coordinates of the image $\mathcal{I}_k$:

$$\begin{aligned} \phi_k : \mathbb{R}^2 &\mapsto \mathbb{R}^3 \\ \phi_k(\hat{x},\hat{y}) &= \begin{bmatrix} \frac{\hat{x}}{\beta(\hat{x},\hat{y})} & \frac{\hat{y}}{\beta(\hat{x},\hat{y})} & \frac{1}{\beta(\hat{x},\hat{y})} \end{bmatrix}^\top, \end{aligned} \tag{3.9}$$

where $\beta_k(\hat{x},\hat{y})$ is the inverse depth of each point. The normal $\vec{\mathbf{n}}_j(\hat{x},\hat{y})$ of the surface expressed *wrt.* this parametrization is given as:

$$\vec{\mathbf{n}}_j(\hat{x},\hat{y}) \propto \begin{pmatrix} K_{\hat{x}} \\ K_{\hat{y}} \\ 1 - \hat{x}K_{\hat{x}} - \hat{y}K_{\hat{y}} \end{pmatrix}, \tag{3.10}$$

where $K_{\hat{x}} = \frac{\beta_k(\hat{x},\hat{y})_{\hat{x}}}{\beta_k(\hat{x},\hat{y})}$ and $K_{\hat{y}} = \frac{\beta_k(\hat{x},\hat{y})_{\hat{y}}}{\beta_k(\hat{x},\hat{y})}$, the subindexes $\hat{x} - *$ and $\hat{y}$ denote the partial derivatives.

NRSfM exploits the relationship between the metric tensor $g_k(\hat{x},\hat{y})$, and the Christoffel symbols $\Gamma_k^{\hat{x}}(\hat{x},\hat{y})$ and $\Gamma_k^{\hat{y}}(\hat{x},\hat{y})$, of the surface of the keyframe $\mathcal{S}_k$ and those of its covisible keyframes $\mathcal{S}_{k^*}$. Assuming infinitesimal planarity and isometry, $\Gamma_k^{\hat{x}}(\hat{x},\hat{y})$ and $\Gamma_k^{\hat{y}}(\hat{x},\hat{y})$ only depend on $K_{\hat{x}}$ and $K_{\hat{y}}$ for each point in every keyframe image. The warp η_{kk^*} between the keyframes k and k^* represents the transformation from the image $\mathcal{I}_k$ to the image $\mathcal{I}_{k^*}$. Figure 3.4 shows the different elements of the two view relation, the warp η_{kk^*}, the surface embeddings for each keyframe ϕ_k and ϕ_{k^*}, and the isometric deformation ψ_{kk^*} between the surfaces $\mathcal{S}_k$ and $\mathcal{S}_{k^*}$. Due to the infinitesimal planarity and isometry assumptions, the metric tensor and the Christoffel symbols in two different surfaces k and k^* are related through the warp between these keyframes η_{kk^*} as:

$$g_k(\hat{x},\hat{y}) = J_{\eta_{kk^*}}^\top g_{k^*}(\hat{x}^*,\hat{y}^*) J_{\eta_{kk^*}} \tag{3.11}$$

$$\Gamma_k^q(\hat{x},\hat{y}) = \sum_h \frac{\partial \hat{x}_h}{\partial \hat{x}_h^*} (J_{\eta_{kk^*}}^\top \Gamma_k^h(\hat{x}^*,\hat{y}^*) J_{\eta_{kk^*}} + H_{\eta_{kk^*}}^h), \tag{3.12}$$

where $J_{\eta_{kk^*}}$ and $H_{\eta_{kk^*}}^q$ are the Jacobian and the Hessian for the variable $q = \{\hat{x},\hat{y}\}$ of the warp η_{kk^*} respectively. Eqs. (3.11) and (3.12) can be transformed in two cubic polynomial equations $P(K_{\hat{x}}^k, K_{\hat{y}}^k)$ and $Q(K_{\hat{x}}^k, K_{\hat{y}}^k)$ for each point correspondence:

$$\begin{aligned} P(K_{\hat{x}}^k, K_{\hat{y}}^k) &= \sum_{u,v \in [0,3]} p_{uv} (K_{\hat{x}}^k)^u (K_{\hat{y}}^k)^v = 0 \qquad (3.13) \\ Q(K_{\hat{x}}^k, K_{\hat{y}}^k) &= \sum_{u,v \in [0,3]} q_{uv} (K_{\hat{x}}^k)^u (K_{\hat{y}}^k)^v = 0, \qquad (3.14) \end{aligned}$$

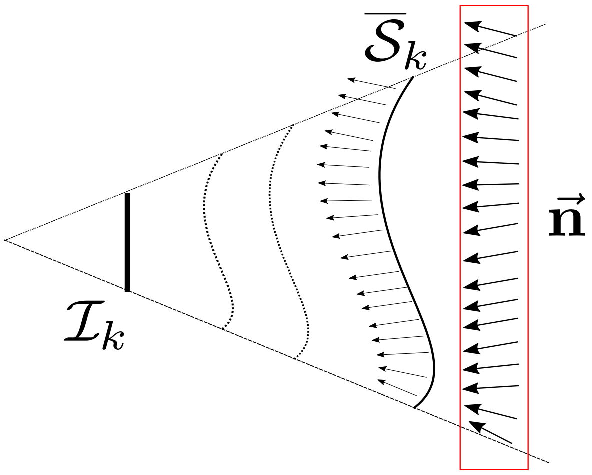

Fig. 3.5 $\overline{\mathcal{S}}_k$ is the estimated up-to-scale surface. $\vec{\mathbf{n}}$ are the set of normals. Two examples of surfaces at a different scale but having the same normals are displayed in dotted lines.

where the coefficients p_{uv} and q_{uv} depend only on the normalized coordinates of the points and the derivatives of first and second-order derivatives of the warp η_{kk^*}. We refer to the paper presented by Parashar et al. (2017) for further details in the coefficients p_{uv} and q_{uv}.

3.5.2 Incremental Surface Normals Refinement

If a point is matched in two or more keyframes, we can calculate its normal in its anchor keyframe k, defined by $K^k_{\hat{x}}$ and $K^k_{\hat{y}}$, by means of non-linear optimization:

$$\underset{K^k_{\hat{x}},K^k_{\hat{y}}}{argmin} \left(P\left(K^k_{\hat{x}},K^k_{\hat{y}}\right)\right)^2 + \left(Q\left(K^k_{\hat{x}},K^k_{\hat{y}}\right)\right)^2. \tag{3.15}$$

In contrast to Parashar et al. (2017), optimization (3.15) is incrementally computed. We initialize it with its last estimate achieving a fast convergence. Once the normals are refined in their anchor keyframe, we transfer the normals to the new reference keyframe with eq. 3.12. We recover the up-to-scale $\overline{\mathcal{S}}_k$ from the set of estimated normals $\vec{\mathbf{n}}$ using Shape-from-Normals (SfN) (Chhatkuli et al., 2014a). The surface $\overline{\mathcal{S}}_k$ is regressed with a bicubic b-spline parametrized by its control nodes depth. The control nodes are defined by a regular mesh in the image $\mathcal{I}_k$. We fit the depth of the nodes to obtain a surface orthogonal to the estimated normals with a regularizer in terms of bending energy (Fig. 3.5).

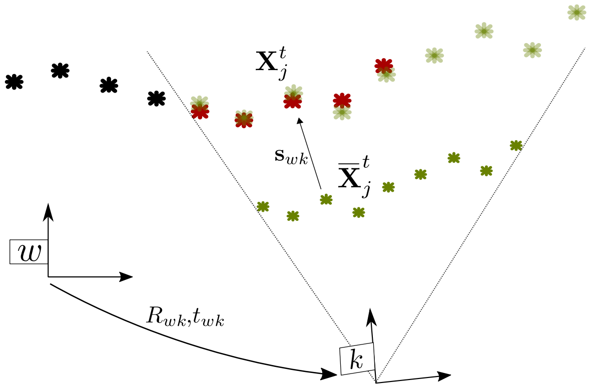

Fig. 3.6 Sim(3) alignment. We align the the map points $\overline{\mathbf{X}}_j^k \in \overline{\mathcal{S}}_k$ of the up-to-scale estimation with the pose of the map points $\mathbf{X}_j^k \in \mathcal{T}_{k-1}^k$ estimated for the frame k deforming the previous template $k-1$

3.5.3 Surface Alignment

The new estimated surface $\overline{\mathcal{S}}_k$ is up-to-scale. We need to recover the solution with a coherent scale s_{wk} *wrt.* the already estimated map. This means that the scale-corrected shape-at-rest $\mathcal{S}_k$ must have an scale coherent with the deformed template $\mathcal{T}_{k-1}^k$ estimated by the tracking when the keyframe was inserted.

We align these surfaces map points through a transformation which belongs the group of similarity of 3-space Sim(3), by means of non-linear optimization:

$$\underset{R_{wk}, t_{wk}, s_{wk}}{argmin} \sum_{j \in \mathbf{X}^k} \left\| \mathbf{s}_{wk} \mathbf{R}_{wk} \overline{\mathbf{X}}_j^k + \mathbf{t}_{wk} - \mathbf{X}_j^k \right\|^2 , \tag{3.16}$$

where $\mathbf{R}_{wk}$,$\mathbf{t}_{wk}$,$\mathbf{s}_{wk}$ are the rotation translation and scale defining the Sim(3) transformation (Fig. 3.6).

To build our new template $\mathcal{T}_k$, we finally create a triangular mesh from the scale-corrected surface $\mathcal{S}_k$ by means of regular triangular mesh in the image. The new map points 3D pose is computed from the matched keypoints by constraining them to be in the estimated surface $\mathcal{S}_k$. Then, we embed the re-observed map points and the new map points by projecting them into their corresponding template facet. With this embedding, we calculate the barycentric coordinates of the map points which will be used by the tracking.

3.5.4 Template Substitution

Once the surface S_k is computed, the keyframe k is set as the reference keyframe and the current template $\mathcal{T}_{k-1}$ is substituted by $\mathcal{T}_k$ computed from S_k. The shape observed in the

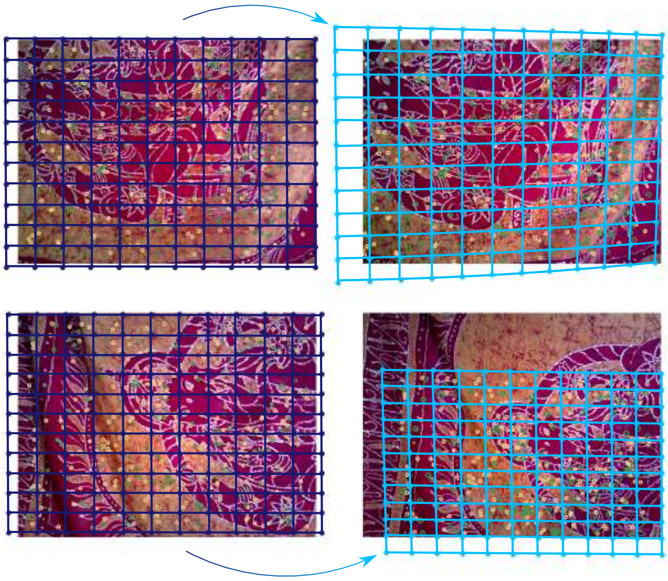

Fig. 3.7 Two examples of warp estimation. Warp estimation between the keyframe k (left) and k^* (right). The warp between k and k^* is plotted in blue. Yellow points are the initially matched map points, green points are the matches added by guided matching stage using the warp.

current frame t differs from the shape of the new template $\mathcal{T}_k$. This yields to failures in the data association stage, which assumes small deformations, if we substitute the template directly by $\mathcal{T}_k$. Therefore, we transfer the matches from $\mathcal{T}_{k-1}^t$ to $\mathcal{T}_k$ and compute the current shape $\mathcal{T}_k^t$ using optimization (3.3).

3.5.5 Warp Estimation and Non-Rigid Guided Matching

The input of NRSfM is the set of warps η_{kk^*} between an anchor keyframe k and their covisible keyframes k^*. The image warp η_{kk^*} is a function that transforms a point in the anchor keyframe into the corresponding point in its covisible k^*:

$$\eta_{kk^*} : [\hat{x}, \hat{y}] \in \mathbb{R}^2 \quad \mapsto \quad [\hat{x}^*, \hat{y}^*] \in \mathbb{R}^2.$$

First, we use a particular family of warps called Schwarps Pizarro et al. (2016), because, as discussed in Parashar et al. (2017), the formulation of the 2D Schwarzian equation regularizers are equivalent to the infinitesimal planarity of the NRSfM. See Figure 3.7 for two examples of warp between keyframes.

First, we estimate an initial warp between the anchor keyframe k and its covisible keyframe k^* with the matches given by the deformable tracking. Then we use the intial warp to perform a guided matching stage between the keypoints in keyframes k and k^*. We accept as a match the keypoint inside a search region with the smallest Hamming distance for the ORB descriptor. We apply a threshold on the ORB similarity to definitively accept a match. Once that we have the new matches we incorporate them to the initial ones and estimate the final warp. See Figure 3.7 for two examples of warp between keyframes.

3.5.6 SLAM Initialization

At initialization we need to have a template available for the scene surface. We compute it from the first frame of he sequence, assuming its surface $\mathcal{S}_1$, and hence its template $\mathcal{T}_1$ is a plane parallel to perpendicular to the camera optical axis.

With the second keyframe inserted, the mapping thread starts to compute a new template, that replaces the initial one. The accuracy of the first computed templates strongly depends on how many keyframes are fed in the NRSfM and on how large is the parallax they render.

According to the experiments, our algorithm can track from an inaccurate template with a high quality data association between keyframes, yielding long tracks and a low false positive rate. As a result, as more keyframes rendering high parallax are created, the estimated template eventually converges to the actual scene shape.

3.6 Implementation Details

The method is implemented in C++ and runs entirely on the CPU. We have used the OpenCV library (Bradski, 2000) for base computer vision functions. For the SfT optimization and the LS $\mathrm{Sim}(3)$ registration, we have used the g2o library (Kümmerle et al., 2011) and its implementation of Levenberg-Marquardt. For the Schwarps optimization, the normal estimation and the shape-from-normals, we have used the Ceres library (Agarwal et al., 2010). The runtime depends on the resolution of the mesh used as template. For a mesh of 10×10 nodes the runtime is approximately $50\,\mathrm{ms}$ for the deformable tracking thread and approximately $400\,\mathrm{ms}$ for the deformable mapping in a machine with an i7-4700HQ CPU and with 7.7 Gb RAM. The code will is available as a public git repository[1].

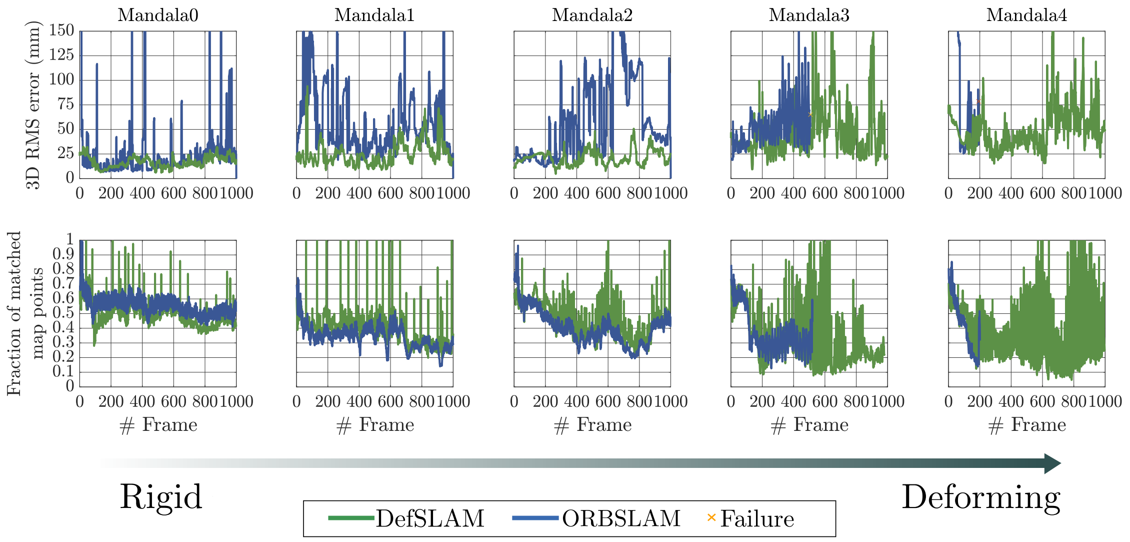

Fig. 3.8 Overall quality for Mandala dataset sequences. From left to right, the scenario contains more deformation. Top: 3D RMS error (mm) per frame (the smaller, the better). Bottom: Fraction of matched map points (the higher, the better).

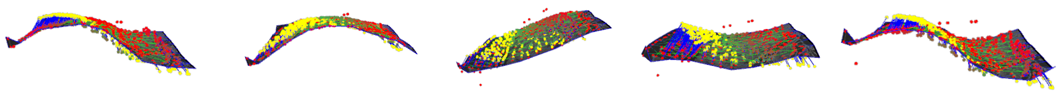

Fig. 3.9 Recovering local deformations in the mandala3 sequence. 3D map points in red, 3D point in yellow is the ground truth and blue lines are the difference. DefSLAM can perceive and reconstruct the deforming scene.

3.7 Experiments

We tested DefSLAM in two datasets. The first dataset is the Mandala dataset which we create to evaluate deformable monocular SLAM in a laboratory controlled situation. The second is a selection of sequences from the medical Hamlyn dataset (Mountney et al. (2010a), Stoyanov et al. (2005)), which comprises a phantom heart, and in-vivo sequences including exploratory trajectories. The sequences in both datasets have ground truth depth for each frame, either from stereo or from CT.

We focus on two per frame metrics: the 3D RMS error of the in-frustum map points and the fraction of matched map points. The RMS error is computed after a scale alignment for each frame of the sequence, it features the geometrical accuracy. The fraction of map points matched is the quotient between the map points effectively matched in the current frame,

[1] https://github.com/UZ-SLAMLab/DefSLAM

and the number of map points in-frustum of the current frame, i.e maximum number of map points that ideally can be matched. A low fraction signals a poor map that can only represent partially the scene imaged in the current frame.

In addition, we carried out an ablation analysis of the mapping and the tracking. In the mapping, we focused in NRSfM stages of the normals estimation. In the tracking, we evaluate the performance of the deforming template when compared with a rigid one. We also analyzed the sensitivity of the system to the tuning of the regularizers weights in the tracking optimization (eq. 3.5).

Currently, there is no other monocular SLAM for deformable environments to compare with. Thus we select a rigid monocular SLAM method, ORBSLAM Mur-Artal et al. (2015), as one of the closest for comparison. We had to re-tune several stages of ORBSLAM to process deforming sequences. 1) We relaxed the thresholds for matching and outlier rejection to retain matches despite the deformation. 2) We initialized it with the first frame ground truth map, to avoid the dramatical failure of the monocular intialization. 3) We decreased the rate of new keyframe creation up to one keyframe out of 3 frames, to adapt the map to the scene deformations. On the other hand we compare with MISSLAM (Song et al., 2018) in the Hamlyn phantom heart dataset, as the closest in the medical arena, despite MISSLAM is stereo instead of monocular.

For the sake of repeatability, DefSLAM was run sequentialized in single-thread, inserting one new keyframe every 10 frames. All the reported results are the median of 5 executions in each sequence. Some results are updated in the youtube video[2].

3.7.1 Mandala Dataset

We introduce the **Mandala dataset** to evaluate the map quality of deformable monocular SLAM systems in a controlled environment. It is composed of 5 sequences (640x480 pix. at 30 fps) with exploratory trajectories observing a textured kerchief deforming near-isometrically. We increased the hardness of deformation progressively by reducing the period of the waves generated on the kerchief and increasing their amplitude from the shape-at-rest. Fig. 3.10 shows the two configurations: planar and hanged.

In the sequence mandala0, the kerchief remains rigid on the floor. In mandala1, the deformation had an amplitude of 15 cm and a period of 2 s. In mandala2, the amplitude is 10 cm and the period 1 s. In the mandala3, the amplitude is 25 cm and the kerchief oscillates with a period of 2 s. In the mandala4, the amplitude is 30 cm, and its period is halved to 1 s.

[2] https://www.youtube.com/watch?v=6mmhD2_t6Gs&t=43s

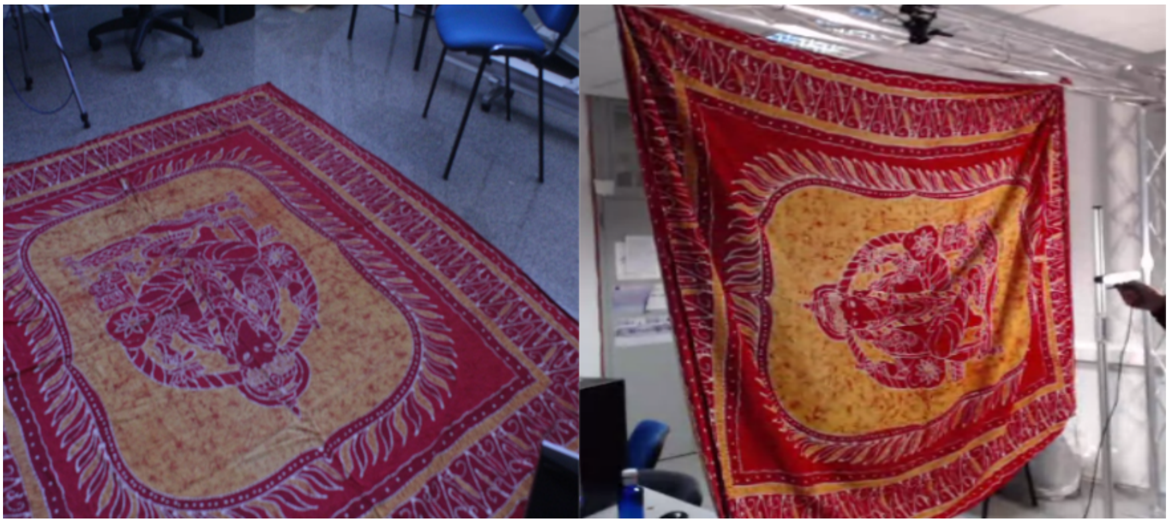

Fig. 3.10 Two configurations of Mandala dataset: rigid planar (Mandala0), and hanged in the rest of the sequences.

Overall quality experiment

We analyze the overall quality of the estimated map. Figure 3.8 shows the final results along the five sequences for DefSLAM in green, and ORBSLAM in blue.

In rigid mandala0, DefSLAM obtains a similar 3D RMS error to ORBSLAM. Concerning the fraction of matched map points, both DefSLAM and ORBSLAM got a high percentage which means that the map points that are highly reused, due to the rigidity of the scene.

In mandala1 and mandala2, the kerchief has low frequency and amplitude deformation. DefSLAM obtains a similar 3D RMS error to the one obtained in mandala0 for both sequences, being able to recover the deformation of the kerchief. ORBSLAM could process the entire sequences, but its 3D RMS error was highly penalized by the deformation, triplicating the error obtained in the mandala0 sequence, and the RMS error of DefSLAM. DefSLAM could recover more accurately the deformation of the scene observed during the sequence both in terms of RMS error and in fraction of matched map points per frame.

In the mandala3 and mandala4 sequences, the conditions are more extreme. ORBSLAM could not process any of these sequences entirely. In this sequences, the fast deformation yields difficulties for DefSLAM which experiments some delay to converge the correct shape. This provoked some peaks in the RMS error. In any case, the error average was around the 4 cm during both sequences. In Fig. 3.9 we can observe the quality of the reconstruction of the local deformations in the sequence mandala3. The fraction of matched map points for DefSLAM was also smaller. Supplementary material includes a video with fragments of the mandala dataset quality results.

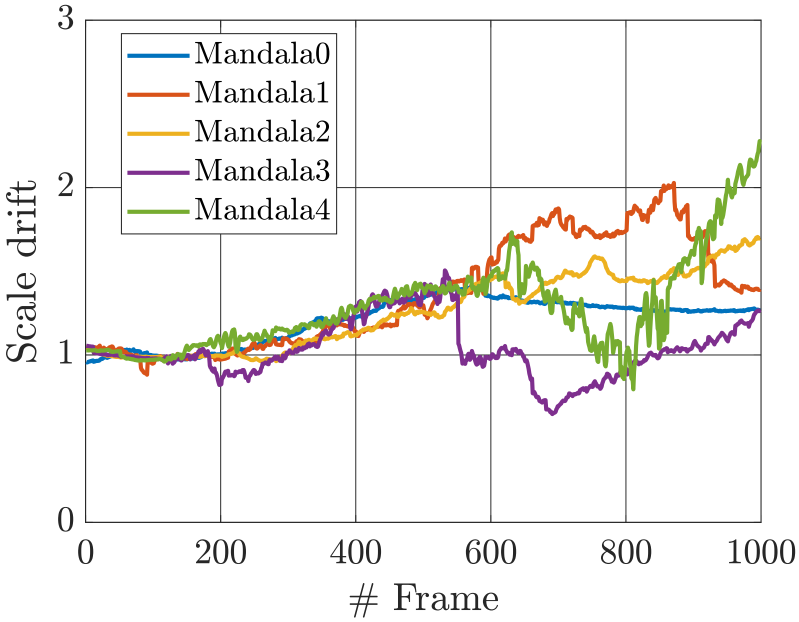

Fig. 3.11 Scale drift along the Mandala sequences. It increases more with more challenging. It is reduced in case of re-observation.

Scale drift analysis

The previous section RMSE focuses on the up-to-scale shape accuracy. Fig. 3.11 shows the scale drift along the different sequences. The main source of scale drift is the alignment (Sec. 3.5.3), where to estimate the scaled template, we align the reference up-to-scale template with the previous reference scaled template. This makes the scale accumulate the misalignment between the new and the old template. The scale drift is close to null in the mandala 0 and increases to higher values to peak the deformation becomes more challenging. Eventually the scale drift can be reduced due to re-observations of the map during the sequence.

Sensitivity Analysis

All the experiments reported, both in the Mandala dataset and Hamlyn, were run with $\lambda_s = 16000$, $\lambda_b = 300$ and $\lambda_r = 0.02$ as standard tuning.

To better understand the role of the weights, we varied their values to study their effect in the final 3D RMS error and scale drift in the challenging mandala3. We run the entire sequence and evaluated the RMS error at the end of the sequence from frames # 800 to # 1000. The error is not servery affected, remaining between 20 and 40 m, for a range of values from $\lambda_s = [1600, 100000]$, $\lambda_b = [100, 1000]$, $\lambda_r = [0, 0.1]$. By decreasing the λ_s and λ_b values, the system becomes unconstrained and fails in process the entire sequence. By increasing λ_s and

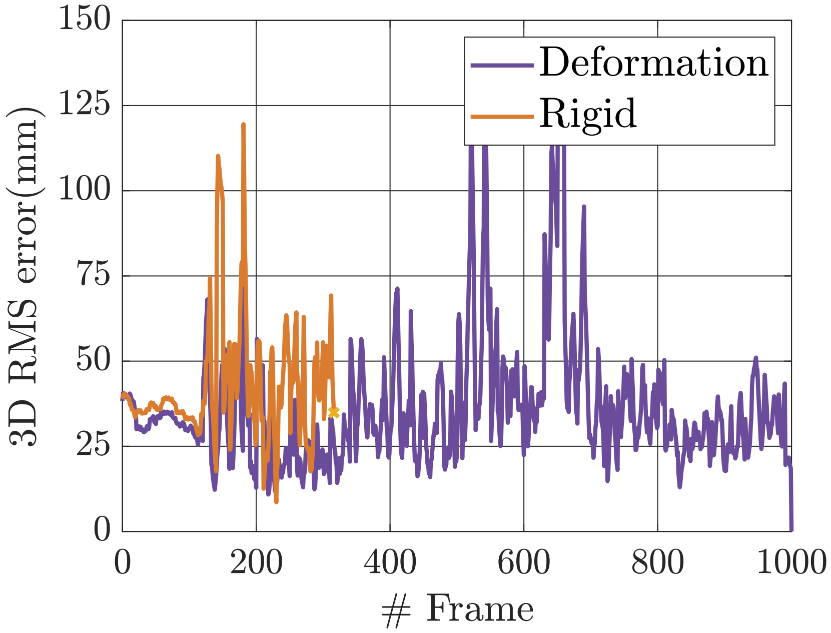

Fig. 3.12 Rigid tracking vs deformation tracking surface error as 3D RMS scene reconstruction error per frame in mm.

λ_b, the system assumes rigidity thus causing another failing scenario. Figure 3.12 shows the extreme case of a perfectly rigid and fixed template compared with our standard tuning. It can be seen how a rigid template for tracking fails to survive strong scene deformations. This case correspond to high values for the three coeficients λ_s, λ_b and λ_r.

The reference regularizer has proven critical to reduce the scale drift specially in the Hamlyn SeqHeart sequence where the camera is imaging constantly the same zone and observing the entire template with few boundary point constraints (Sec. 3.7.2), from 36 % for λ_r to 2 % for $\lambda_r = 0.02$

Deformation mapping normal estimation accuracy

We analyze the quality of the deformation mapping for sequence mandala3 focusing in the angle error between the estimated normal and the ground truth normal, in the two stages of the normal estimation, the initial NRSfM and the subsequent SfN (Sec. 3.5.2). Figure 3.13 shows the RMS angle error of the shape estimated by the NRSfM versus the error after the SfN stage. SfN consistently reduces the error through the entire sequence improving the normals. Averaging the error for all the keyframes in the sequence the SfN achieves a 15 deg RMSE versus the 22 deg of the NRSfM.

The output of the NRSfM is the set of surface normals for each map point in the reference keyframe. The normal of a map point is reestimated after each re-observation of that point in

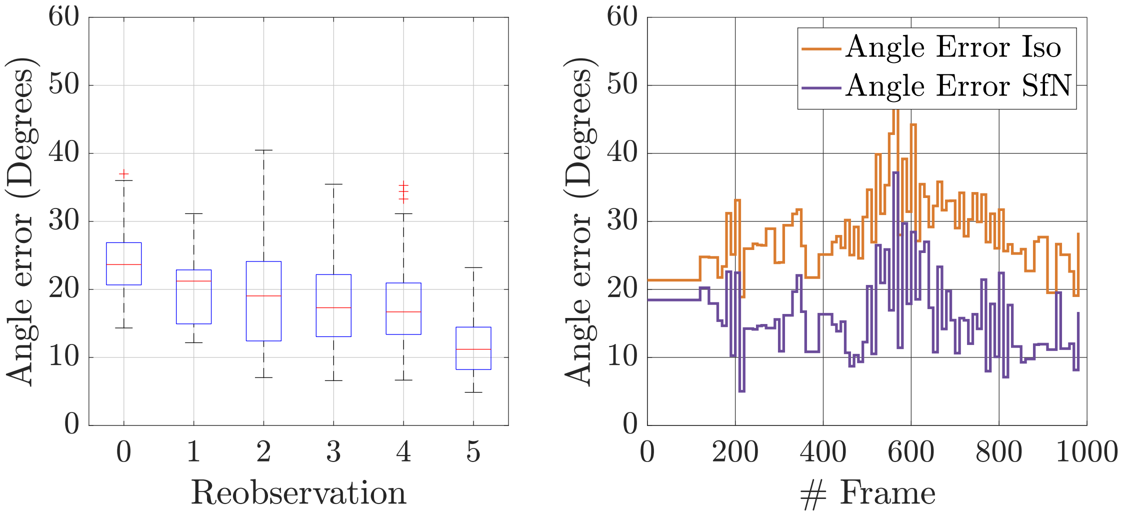

Fig. 3.13 (Left) Box-and-whisker plot for the normals angle error in a keyframe after SfN, improvement as a function of the keyframe resobservations. (Right) per keyframe RMSE angle error for the normal orientation after NRSfM and after SfN.

a new keyframe. Figure 3.13 shows the evolution of the RMS angle error for the normals in a keyframe along 5 re-observations after its creation. We can see how the median error goes from 23 degrees at initialization down to 12 deg after the 5th re-observation.

3.7.2 Hamlyn Dataset

Our last experiments test DefSLAM in six intracorporeal sequences from the *Hamlyn dataset* (Mountney et al., 2010a, Stoyanov et al., 2005) to evaluate our algorithm in medical images. The first two sequences are recorded with a ex-vivo phantom heart (Stoyanov et al., 2010) syncronised with a CT scanner to register ground truth. In addition, we processed four in-vivo laparoscopic sequences (See Fig. 3.15): 1) SeqAbdomen (Dataset 1) is an exploration of the abdominal wall where the scene remains almost rigid (Fig. 3.15 Bottom left). 2) SeqExploration (Dataset 20) performs an exploration around the exterior of the bowel with low texture. it has a small deformation at the beginning (Fig. 3.15 Bottom right). 3) SeqHeart (Dataset 4) (Stoyanov et al., 2005) is a non-rigid beating heart observed by a fixed camera. 4) SeqOrgans (Dataset 19) is an abdominal exploration and deformation of the scene due to tool interfering (Fig. 3.15 Top right).

The closest SLAM system to ours reporting accuracy *wrt.* an external sensor in medical sequences is MISSLAM (Song et al., 2018). We evaluate our system in the same sequences, i.e. the ex-vivo phantom heart sequences. Despite the lack of camera motion, the scenes have enough deformation for DefSLAM to reconstruct them. We report a mean accuracy of

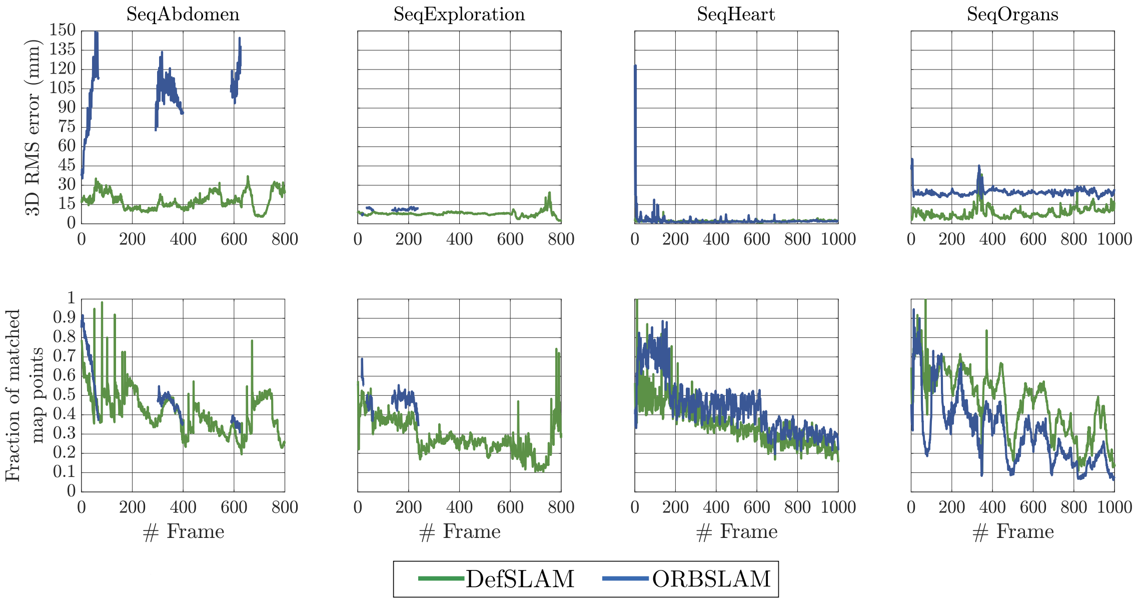

Fig. 3.14 Processing Hamlyn sequences. Green DefSLAM, blue ORBSLAM. From left to right: Heart, organs, abdomen and exploration sequences. Per frame RMS scene reconstruction error in mm after a per frame scale alignment with the stereo ground truth.

3 and 4 mm in the sequence phantom5 and phantom7, respectively. The average accuracy MISSLAM (Song et al., 2018) as reported by the authors is 0.28 mm and 0.35 mm. Concerning the execution time, we report a similar runtime per frame, but DefSLAM runs in CPU unlike MISSLAM that uses GPU. It has to be noted that they use stereo input in contrast with DefSLAM which is a purely monocular method.

Fig. 3.14 reports the median of 5 executions RMS error during the four in-vivo Hamlyn sequences and Fig. 3.16 shows its corresponding scale drift. As it happened with the Mandala dataset (Sec. 3.7.1, the scale drift got slighly increased for the more challenging sequence. In the sequence where the camera remains in the same zone there is no scale drift.

DefSLAM is able to process SeqAbdomen and SeqExploration entirely with a mean 3D RMS error of 17 mm and 10 mm respectively. In these scenes, the camera explore but it come back to the same zone. DefSLAM was able to re-observe part if the map already built and thus reduced the scale drift. ORBSLAM performed poorly in this sequences and could not process them entirely.

In SeqHeart, the camera is practically static, but DefSLAM was able to initialize with the monocular strategy proposed even with a short parallax. The 3D RMS error was approximately 3 mm, equal to the ex-vivo phantom result with a much better groundtruth.

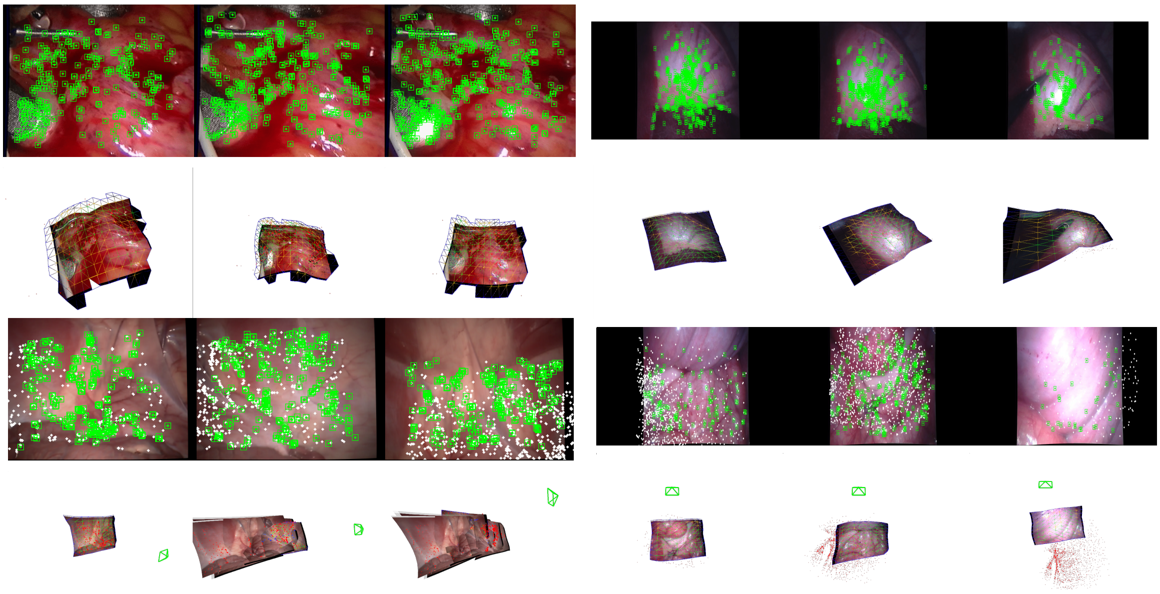

Fig. 3.15 DefSLAM in in-vivo Hamlyn dataset sequences. 3 typical 2D images an the corresponding 3D maps. (Top left) Heart sequence. (Top right) Organs Sequence. (Bottom left) Abdominal sequence. (Bottom right) Exploration sequence.

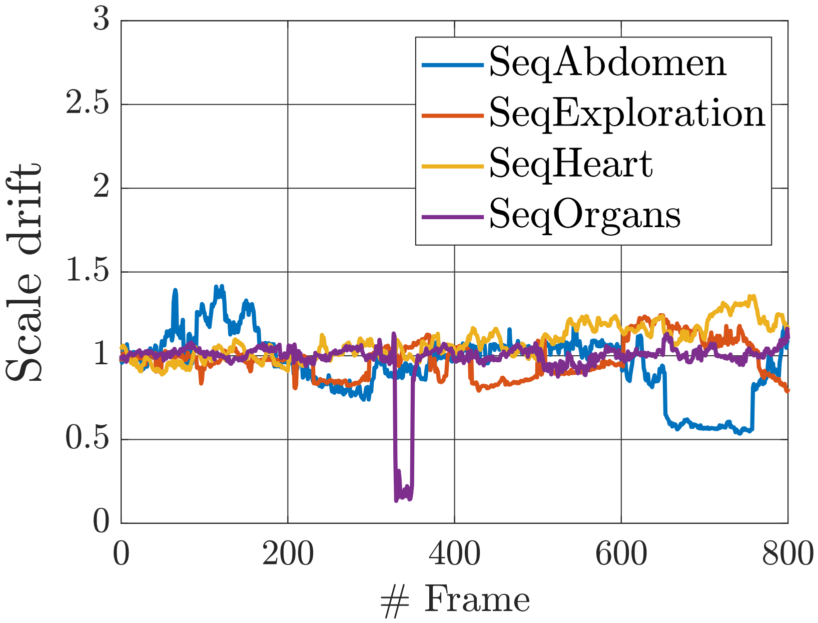

Fig. 3.16 Scale drift along the Hamlyn dataset sequences.

ORBSLAM initializated with the ground truth was able to process the entire sequence with an error of 5 mm

Finally in the sequence SeqOrgans, DefSLAM shows its ability to perform the reconstruction of a deformable scene in exploratory sequence with an accuracy of 8 mm. It survives to the tool clutter that cover almost entirely the image, correcting the scale drift. In the end of the sequence, the tool deforms the organ imaged and DefSLAM was able to recover the deformation of the scene with the same error than in the rest of the sequence.

Fig. 3.15 shows the overall quality of the 3D reconstruction of the medical sequences. Supplementary material includes the video with the results in all the sequences.

3.8 Conclusions

We have formulated DefSLAM, the first deformable SLAM able to process monocular sequences. We have proposed to split the computation of DefSLAM in two parallel threads. The deformation tracking thread is devoted to estimating the camera pose and the deformation of the scene, it is based on SfT. SfT needs a prior of the geometry of the scene encoded in the template. When exploring new zones, our method estimates new templates to cover new areas. Our second thread, the deformation mapping, is devoted to periodically re-estimating the template to better adapt it to the currently observed scene. Both SfT and NRSfM model the cameras as perspective, hence the system is able to handle close-ups typical in scene exploration where perspective effects are prevalent.

Our experiments confirm that the proposed method is able to handle real exploratory trajectories of a deforming scene. Direct comparison with other systems is not possible, we have focused the comparison with the rigid monocular ORBSLAM after its re-tuning to handle non-rigid scenes. This comparison proves that DefSLAM is able to robustly initialize from monocular sequences, continuously adapt the map to the scene deformation, and producing accurate scene estimates.

We have also shown in preliminary experiments that the system is able to handle medical endoscopy images. In the next Chapters, we will adapt this system for medical imagery to handle all kinds of challenges not taken into account in the present work, *i.e.* uneven illumination, poor visual texture, non-isometric deformations or ultra close-up shots exploring the endoluminal cavities.

Another future work is to develop a full-fledged mapping system including multiple maps, relocalization, loop closure or long term place recognition to achieve robust performance for extended periods of time or multiple moving and deforming bodies.

Chapter 4

SD-DefSLAM: Semi-Direct Monocular SLAM for Deformable and In-Body Scenes

Conventional SLAM techniques strongly rely on scene rigidity to solve data association, ignoring dynamic parts of the scene. In this work we present Semi-Direct DefSLAM (SD-DefSLAM), a novel monocular deformable SLAM method able to map highly deforming environments, built on top of the method proposed in the Chapter 3, DefSLAM. To robustly solve data association in challenging deforming scenes, SD-DefSLAM combines direct and indirect methods: an enhanced illumination-invariant Lucas-Kanade tracker for data association, geometric Bundle Adjustment for pose and deformable map estimation, and bag-of-words based on feature descriptors for camera relocation. Dynamic objects are detected and segmented-out using a CNN trained for the specific application domain.

We thoroughly evaluate our system in two public datasets. The mandala dataset is a SLAM benchmark with increasingly aggressive deformations. The Hamlyn dataset contains intracorporeal sequences that pose serious real-life challenges beyond deformation like weak texture, specular reflections, surgical tools and occlusions. Our results show that SD-DefSLAM outperforms DefSLAM in point tracking, reconstruction accuracy and scale drift thanks to the improvement in all the data association steps, being the first system able to robustly perform SLAM inside the human body.

4.1 Introduction

Simultaneous Localization and Mapping (SLAM) and Visual Odometry (VO) are fundamental blocks for many applications like autonomous robots or augmented reality. Existing methods can be classified as indirect or direct depending of the manner they perform data association. On the one hand, indirect methods estimate 3D geometry from a set of matched keypoints along covisible images, minimizing a geometric error. On the other hand, direct methods avoid extracting features, and work directly on pixel intensities to estimate the 3D geometry, optimizing a photometric error. Finally, semi-direct methods extract features and combine both types of errors.

However, regardless of that classification, all methods rely on a simple, yet important assumption: scene rigidity. This assumption greatly simplifies the SLAM and VO problem and perfectly models many of their application domains. Nevertheless, the increasing interest in Minimally Invasive Surgery (MIS) and medical robots has placed in the spotlight the rigidity assumption, as these kinds of applications work on highly deforming scenarios. That is why a new classification arises as rigid and non-rigid methods, the latter assuming that the 3D position of triangulated landmarks can vary over time.

In this work, building on DefSLAM, we propose SD-DefSLAM, the first deformable semi-direct SLAM system, able to robustly process sequences under great deformations and weak texture, as it is the case of MIS videos. SD-DefSLAM is semi-direct as it extracts ORB features and uses an illumination-invariant Lukas-Kanade (LK) Lucas and Kanade (1981) optical flow algorithm to perform data association, minimizing a photometric error, while the camera pose and deforming 3D geometry is estimated minimizing the geometric error (Fig. 4.1).

In non-rigid SLAM, dynamic objects are difficult to separate from the deforming background using conventional techniques. To achieve robustness, we mask-out moving objects with the help of a convolutional neural network (CNN) specifically trained to segment surgical tools. Finally, we include relocalization capabilities for which we perform long-term data association with ORB descriptors (Rublee et al., 2011) and a bag of words (Gálvez-López and Tardós, 2012), achieving robustness to camera occlusions.

4.2 Related Work

4.2.1 Rigid SLAM and VO

The first real-time SLAM systems followed the indirect approach. MonoSLAM (Davison et al., 2007) matches a set of sparse keypoints and recovers the scene geometry in an EFK-

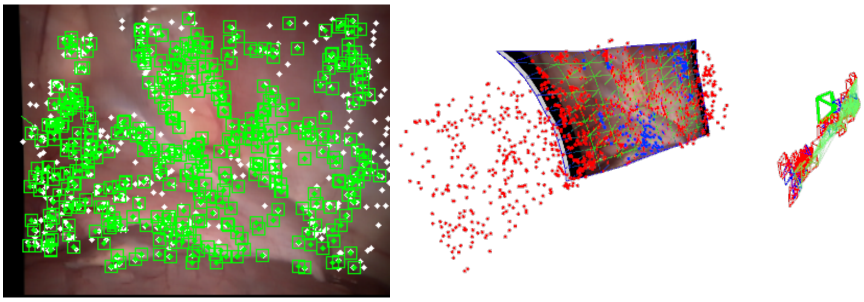

Fig. 4.1 SD-DefSLAM working on Dataset1 of Hamlyn dataset. Left: features tracked in the endoscopic image using photometric techniques. Right: camera motion and growing deformable map estimated by minimizing geometric error.

based framework. This work was later extended by Civera et al. (2008), using an inverse depth parametrization. Later PTAM (Klein and Murray, 2007) proposed a parallelization of the main tasks of an SLAM system to allow a Bundle Adjustment (BA) scheme to optimize the 3D geometry. ORB-SLAM (Mur-Artal et al., 2015) is currently the reference system among indirect methods by using the combination of FAST-ORB feature-descriptor (Rublee et al., 2011) and BA to optimize the 3D information. In its successive versions (Mur-Artal and Tardós, 2017, Campos et al., 2021) it is extended to different type of sensors, ranging from stereo cameras to wide-lens to inertial sensors.

As for direct methods, DSO (Engel et al., 2017) is the first fully direct VO algorithm that jointly optimizes structure and motion with photometric BA. This work is later extended in DSM (Zubizarreta et al., 2020) by building a direct SLAM algorithm that uses the same photometric model of DSO. While current direct methods are more robust in weakly textured areas, their accuracy degrades in presence of geometric distortions, and they assume photometric invariance, being only able to adapt to global illumination changes (Engel et al., 2017). So, they are not applicable in endoscopic images where strong deformations and local illumination changes are prevalent.

Our work is more similar to SVO (Forster et al., 2014) that proposed an hybrid approach combining direct and indirect methods. SVO is a semi-direct VO method that extracts features in keyframes, uses photometric techniques to perform short-term data association, and ultimately optimizes the reprojection error in a BA.

The crucial novelty of our method is the use of per-feature illumination-invariant photometric data association, instead of the global image alignment used by DSO and SVO, that cannot handle deforming scenes. Our method also allows to obtain medium-term photometric data associations, improving reconstruction accuracy.

4.2.2 Deformable SLAM and VO

Many deformable SLAM and VO systems were developed from rigid ones, aiming in many cases to process intracorporeal sequences, as it is a naturally deforming environment of high practical interest for which several datasets exist (Stoyanov et al., 2010, Pratt et al., 2010, Stoyanov et al., 2005, Mountney et al., 2010b). The first systems that processed this kind of images were proposed by Grasa et al. and Lin et al. (2013), both making use of conventional feature-based SLAM and threshold strategies to differentiate between rigid and non rigid points. Later, ORBSLAM was tuned by Mahmoud et al. (2016) and Mahmoud et al. (2017) to be able to localize in MIS sequences. The seminal work DefSLAM (Lamarca et al., 2020) is the first indirect monocular SLAM system able to tackle with exploration in deformable scenarios. The system grows the map using a sequential Non-Rigid Structure-from-Motion (NRSfM) algorithm based on the work proposed by Parashar et al. (2017), and estimates at frame rate the deformation occurred and the camera pose by means of a Shape-from-Template (SfT) algorithm (Lamarca and Montiel, 2018a). DefSLAM has been proved to work in some simple medical sequences, but the presence of typical challenges like poor texture, illumination changes and tools intrusion, make it fail.

This evidences the need of more robust data-association methods to process highly deforming environments. In endoscopic sequences this is usually done by correlation matching in consecutive images (Grasa et al.), (Lin et al., 2013), as feature matching using descriptors such as ORB (Rublee et al., 2011) or SIFT (Lowe, 2004) usually do not perform well in low texture regions. AKAZE, proposed by Alcantarilla et al. (2011), is a feature designed to preserve the low texture gradient in the multiscale detector, performing especially well for intracorporeal images. However, it is too slow to be applied in a real-time SLAM algorithm. Du et al. (2015) proposed a deformable Lucas-Kanade (Baker and Matthews, 2004) implementation is proposed for tracking tissue surfaces in non-exploratory sequences, including a term that controls the deformation. Deep learning techniques can also play an important role as shown by Liu et al. (2020) in which they train a CNN to get dense descriptors in a sinus endoscopy dataset.

Finally, as deformable sequences pose a big challenge for SLAM and VO algorithms, it is essential a better understanding of the scene, identifying and removing dynamic objects that could degrade performance. DynaSLAM (Bescos et al., 2018) uses CNNs to detect, remove and inpaint potentialy dynamic objects such as persons or cars. DOT (Ballester et al., 2020) follows up the ideas from DynaSLAM to only mask-out objects that are actually moving. In the case of endoscopic images, the most typical dynamic objects are surgical tool. Segmentation of this kind of objects is of interest to the scientific community and several

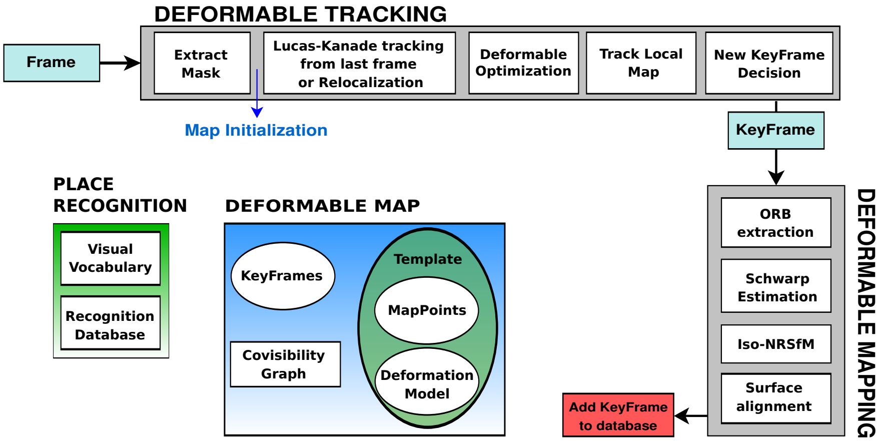

Fig. 4.2 SD-DefSLAM scheme with a tracking and a mapping thread running concurrently. The main novelties are in the tracking thread, that masks surgical tools using a CNN, achieves robustness with an illumination-invariant photometric method that tracks the previous frame and the local map, and includes bag-of-words relocalization and a new regularizer that smooths camera motion.

methods (Laina et al., 2017, Kurmann et al., 2017, Pakhomov et al., 2019) have arisen as response.

4.3 Semi-Direct DefSLAM

Our approach is called Semi-Direct Deformable SLAM (SD-DefSLAM) as it performs short-term and medium-term data association Campos et al. (2021) using a photometric method (subsection 4.3.1) while the deformable optimization backend (subsection 4.3.2) optimizes a geometric error. A global overview of SD-DefSLAM is depicted in Fig. 4.2. It uses two threads, one for deformable mapping, that progressively builds a growing deformable map and other for deformable tracking, that estimates camera pose and map deformation for each frame processed. Although the main novelties with respect to DefSLAM are in the deformable tracking thread, for the reader convenience, we present here a brief summary of the whole system.

The map is formed by a set of *reference keyframes*, that have observed new parts of the scene as exploration progresses, with an associated surface template. Each template models the observed surface with a triangular mesh that represents its shape-at-rest, whose vertices are the 3D map points. The map also contains a set of *refining keyframes* that are used to refine the templates. Templates are created and refined by the deformable mapping thread at keyframe rate, and their deformation model is estimated by the deformable tracking thread at frame rate. Keyframes are added to a place recognition database Gálvez-López and Tardós (2012) to enable relocation after occlusions.

The deformation mapping thread estimates the surface observed in the reference keyframes and uses refining keyframes to improve this estimation incrementally. Templates are created to grow the map when exploring new places. The core of the deformation mapping is a Non-Rigid Structure-from-Motion (NRSfM) algorithm based on isometry and infinitesimal planarity Parashar et al. (2017). It estimates the normal of the points of a keyframe. The points are initialized assuming smoothness in the surface with respect to the rest of normals estimated and they are refined with each new observation. After estimating the normals, a shape-from-normals algorithm estimates a proportional shape of the surface that fits with those normals. Finally, it performs a SE(3) alignment to recover the correct scale with respect to the rest of the map. This new surface becomes the template for the deformation tracking.

The deformation tracking thread estimates the localization of the camera and the deformation of the 3D map surface at frame rate. The map surface is coded by means of its shape-at-rest and a deformation model. The input of the deformation tracking is the last pose of the camera, the last deformation of the template and the new frame. We use a LK tracker to get initial putative matches, that are computed independently for each point. With the putative matches we estimate a initial deformation of the mesh. This optimization is robust to outliers and give us a better estimation of the position of the points. With these new estimates we reinitialize the LK tracker and search for map points in the observed zone. This allow matches with larger baselines than with a standard LK tracker. In case of tracking lost, we have design a relocalization module (subsection 4.3.3) able to relocate the system in this map. For our final application, we have incorporated a CNN that segments tools (subsection 4.3.4) to remove matches in dynamic non-modeled objects.

4.3.1 Data Association

For data association, indirect methods rely on good texture to obtain distinctive features, a RANSAC step to enforce rigidity of the set of matchings found, and robust costs functions in BA to reduce the impact of the remaining outliers. In contrast, direct methods use global image alignment that can use pixels with lower texture but rely even more strongly in scene rigidity. In this section we present a photometric data association method that works reliably in low-textured areas, without relying neither in illumination constancy, not in scene rigidity. For this, we use an enhanced Lucas-Kanade (LK) algorithm to perform short-term data association among all the images in the sequence. Our LK algorithm allows us to track low textured surfaces with subpixel accuracy even though there have been local changes in lighting. Next, we describe the basic LK algorithm to better explain the improvements performed to increase accuracy and robustness.

Basic Lucas-Kanade algorithm

Let be I and J the reference and the current grayscaled images respectively, $\mathbf{u} = (x, y)^T$ a generic image point found in I and $P(\mathbf{u})$ a squared patch centered on $\mathbf{u}$ of size $(2\omega_x + 1) \times (2\omega_y + 1)$ pixels. The goal of LK algorithm is to find the optical flow vector $\mathbf{d} = (d_x, d_y)^t$ such us $I(P(\mathbf{u}))$ and $J(P(\mathbf{u}+\mathbf{d}))$ are similar. This is solved using Gauss-Newton gradient descent non-linear optimization:

$$\arg\min_{\mathbf{d}} \sum_{\mathbf{x} \in P(\mathbf{u})} (I(\mathbf{x}) - J(\mathbf{x}+\mathbf{d}))^2 \tag{4.1}$$

Note that the goal function depends directly on the gray values of both images and the size of the patch ω_x, ω_y.

Enhanced Lucas-Kanade algorithm

The basic LK optimization (Eq. 4.1) depends directly on the raw intensity values of I and J, which makes the LK algorithm very sensitive to illumination changes. While some direct methods address this issue with a global illumination compensation Engel et al. (2017), we solve it in a more flexible way using local illumination compensation. In other words, we compute a gain factor α and a bias value β per each tracked patch, which are added in the optimization:

$$\arg\min_{\mathbf{d},\alpha,\beta} \sum_{\mathbf{x} \in P(\mathbf{u})} (I(\mathbf{x}) - \alpha J(\mathbf{x}+\mathbf{d}) - \beta)^2 \tag{4.2}$$

This is especially important when light changes do not occur uniformly across the image, as it happens in outdoor scenes in a cloud-and-clear day, in autonomous car sequences taken during the night, or crucially in endoscopic sequences where the light sources are attached to the endoscope, brightening the image in the areas that get approached, while other areas get darkened. In this cases, global illumination compensation would produce very poor results.

It is also important to keep in mind that the LK algorithm needs the initial guess for $\mathbf{d}$ to be close to the solution in order to converge. That means that if the point to be tracked suffers a big displacement in pixels (for example, due to camera motion or strong deformations) between images, LK may display poor convergence. This can be solved by taking the pyramidal approach proposed in Bouguet (2001) in which the algorithm estimates the optical flow along a pyramidal representation of I and J from the coarsest to the finest level.

Moreover, as we have geometrical estimates of the 3D scene surface and camera poses provided by the SLAM, we can compute an initial guess for $\mathbf{d}$ using that information to improve convergence. For that purpose, assuming local planarity around each tracked point, we can further compute a homography ($\mathbf{h}$) per point that synthesizes the shape of the patch in the new image, yielding the following error term:

$$\underset{\mathbf{d},\alpha,\beta}{\arg\min} \sum_{\mathbf{x}\in P(\mathbf{u})} \left(I(\mathbf{x}) - \alpha J\left(\mathbf{h}(\mathbf{x}) + \mathbf{d}\right) - \beta\right)^2 \tag{4.3}$$

Transformation defined by $\mathbf{h}$ compensates any rotation or scale change that the patch could have suffered, making our enhanced LK algorithm rotation and scale invariant. It is also essential to note that computations to synthesize the patch use bilinear interpolation to achieve subpixel accuracy. Now the algorithm guesses for $\mathbf{d}$, can be safely set to 0 because most of the flow is estimated from the available geometry.

Finally, even though LK algorithm converges, it is not guaranteed that it has converged to the correct solution. This can produce spurious feature tracks that negatively affect the overall robustness and accuracy of the algorithm. Most systems address this issue imposing scene rigidity, either in a RANSAC step or with global image alignment. In our case we detect and discard most outliers computing the *Structural Similarity Index* (SSIM) Wang et al. (2004) between the reference and the tracked patches. The remaining outliers are successfully handled by a robust influence function in the deformable optimization.

4.3.2 Deformable optimization

Despite our LK algorithm is able to track low textured surfaces in the presence of deformation using photometric error, the innovation between the reprojected map points and their position in the image would be so high that they will be considered as outliers in a pure camera pose optimization. Instead, the tracking thread estimates simultaneously the camera pose and the surface deformation minimizing the geometric reprojection error. This dualism leads to the semi-direct name of our algorithm.

More precisely, our deformable tracking thread performs a two-step optimization (Fig. 4.3) designed to increase SLAM accuracy by reusing the map. For that purpose, as the camera performs exploration, we compute a local map around the current camera pose with covisible keyframes.

The first step aims to compute a first coarse estimation $\mathbf{T}_{cw}^{t}$ for the camera pose. It obtains putative matches for the points in the previous image using the LK tracker with no

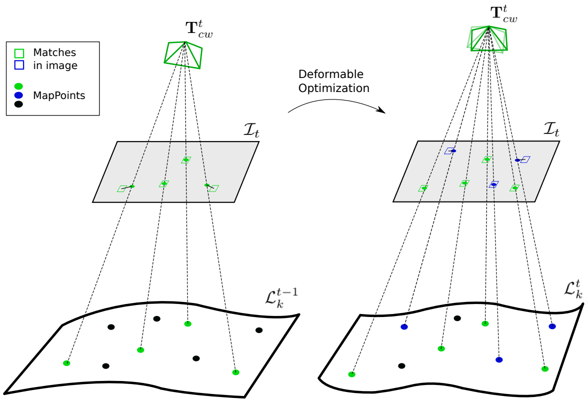

Fig. 4.3 Two-step optimization. First, points from the previous image are tracked using photometric error, getting some matches (green squares) associated to map points (green dots), and we perform a deformable optimization that improves both the camera pose $\mathbf{T}^t_{cw}$ and the local map $\mathcal{L}_k^{t-1}$, reducing the geometric error. Then, we project points from the local map $\mathcal{L}_k^t$ into the current image, track them them with our enhanced LK method and perform a final deformable optimization. The reobserved map points are marked in blue.

geometric information (eq. 4.2) and runs a first deformable pose optimization. With this early optimization, we also compute the local map $\mathcal{M}$ for the next step.

With the computed camera pose $\mathbf{T}^t_{cw}$ and local map $\mathcal{M}$ from the previous step, we reproject map points from the local map into the current image. Using the projections and the geometrical information from $\mathcal{M}$, we compute an homography $\mathbf{h}$ per projected point and we search its true image position by running our LK tracker with homographies (eq. 4.3). Finally, with the additional matches found, we run a second deformable pose optimization.

Both deformable optimizations estimate the local map $\mathcal{L}_k^t$ deformation at frame t, along with the camera pose $\mathbf{T}^t_{cw}$, using a modified version of the cost function proposed in Lamarca et al. (2020):

$$\begin{aligned} \underset{L_k^t, T_{cw}^t}{arg\,min}\; & \varphi_d(\mathcal{I}^t, \mathbf{T}^t_{cw}, \mathcal{L}_k^t) + \varphi_e(\mathcal{L}_k^t, \mathcal{L}_k^{t-1}, \mathcal{T}_k) \\ & + \varphi_c(\mathbf{T}_{t,t-1}) \end{aligned} \tag{4.4}$$

where $\varphi_d(\mathcal{I}^t, \mathbf{T}^t_{cw}, \mathcal{L}_k^t)$ is the total squared reprojection error weighted with a robust Huber influence function, and $\varphi_e(\mathcal{L}_k^t, \mathcal{L}_k^{t-1}, \mathcal{T}_k)$ is the deformation energy of the template $\mathcal{T}_k^t$ that considers bending and stretching (see Lamarca et al. (2020) for more details).

To smooth camera motion in frames with low number of matches due to occlusions or sudden deformations, we add here a new regularization term:

$$\varphi_c(\mathbf{T}_{t,t-1}) = \xi^T \mathbf{W} \xi \tag{4.5}$$

where $\xi = log\,(\mathbf{T}_{t,t-1})$ encodes the translation and rotation between the current and previous frame in the Lie algebra, and $\mathbf{W}$ is a hand-tuned information matrix that controls the degree of smoothing performed.

4.3.3 Relocalization

The presence of deformations, really low textured areas, or complete occlusions can lead to system failure. In that context, it is of paramount importance to have a procedure that allows tracking recovery. As in ORB-SLAM, the detection of candidate keyframes for relocation uses the bag-of-words (BoW) technique from Gálvez-López and Tardós (2012), building a database with every keyframe in the sequence, converting them into BoW after extracting ORB descriptors. When the system gets lost, we convert the lost frame into BoW and query the recognition database, obtaining some keyframe candidates. For each keyframe, correspondences associated to map points are computed and then, we obtain an initial camera pose with PnP, performing RANSAC iterations. The main difference with the rigid case is that the inlier threshold has been increased to allow points with some deformation. If PnP is successful, we retrieve the template associated with the candidate keyframe and perform a deformable optimization, optimizing both the template and the camera pose. Tracking continues with this retrieved template. Although our method only works under mild deformations, as PnP is constrained by (weakened) rigidity, it is able to successfully solve the typical short-time occlusions appearing in endoscopies.

4.3.4 Moving Objects

In conventional SLAM, moving objects can be successfully detected as their motion is not consistent with the motion of the rest of the scene, except if they move too slowly. However, in a deformable scenario, separating object motion from scene deformation is far from trivial using just geometric information. Matches coming from moving objects lead to severe errors in scene deformation or even to total SLAM failure. We propose to solve this issue using semantic information with a CNN trained to identify and segment the typical moving objects in each application domain, masking the corresponding image regions to avoid matching features in them.

To segment surgical tools in medical scenes we use the CNN defined and trained in Shvets et al. (2018). The network is directly integrated in the system and computes a mask for each

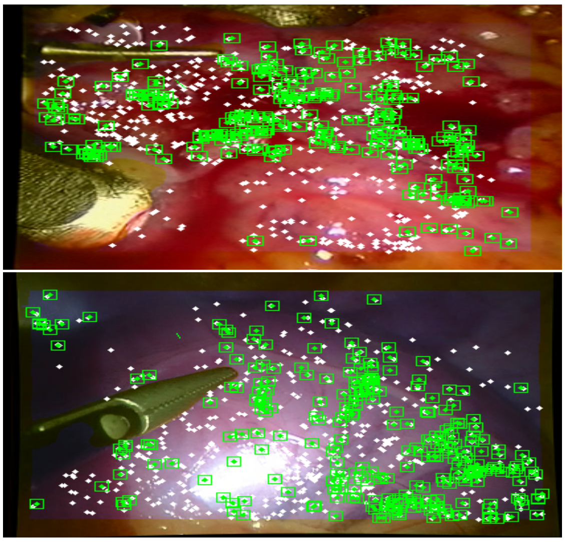

Fig. 4.4 Frames from Hamlin datasets 4 and 19 showing surgical tools, that are successfully detected and masked-out (yellow color) using semantic segmentation with a CNN.

incoming image. The mask is finally dilated to avoid keypoint detection in the borders of tools. In Fig. 4.4, we show examples of the masks obtained in two different sequences.

If the tool occludes large parts of the image, the camera pose estimation will become an ill-conditioned problem. For this reason, we constraint the camera motion with a smooth motion prior. When the occlusion is complete, tracking is lost and the system relies on relocation.

4.4 Experiments

We have evaluated the proposed system and compared it with DefSLAM (Lamarca et al., 2020) in two datasets. The first one is the Mandala dataset created to evaluate deformable SLAM. The purpose of this dataset is to evaluate the performance of the system in a controlled environment with good texture and illumination conditions. Secondly, we further validated our system in several medical sequences of the Hamlyn dataset which pose a substantial challenge to SLAM algorithms. Although our method is pure monocular, in both cases, we use datasets obtained with stereo cameras, to extract a ground truth solution for the scene surface. We analyze the 3D RMS error of the reconstruction, by means of the Euclidean distance between the ground truth and the reconstruction of the system correcting the scale by frame, and reporting the scale drift observed along the trajectory. We also provide a data association quality to compare the performance of the feature matching technique in

Table 4.1 Comparison in Mandala Dataset.

	DefSLAM [Lamarca et al. (2020)]		SD-DefSLAM	
	RMSE (mm)	Scale drift	RMSE (mm)	Scale drift
Mandala0	26.3	1.06	**23.1**	**1.03**
Mandala1	22.3	1.44	**21.3**	**1.32**
Mandala2	17.9	1.46	**16.1**	**1.41**
Mandala3	43.7	2.07	**41.8**	**1.26**
Mandala4	55.6	1.78	**48.1**	**1.27**

Table 4.2 Comparison in Hamlyn Dataset

	DefSLAM [Lamarca et al. (2020)]		SD-DefSLAM	
	RMSE (mm)	Scale drift	RMSE (mm)	Scale drift
f5	5.00	1.01	**3.00**	0.99
f7	4.50	0.99	**4.35**	0.99
Seq_heart	3.84	2.00	**1.17**	**1.32**
Seq_abdominal	23.98	0.98	**22.2**	**1.01**
Seq_organs	13.02	1.27	**6.63**	**1.05**
Seq_exploration	17.02	2.60	**12.56**	**1.36**

DefSLAM with the new semi-direct technique that uses photometric information and gives subpixel accuracy.

4.4.1 Mandala dataset

The Mandala dataset consists of 5 sequences exploring a mandala kerchief that goes from a totally rigid situation (Mandala0) to a intensively deforming one (Mandala4). The kerchief is hanged and deformed creating waves that go through it. The intensity of the deformation is measured depending on the speed and amplitude of the waves.

Table 4.1 shows that SD-DefSLAM outperforms DefSLAM in all Mandala sequences, both in RMS reconstruction error and in scale drift. While in the most rigid sequence (Mandala0) the improvement is marginal, for those sequences with more aggressive deformations (Mandala3 and Mandala4), SD-DefSLAM achieves a significant improvement.

4.4.2 Medical scenes

We have evaluated our system in several laparoscopic scenes of the Hamlyn dataset. This sequences present a huge variety of scenarios, including phantom hearts with CT ground truth (Dataset11-f5 and Dataset12-f7 in Hamlyn (Stoyanov et al., 2010, Pratt et al., 2010)), a non-exploratory heart sequence with tool intrusions (Dataset4 - Sequence_heart in Hamlyn (Stoyanov et al., 2005)) and three exploratory sequences (Dataset1 - Sequence_abdominal, Dataset19 - Sequence_organs and Dataset20 - Sequence_exploration in Hamlyn (Mountney et al., 2010b)).

In general, SD-DefSLAM achieves better RMSE and Scale Drift that DefSLAM in all sequences, as shown in Table 4.2. The improved data association enables our system to better compute the map deformation, improving the RMSE and Scale Drift while the addition of a CNN to mask out surgical tools in the Sequence_heart and Sequence_organs allows our system to robustly process the sequences with significant improvement in the performance. An example of the reconstructed surfaces under deformations is shown in Fig. 4.9.

4.4.3 Data association

The results in the last sections show how SD-DefSLAM outperforms DefSLAM in all the tested datasets. One of the keys for the improvement is the data association which is a fundamental part for both the tracking and the mapping. For the tracking, better association leads to better estimation of the deformations in the map. Concerning the mapping, longer tracks between keyframes speed up the convergence of NRSfM.

In this section, we analyse the proposed data association scheme. There are two key differences *wrt.* the original method. The most evident one is that the matching is performed photometrically, reaching subpixel accuracy. The other one is that each patch is initialized with the keyframes and actively tracked in the consecutive images, removing feature extraction from the matching stage. This is significant as FAST features have low repeatability between temporarily close images, impairing SLAM performance.

Figures 4.5 and 4.6 depict a comparison between the SD-DefSLAM photometric data association (top) and the ORB matching of DefSLAM (bottom) in Mandala3 and Hamlyn Dataset20. In both images, the percentage of matched map points (matched tracks) is shown in blue and the inliers after the deformation optimization (DO inliers) in orange. The true positive are the matched map points considered inliers by the deformable optimization, representing the efficiency of the matching system.

In Mandala3 sequence, SD-DefSLAM doubles the percentage of correct matches obtained by DefSLAM. This greatly improves the overall robustness of the system at the same time that improves the accuracy. Dataset 20 poses a bigger challenge as the combination of low texture and image blurring penalizes both types of data association algorithms, but the new method is still clearly superior. This, together with the subpixel accuracy explains the more accurate reconstruction and smaller scale drift obtained (last row in Table 4.2).

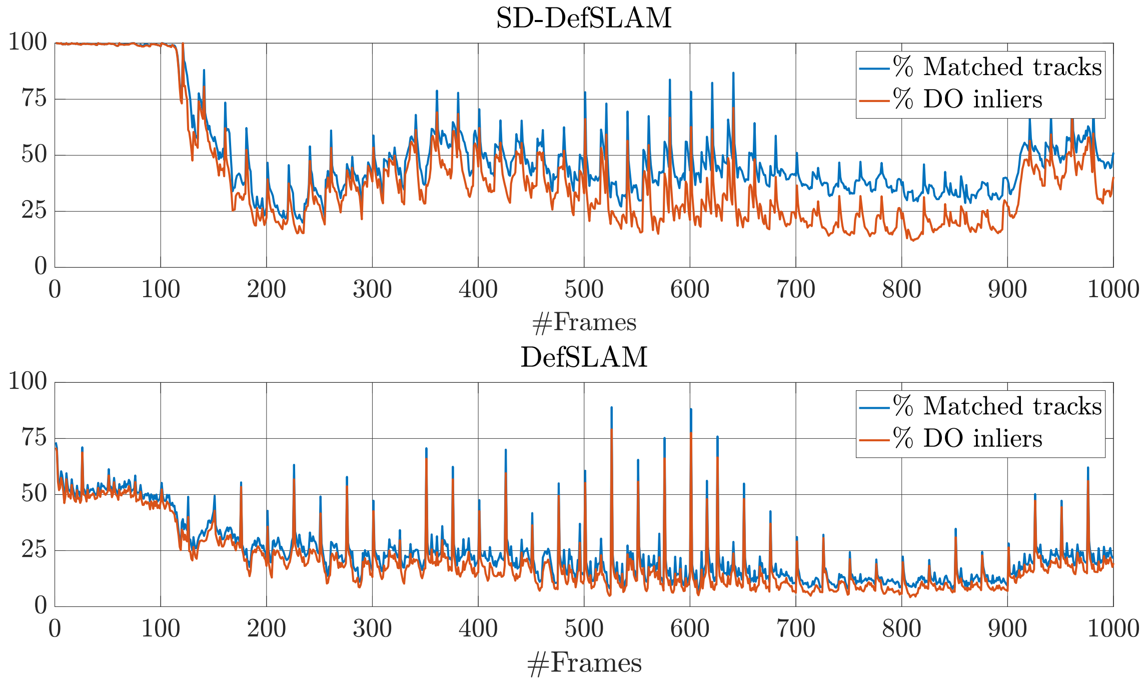

Fig. 4.5 Percentage of points in the local map that are tracked (blue) and that are considered inliers after deformable optimization (orange) in Mandala3 .

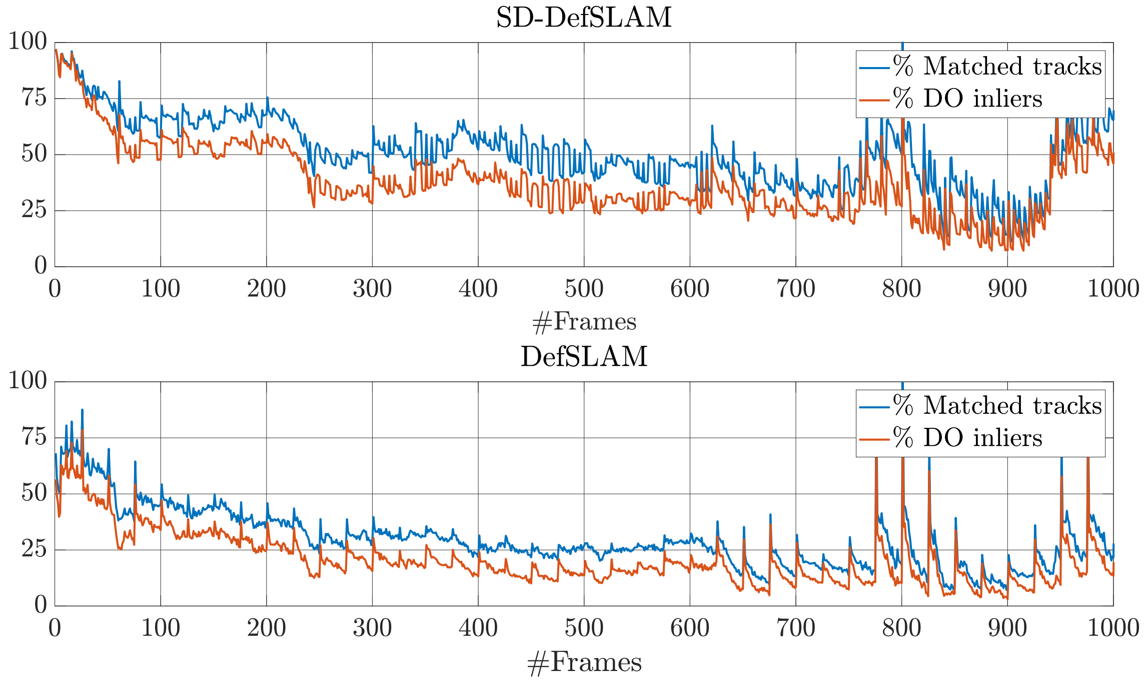

Fig. 4.6 Percentage of points in the local map that are tracked (blue) and that are considered inliers after deformable optimization (orange) in Hamlyn Dataset20.

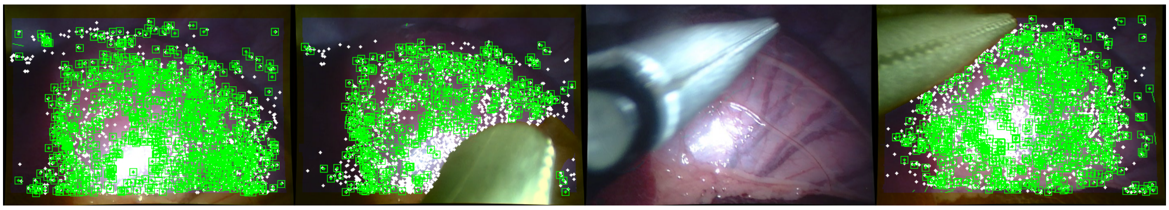

Fig. 4.7 Relocalization in Dataset19. Relocalization due to tool intrusion. The CNN is able to detect the tool correctly, but when it occludes most of the image, the system fails and performs relocalization.

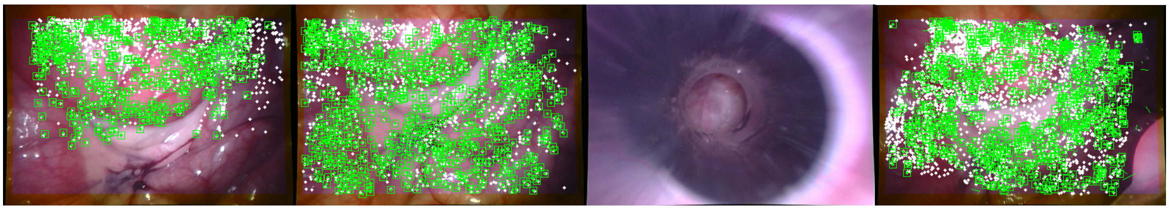

Fig. 4.8 Relocalization in Dataset20. The endoscope is extracted from the scene to clean it, but the system is able to relocate the pose once it is introduced again in the body.

4.4.4 Relocalization

Besides the robustness of the system to tools or low texture, the camera still can get totally occluded or even the endoscope must leave the scene to clean the optics. Thanks to the relocalization module, we were able to relocate the system after a tracking failure. In contrast with DefSLAM, which only cannot manage tracking lost, we were able to process more frames in the proposed sequences, see Fig. 4.7 and Fig. 4.8.

4.5 Conclusions

While rigid SLAM is mature, deformable environments pose serious challenges requiring to re-think all data association steps. We have shown that a semi-direct approach based on per-feature illumination-invariant photometric tracking greatly improves data association, reconstruction accuracy and scale drift. Its combination with CNN segmentation to detect moving objects, and relocalization capabilities to deal with occlusions, gives the first SLAM system able to robustly address the real-life challenges of medical sequences.

Our deformable model assumes isometric deformations. This is quite a restrictive assumption that is not always fulfilled as is the case of MIS sequences. This causes a worsening in the estimation of the deformation which in turn affects the quality of the data association. This can be addressed by exploring new deformation models that properly represent non-isometric deformations.

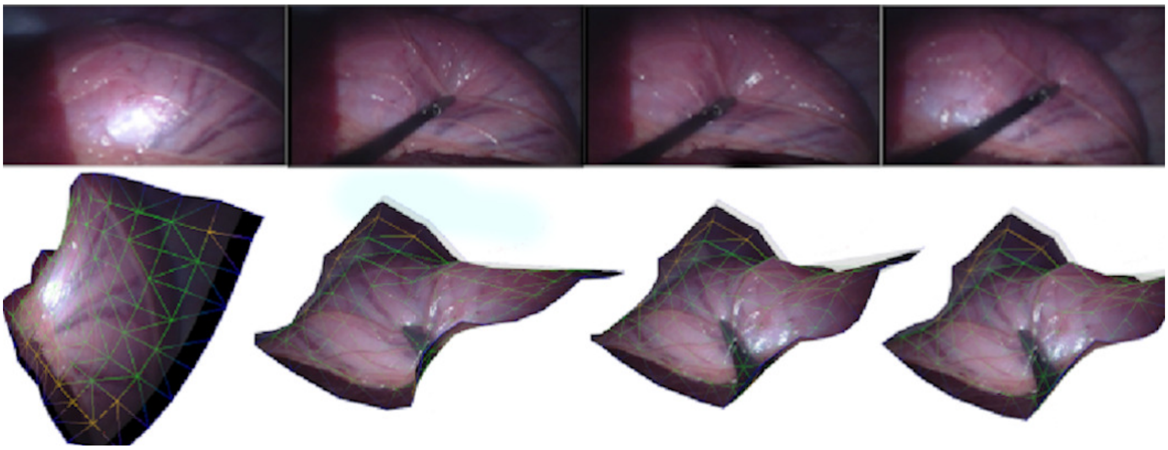

Fig. 4.9 Examples of the reconstructed surfaces in the *Sequence_organs*. Note how we reconstruct the deformation produced by medical tools. Top is the frame inserted, bottom the 3D reconstruction. From right to left: Frames #315, #1010,#1030,#1055

Chapter 5

Direct and Sparse Deformable Tracking

Deformable Monocular SLAM algorithms recover the localization of a camera in an unknown deformable environment. Current approaches use a template-based deformable tracking to recover the camera pose and the deformation of the map. These template-based methods use an underlying global deformation model. In this paper, we introduce a novel deformable camera tracking method with a local deformation model for each point. Each map point is defined as a single textured surfel that moves independently of the other map points. Thanks to a direct photometric error cost function, we can track the position and orientation of the surfel without an explicit global deformation model. In our experiments, we validate the proposed system and observe that our local deformation model estimates more accurately and robustly the targeted deformations of the map in both laboratory-controlled experiments and in-body scenarios undergoing non-isometric deformations, with changing topology or discontinuities.

5.1 Introduction

VSLAM (Simultanenous Localization And Mapping from Visual sensors) is becoming a mature technology to navigate in human-made environments, being crucial for technologies like augmented reality and autonomous robot operation. Current state-of-the-art VSLAM algorithms (Campos et al., 2021, Engel et al., 2017, Zubizarreta et al., 2020) strongly rely on scene rigidity. As a consequence, they perform poorly in deforming scenes, e.g. in medical environments.

Since PTAM (Klein and Murray, 2007), VSLAM algorithms divide the computation in a tracking and a mapping concurrent threads. The tracking thread computes the camera position *wrt.* the map at frame-rate. In parallel, the mapping thread recovers the structure of the scene with a higher computational cost from some selected frames, so-called keyframes.

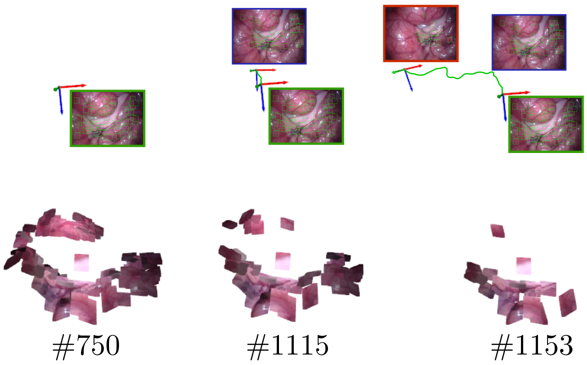

Fig. 5.1 Direct and Sparse Deformable Tracking processing Hamlyn Dataset sequence 6, results after frames #750, #1115 and #1153. Bottom: The map composed of sparse surfels. Top: Camera Trajectory in green.

In the deformable case, both DefSLAM (Lamarca et al., 2020) and SD-DefSLAM (Rodríguez et al., 2020), presented in Chapter 3 and 4, use a deformable mapping based on a Non-Rigid Structure-from-Motion (NRSfM) (Parashar et al., 2017) to recover the structure of the scene at keyframe rate, and a deformable tracking (Lamarca and Montiel, 2018a) that estimates simultaneously the camera pose and the deformation of the map for every frame.

The deformable tracking of these previous methods relies on the usage of a mesh that embeds the map points, and it recovers the most likely shape of the mesh according to a deformation model. This deformation model is global i.e. each map point is connected with their neighbours. This shows excellent performance in scenes with a single surface where all the points are indeed connected. However, when points are not connected, like in scenes with several surfaces, non-isometric surfaces, or with topological changes, the global model does not represent properly the deformation of the map, yielding low performance.

In this paper, we propose a novel deformable tracking method that uses local deformation models to treat the map points as independent bodies. Our first contribution is to model the map as a sparse set of 3D moving textured surfels observed by a moving perspective camera. Each surfel is assumed to have independent rigid displacements from the other surfels around its position at rest. The formulation of the surfel is a first-order Taylor approximation of the map point. The main advantage of this approach is that any smooth surface, e.g. cylinders, planes, spheres or discontinuous surfaces, can be represented locally by a plane, independently of its topology.

Our second contribution is to use a direct photometric error resulting from back-projecting the surfel texture. We jointly optimize the 3D position and orientation of the surface to minimize the direct photometric error. In contrast to previous approaches, in our proposed direct deformable tracking there is no hard data association, instead, the final matching is a byproduct of the photometric alignment.

In our experimental section, we prove that our method can deal with discontinuous surfaces and topological changes, and achieves better performance than the tracking method used in current Deformable Monocular SLAM methods (Lamarca et al., 2020, Rodríguez et al., 2020), obtaining longer tracks with better geometrical accuracy in medical sequences.

Next, in Sec.5.2, we discuss in detail the related works in non-rigid reconstruction and VSLAM. In Sec.5.3, we present our formulation for the surfel. In Sec.5.4, we develop our deformable tracking with fixed camera to prove the potential of surfel tracking adapting to different surfaces. In contrast to the previous methods, we propose a fully direct and sparse approach able to recompute the matches during the optimization. In Sec.5.5, we formulate a world-centric direct deformable tracking to estimate the pose of the camera based on a equilibrium regularizer. Finally, in the last Sec.5.6, the results obtained show a considerable improvement *wrt.* the previous deformable tracking methods both in terms of robustness and accuracy.

5.2 Related work

Deformable SLAM problem consists in reconstructing a map whose shape is constantly deforming and recovering the camera trajectory *wrt.* the reconstructed map.

The first deformable SLAM method proposed was DynamicFusion Newcombe et al. (2015). This method proposes a pipeline where the entire shape of map was reconstructed from partial RGB-D observations from different positions. MISSLAM Song et al. (2018) transferred this technique to medical scenarios by using stereo pairs. Concerning monocular SLAM, the lack of depth information significantly entangles the reconstruction problem. The first work to solve Deformable SLAM with monocular cameras was DefSLAM (Lamarca et al., 2020). Like other monocular SLAM systems (Campos et al., 2021, Engel et al., 2017, Zubizarreta et al., 2020), DefSLAM is composed of two main threads: deformable tracking and mapping. These two components are based on the two main families of non-rigid monocular methods: Non-Rigid Structure-from-Motion for mapping, and template-based techniques for tracking.

The first approaches of NRSfM were formulated using statistical models (Bregler et al., 2000, Dai et al., 2014, Akhter et al., 2011). A low dimensional basis model is used to obtain the configuration of the 3D points for several images. The problem has been formulated with different regularizers, e.g. spatial (Dai et al., 2014, Garg et al., 2013), temporal (Akhter et al., 2011), or spatio-temporal (Gotardo and Martinez, 2011a). The main weakness of these methods is the assumption of orthographic camera model, not suitable for VSLAM due to the noticeable perspective effects in many targeted scenes where close-ups are dominant.

Recent geometric methods have been proved to work with perspective cameras under the assumption of local isometry in the surface (Chhatkuli et al., 2014a, 2016, Parashar et al., 2017, Taylor et al., 2010, Vicente and Agapito, 2012). The method proposed by Parashar et al. (2017) was the base of the deformable mapping proposed in the two previous Chapters due to its ability to naturally handle occlusions and missing data .

Template-based techniques recover the deformation of the scene from a single-image relying in a known textured surface and a deformation model. The 3D shape at rest of the textured surface is the so-called template. In the deformable SLAM approaches, the template is used to estimate the deformation of the map during tracking. The main difference between these methods is the representation of the surface and its deformation model. Among the analytic solutions, one of the most extended assumptions is that the surface is isometric. In other words, the geodesic distance between points in the surface is preserved during the entire sequence. Isometry for shape-from-template –SfT– has been proven to be well-posed and to quickly evolve to stable and real-time analytical solutions (Bartoli et al., 2015, Chhatkuli et al., 2017, Collins and Bartoli, 2010). On the other hand, energy-based methods (Salzmann and Fua, 2011) are numerical approaches that jointly minimize the shape deformation energy *wrt.* the shape-at-rest and the reprojection error for the current image correspondences. These optimization methods are well suited to implement sequential data association with robust kernels to deal with outliers.

The mentioned methods consider the camera static and usually reconstruct small objects that move in the camera field of view. The deformable tracking methods estimate the camera pose in addition to the deformation of the map. Usually, this is done by constraining the problem with boundary conditions (Agudo et al., 2014, Lamarca and Montiel, 2018a). The deformable tracking for deformable monocular SLAM (Lamarca et al., 2020, Rodríguez et al., 2020) was built on top of the method proposed in Chapter 2. Template-based methods rely on a global model that connects all the map points and are prompt to fail when the map points are simply not connected or have a different relation.

In this paper, we formulate the points of the surface separately as surfels -surface element- and jointly estimate its position for each frame and the position of the camera. One of the closest approach was the scene flow technique proposed by Devernay et al. (2006), that uses surfels to track some points of the scene, however they rely in a multi-camera setup, while we use a monocular camera. Using surfels, we can represent more general disconnected shapes of the scene and movements, and avoid the usage of a global deformation model.

Piecewise methods are local techniques where the non-rigid object is a collection of predefined patches that move independently as rigid objects. The first work in using this strategy was proposed by Varol et al. (2009), imposing a 3D global consistency in overlapping points.

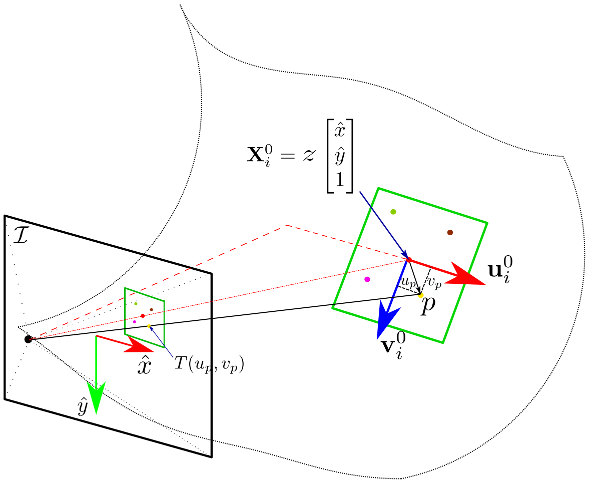

Fig. 5.2 Parametrization of a surfel in the initial image. Coordinates of the surface u and v correspond to the normalized coordinates in the image $\hat{x}$ and $\hat{y}$. We obtain z from the depth image, and we estimate the tangent space vectors $\mathbf{u}_i^t$ and $\mathbf{v}_i^t$ as the directional derivatives in the image coordinates $\mathcal{I}$.

A relaxation to the piecewise rigid constraint was given by Fayad et al. (2010), assuming each patch deforms with a quadratic physical model accounting for linear and bending deformations. All these methods required an initial patch segmentation and the number of overlapping points, to this end Russell et al. (2011) optimize the number of patches and overlapping through an energy-based optimization. In contrast, Taylor et al. (2010) constructs a triangular mesh, connecting all the points, and considering each triangle as being locally rigid, being able to deal with topological changes. Our method belongs to this family of methods, but in contrast, we do not assume that the points are overlapping.

The SD-DefSLAM proposed in Chapter 4 is a semi-direct method that replaces the feature-based tracking of DefSLAM with a multiscale Lucas-Kanade tracker, resulting in an improvement of the track lengths and reconstruction accuracy. In this work, we take advantage of direct photometric error to recover the 3D relative position of the surface points. Direct methods use the photometric error and have been proven extremely accurate in the rigid SLAM case (Engel et al., 2017, Zubizarreta et al., 2020) and other NRSfM works (Yu et al., 2015).

5.3 Formulation

This section is devoted to formalize the parametrization of a surfel and the photometric equations describing its observation by a projective camera.

5.3.1 Notation

Bold letters represent vectors or matrices ($\mathbf{X}$). Scalars will be represented by light lowercase letters (t), image brightness functions by light uppercase letters (I). Superindex t denotes the frame in which the estimation is done. Subindex i identifies the surfel. Subindex p refers to pixel coordinates in reference local to the surfel. To simplify the index notation all the scene points coordinates are in the world reference. Camera poses are represented as transformation matrices $\mathbf{T}_{cw} \in SE(3)$, transforming the coordinates of point from the world frame into the camera frame.

5.3.2 Surfel parametrization

Assuming a continue and derivable C^1 surface, a point $\mathbf{X}_i^t$ is represented by a surfel $\mathbf{S}_i^t$ contained in the tangent space of the surface at the point. Thus, a generic 3D point p belonging to the surfel can be parametrized using two local coordinates u_p and v_p around $\mathbf{X}_i^t$:

$$\mathbf{S}_i^t(u_p, v_p) = \mathbf{X}_i^t + \mathbf{J}_i^t \begin{bmatrix} u_p \\ v_p \end{bmatrix} \tag{5.1}$$

$$\mathbf{J}_i^t \;\; = \;\; \begin{bmatrix} \mathbf{u}_i^t & \mathbf{v}_i^t \end{bmatrix} \in \mathbb{R}^{3\times 2} \tag{5.2}$$

where $\mathbf{J}_i^t$ is the so-called Jacobian matrix whose columns are a pair of vectors forming a base of the tangent space. As described in Eq. (5.6), $\mathbf{X}_i^t$ and $\mathbf{J}_i^t$ are defined for each frame in terms of the corresponding values at $t = 0$, $\mathbf{X}_i^0$ and $\mathbf{J}_i^0$, whose initialization from the first image is described next.

5.3.3 Surfel initialization

We assume the scene surface is defined by means of the depth function: $z(\hat{x}, \hat{y}) : \mathbb{R}^2 \to \mathbb{R}$ in terms of the normalized retina coordinates $\hat{x}, \hat{y}$. This depth function can be provided by a depth sensor (RGB-D camera or stereo rig). Then, $\mathbf{X}_i^0$ and $\mathbf{J}_i^0$ are estimated as:

$$\mathbf{X}_i^0 = z(\hat{x}, \hat{y}) \begin{bmatrix} \hat{x} \\ \hat{y} \\ 1 \end{bmatrix} \tag{5.3}$$

and

$$\mathbf{J}_i^0 = \begin{bmatrix} z + \hat{x}\frac{\partial z}{\partial \hat{x}} & \hat{x}\frac{\partial z}{\partial \hat{y}} \\ \hat{y}\frac{\partial z}{\partial \hat{x}} & z + \hat{y}\frac{\partial z}{\partial \hat{y}} \\ \frac{\partial z}{\partial \hat{x}} & \frac{\partial z}{\partial \hat{y}} \end{bmatrix} \tag{5.4}$$

For the experiments, we initialize surfels in the interest points extracted with Shi-Tomasi Jianbo Shi and Tomasi (1994).

5.3.4 Photometric error

We denote the projection function as $\pi(\cdot) : \mathbb{R}^3 \to \mathbb{R}^2$. For our experiments, we use the pinhole camera model. Note that this can be easily substituted by any other camera model.

We optimize the difference between the intensities of points in the surfel and the intensities in their reprojections in the current image:

$$\mathcal{P}_i^t = \sum_p \left(\alpha_i^t I_t\left(\pi\left(\mathbf{T}_{cw}\mathbf{S}_i^t(u_p, v_p)\right)\right) + \beta_i^t - T(u_p, v_p)\right)^2 \tag{5.5}$$

where $\mathbf{T}_{cw}$ is the pose of the camera with respect to the world. $\mathbf{S}_i^t(u_p, v_p)$ is in the world reference. We compensate the illumination changes by means of a gain (α_i^t) and a bias (β_i^t) per surfel and per image. That allows us to synthesize the deformed surfel into the image, thus our error function takes into account the local deformation.

We define a symmetric uniform grid in the surfel local coordinates that is reprojected into the inital image to extract the surfel texture $T(u_p, v_p)$ parameterized by u_p and v_p.

5.4 Direct and Sparse camera tracking with static camera

Let's assume in this section that the camera is fixed and the initial values of the textured surfels are given in advance, and we want to estimate the deformation for each incoming image. With our formulation, the initial surfel is defined by its initial position $\mathbf{X}_i^0$, its Jacobian $\mathbf{J}_i^0$ and its texture $T(u_p, v_p)$.

The geometrical transformation of the surfel is expressed as:

$$\mathbf{S}_i^t(u_p, v_p) = (\mathbf{X}_i^0 + \mathbf{t}_i^t) + \mathbf{R}_i^t\mathbf{J}_i^0\mathbf{F}_i^t \begin{bmatrix} u_p \\ v_p \end{bmatrix} \tag{5.6}$$

where $\mathbf{t}_i^t \in \mathbb{R}^3$ is the translation of the surfel, $\mathbf{R}_i^t \in \mathbf{SO}(3)$ is the rotation of the surfel modeled by the 3 parameters of its Lie algebra, and $\mathbf{F}_i^t \in \mathbb{R}^{2\times 2}$ is a symmetric matrix that represent the

Table 5.1 Deformation tensor F_i^t for different local deformation models.

	Isometry	Conformal	Equireal	General
F_i^t	$\mathbb{I}_2$	$s\mathbb{I}_2$	$\begin{bmatrix} \alpha & \beta \\ \beta & \frac{1+\beta}{\alpha} \end{bmatrix}$	$\begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix}$
Variables	-	s	α,β	α,β,γ

deformation tensor. Its diagonal components represent the stretching of the tangent vectors, and its off-diagonal element models the angle change between these vectors, i.e. the shearing.

As seen in Table 5.1, the most restrictive local deformation is isometric. This constraint is equivalent to a rigid movement of the surfel. When the surfel deformation is not bounded the first ambiguity arises:

Growing map ambiguity. The depth component of the translation of the surfel and the surfel size can be coupled in such a way that changing its depth and size produces the same image.

Proof. We define an μ factor that transforms the position and the deformation of the surfel as:

$$(\mathbf{X}_i^0 + \mathbf{t}_i^t) = \mu \mathbf{X}_i^0 \tag{5.7}$$

$$\mathbf{F}_i^t = \begin{bmatrix} \mu & 0 \\ 0 & \mu \end{bmatrix} \tag{5.8}$$

$$\hat{\mathbf{S}}_i^t(u_p, v_p) = \mu \mathbf{X}_i^0 + \mu \mathbf{R}_i^t \mathbf{J}_i^0 \begin{bmatrix} u_p \\ v_p \end{bmatrix} = \mu \mathbf{S}_i^t(u_p, v_p) \tag{5.9}$$

Under perspective projection any surfel $\hat{\mathbf{S}}_i^t(u_p, v_p)$ multiplied by μ produces the same image $\pi(\mathbf{S}_i^t(u_p, v_p)) = \pi(\hat{\mathbf{S}}_i^t(u_p, v_p))$. □

To solve this ambiguity, we impose local isometry within the surfel. Isometry is a distance preserving transformation. We propose two alternatives to code the isometry, as a hard constraint or as a soft constraint.

Isometry as hard constraint implies the transformation of the surfel only as a rigid body motion, in other words, the deformation matrix, $\mathbf{F}_i^t = \mathbb{I}_2$. The motion is defined by 6 parameters (3 for translation + 3 for rotation).

$$\mathbf{J}_i^t = \mathbf{R}_i^t \mathbf{J}_i^0 \quad ; \quad \mathbf{R}_i^t \in \mathbf{SO}(3) \tag{5.10}$$

Thus, our cost function is defined only by the photometric error (Eq. (5.5)):

$$\mathbf{t}_i^t, \mathbf{R}_i^t = \underset{t_i^t, R_i^t, \alpha_i, \beta_i}{argmin} \mathcal{P}_i^t \tag{5.11}$$

In the case of a soft constrain we penalize the stretching and shearing of the surfel. It is formulated through the tangent plane $\mathbf{J}_i^t$. We define a deformation energy quadratic error as:

$$\mathcal{I}_i^t = \left\| \mathbf{F}_i^t - \mathbb{I}_2 \right\|_2^2 \tag{5.12}$$

The soft constraint is modeled by means of the deformation energy coded by 3 additional parameters defining the symmetric matrix $\mathbf{F}_i^t$. In other words, the surfel can stretch and shear if it explains better the image, but it tends to stay as close as possible to its original shape. $\mathcal{I}_i^t$ is a scalar that penalises the shearing and stretching.

Finally, the optimization is a combination of the forward-compositional photometric error and the deformation energy. The deformation energy regularization is weighted by a constant $\omega_{\mathcal{I}}$,

$$\mathbf{t}_i^t, \mathbf{R}_i^t, \mathbf{F}_i^t = \underset{t_i^t, R_i^t, F_i^t, \alpha_i, \beta_i}{argmin} \; \mathcal{P}_i^t + \omega_{\mathcal{I}} \mathcal{I}_i^t \tag{5.13}$$

All the errors considered in (5.11, 5.13) are quadratic, so it can be solved as a non-linear least-squares problem. We propose Levenberg–Marquardt (LM) optimization Nocedal and Wright (2006). The LM algorithm is a trust-region method that combines a Gauss-Newton and steepest descend. The step control is defined through the damping factor λ that weights both methods, λ also allows to control the step size. The Hessian is approximated as $H \approx J^\top J$.

During the optimization, the data association between the images is changed boosting the accuracy, however the convergence basin of the photometric optimization is small. We propose an strict step size control to avoid leaving the convergence basin. We confine the step to an ellipsoidal trust region defined by the diagonal matrix $D_w = \text{diag}(H)$. We apply a step policy where λ is limited to be ≥ 1 during the first steps to avoid long steps when far from the minimum. In a subsequent stage λ is allowed to be reduced in order to benefit from the Gauss-Newton quadratic convergence.

A singular values analysis of the Hessian matrix points out that this matrix is ill-conditioned, i.e. the ratio between the smallest and biggest singular values is $\ll 1$. This reflects a different scaling in translation, rotation and deformation parameters. Thus, we propose to use a diagonal scaling preconditioner matrix $D_s(i,i) = \frac{1}{\sqrt{s_i}}$ to avoid numerical issues, being s_i the diagonal values of $(H + \lambda D_w)$. At each iteration the Δx is then estimated as:

$$\begin{aligned} D_s\left(H+\lambda D_w\right)D_s\Delta x^* &= -D_s J^\top \mathbf{r} && (5.14)\\ \Delta x &= D_s\Delta x^* && (5.15)\end{aligned}$$

In addition, to avoid mismatches due to discontinuities and light reflections, we saturate the photometric error. We also carry out a multi-scale optimization to increase the convergence basin observing that in case of temporal discontinuities or fast movements the algorithm becomes much more robust.

We detect the outlier surfels using a threshold in the Zero Normalized Cross correlation (ZNCC) between the texture of the surfel and the texture of its reprojection because it is illumination invariant. If the ZNCC drops under a threshold the surfel can be assumed as badly tracked and the corresponding observation is marked as an outlier.

The algorithm complexity is linear in the number of points since each new point would suppose a new optimization and cubic in the number of pixels per surfel since the Jacobian block of the surfel is dense and it increases one row per new pixel included.

5.5 Direct and Sparse deformable tracking

Deformable tracking algorithm takes as input the textured surfels and the initial camera pose. Then, it estimates the deformation of the map and the camera pose *wrt.* the map.

Floating map ambiguity. Surfel position and camera pose are coupled and can be varied producing the same projection of the pixel in the image.

Proof. Eq. (5.1) can be rewritten as:

$$\mathbf{S}_i^t(u_p,v_p) = \begin{bmatrix}\mathbf{R}_i^t & \mathbf{t}_i^t\\ \mathbf{0} & 1\end{bmatrix}\begin{bmatrix}\mathbf{J}_i^0\mathbf{F}_i^t\begin{bmatrix}u_p\\ v_p\end{bmatrix} & \mathbf{X}_i^0\\ \mathbf{0} & 1\end{bmatrix} \tag{5.16}$$

where we can define the rigid movement of the surfel as $\mathbf{T}_{iw}^t \in \mathbf{SE}(3)$:

$$\mathbf{T}_{iw}^t = \begin{bmatrix}\mathbf{R}_i^t & \mathbf{t}_i^t\\ \mathbf{0} & 1\end{bmatrix} \tag{5.17}$$

The camera pose $\mathbf{T}_{cw}$ and the transformation of a surfel $\mathbf{T}_{iw}^t$ are coupled and can be interpreted as a arbitrary movements of the camera or as a movement of the surfel.

$$\begin{aligned} \left[\mathbf{S}_i^t(u_p,v_p)\right]_c &= \mathbf{T}_{cw}\mathbf{T}_{iw}^t \begin{bmatrix} \mathbf{J}_i^0\mathbf{F}_i^t \begin{bmatrix} u_p \\ v_p \end{bmatrix} & \mathbf{X}_i^0 \\ \mathbf{0} & 1 \end{bmatrix} & (5.18) \\ \hat{\mathbf{T}}_{cw} &= \mathbf{T}_{cw}\mathbf{T}_{iw}^t = \mathbf{T}_{cw}^*\mathbf{T}_{iw}^{t^*} & (5.19) \end{aligned}$$

□

To avoid the ambiguity, we propose to soft-constrain each surfel position around an equilibrium position $\mathbf{X}_{e_i}^t$ with the regularizer $\mathcal{E}_i^t$:

$$\mathcal{E}_i^t = \left(\mathbf{X}_i^t - \mathbf{X}_i^0\right)^\top \Sigma_i^{-1} \left(\mathbf{X}_i^t - \mathbf{X}_i^0\right) \tag{5.20}$$

This position gives a reference for the camera estimation. We can understand the camera movement in our approach as the global rigid movement, and the deformation of the surfels as movements around that equilibrium. Σ_i is the covariance that the surfels can reach in its movement.

If the trajectory of the points along the sequence is known in advance, the equilibrium point can be estimated as their average position and its covariance. In the case that the position and covariance are unknown, we approximated as it is around the original position and select a heuristic covariance with the expected movement. Lower covariances lead to more rigid interpretation.

Similarly to Sec. 5.4, it is possible code the isometry as a hard or as a soft constraint. The optimization for the hard constraint case is:

$$\mathbf{X}_i^t, \mathbf{J}_i^t, \mathbf{T}_{cw} = \underset{X_i^t,J_i^t,\alpha,\beta,T_{cw}}{argmin} \sum_{i\in\mathcal{X}} \mathcal{P}_i^t + \omega_{\mathcal{E}}\mathcal{E}_i^t \tag{5.21}$$

The movement of the camera is defined by using Lie algebra of $\mathbf{SE}(3)$. We linearize in the solution for each step and update the pose each step as:

$$\hat{\mathbf{T}}_{cw} = \exp(\zeta)\mathbf{T}_{cw} \tag{5.22}$$

The optimization is done by using Levenger-Marquard. We again need to scale the parameters through D_s and control the step with λD_w.

5.6 Experiments

We evaluate the performance of the two proposed methods: Sparse Deformable Tracking with and without static camera, in rigid and deformable scenarios. We use sequences of laboratory-controlled scenarios from CVLab (Varol et al., 2012) and sequences of intracorporeal scenes selected from the Hamlyn Dataset (Mountney et al., 2010b). A video with the results is provided as supplementary material[1].

5.6.1 Tuning

Surfel size

Our primary assumption is that any surface can be locally approximated by the tangent plane. The accuracy of the approximation deceases with the distance to the centre of the surfel, hence it decreases with the surfel size. In contrast, bigger surfels allow more accurate estimates of the surfel geometry. Thanks to the saturation policy that we apply, we have noticed that even big surfels can accurately estimate the surfel geometry. We chose a surfel size of ≈ 23 pixels experimentally. In experiments in the Kinect paper dataset, we have observed that the error is reduced for bigger surfels, even if they do not fully accomplish the planarity assumption. Too big surfels lead to problems with spatial discontinuities in the scene.

Multi-scale

The convergence basin of the photometric methods is around one pixel, using multi-scale increases it to more that one pixel in the finest scale. We use the solution of a coarse scale as the initial guess of the next finer scale. In the kinect paper and T-shirt datasets, we found several missing frames. That precludes the convergence for many surfels if only the finest scale is used. Using 3 scales, the algorithm converges despite the missing frames.

Outlier rejection

We evaluate the ZNCC method to classify inliers as points that have converged correctly in the optimization. Positives are inliers and negatives are outliers. The ground truth of the correct tracks are classified through a threshold in the RMSE *wrt.* the ground truth surfel trajectory. We show the ROC curve in Fig. 5.3 right *wrt.* varying the ZNCC threshold. Ideally,

[1] https://drive.google.com/file/d/1XJFbLsp_76eGqDisJj8Sjcljaf3F1c94/view?usp=sharing

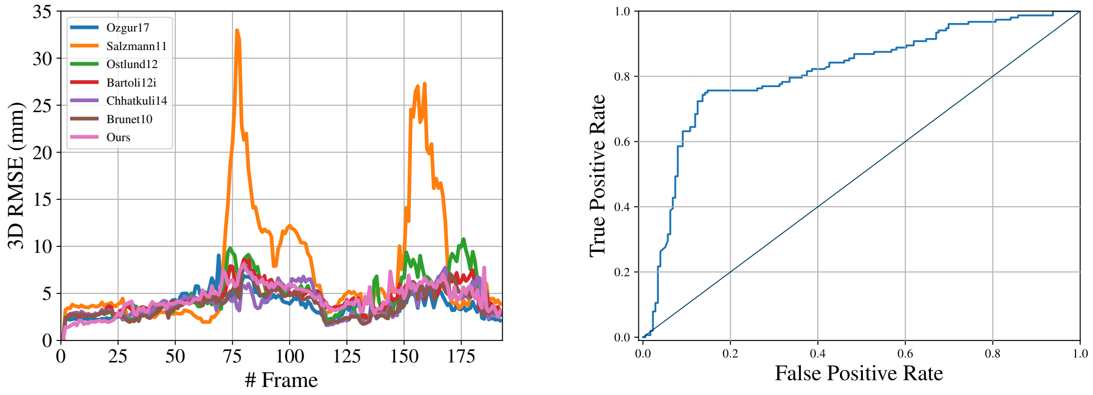

Fig. 5.3 Left: Comparison in kinect Paper dataset from CVLab. Our method tracks individual surfels with similar accuracy than template-based methods that assume surface continuity. Right: Outlier detection, ROC curve *wrt.* the ZNCC threshold

the more up to the left the curve is, the better the classifier is. We finally select a value for the ZNCC of 0.95 for the experiments in the Kinect dataset, and 0.85 for the Hamlyn dataset.

Soft vs. Hard isometric constraint

In Sec.5.3 we have discussed two ways of constraining the deformation of the surfel. We have validated them in different sequences. We have observed that a soft constrained deformation model does not improve the accuracy of the system. Isometry seems to be a good local approach for the local deformation of the surfels, and thanks to treating the points individually we can recover very different global deformations. For instance, in sequence 21 from Hamlyn Dataset we can cover non-isometric global deformations, or in the kinect Paper Dataset, we can track multiple objects, because we are treating each point individually. We use the isometry as a hard constrain (Eq. 5.11 and Eq. 5.21) in all the rest of experiments.

5.6.2 Deformable Tracking with Static Camera

In this section, we analyse the performance of deformable tracking with static camera in real sequences. The CVLab's, T-shirt and paper dataset were recorded with a Kinect RGB-D camera. We also test intracorporeal sequences from the Hamlyn Dataset, in this case with stereo camera. We use the first depth image to initialise the surfels, i.e. the position, Jacobian and texture of each surfel. Notice that our system is monocular, hence we only process the gray images obtained with the RGB-D camera or with the left camera.

We compare our method against some reference shape-from-template (SfT) methods (Özgür and Bartoli, 2017, Salzmann and Fua, 2011, Östlund et al., 2012, Bartoli et al., 2012, Chhatkuli et al., 2014b, Brunet et al., 2010) in the Kinect paper dataset from CVLab. This

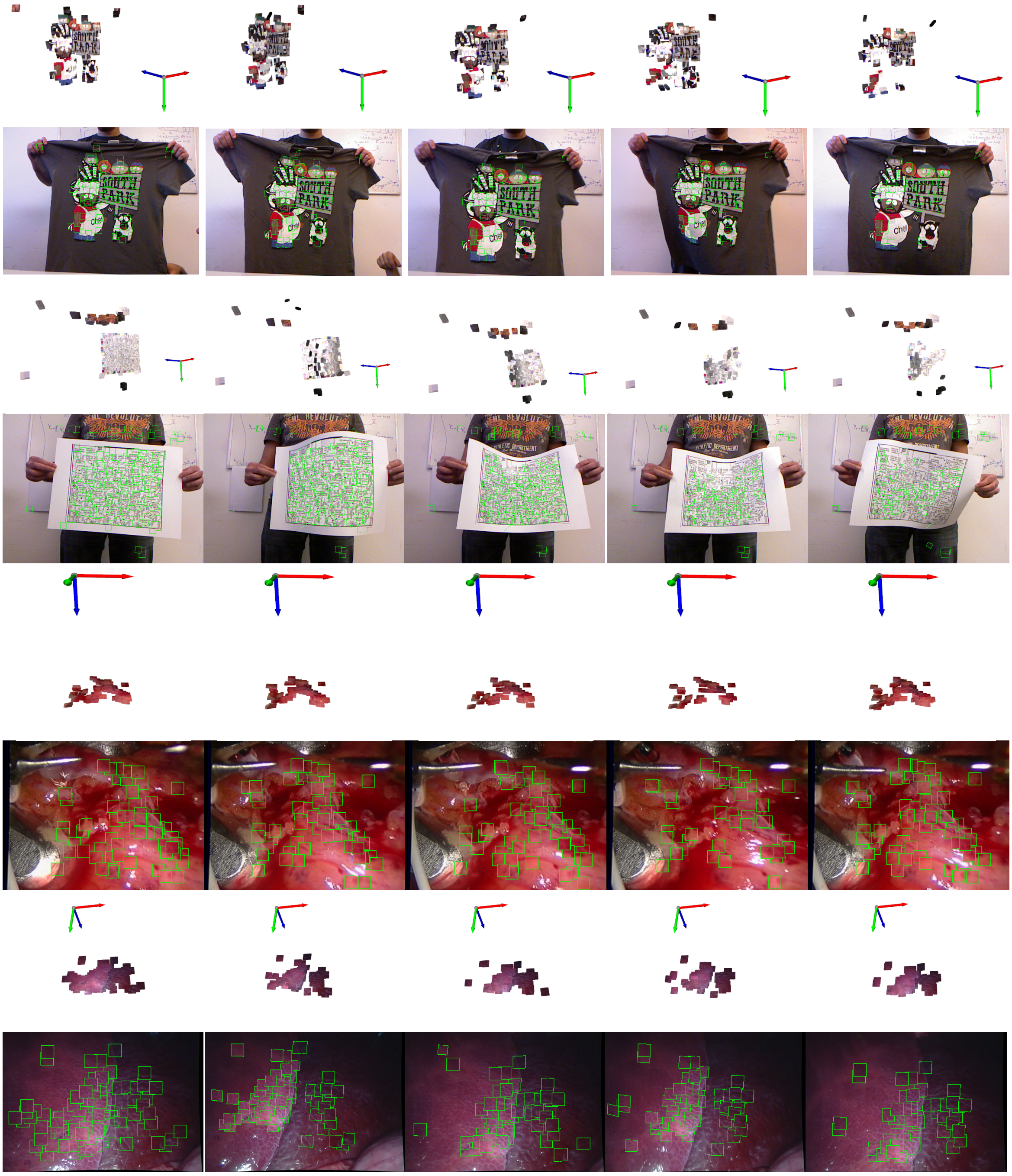

Fig. 5.4 Deformable tracking results, two rows per sequence, first the 3D reconstruction, then the RGB frames. Even if the surfels are estimated independently the entire reconstruction displays an homogeneous consistency. (See entire sequences in supplementary material). **1st-2nd rows, kinect T-Shirt dataset**, frames # 0,# 70,# 150,# 250 and #300. **3rd-4th rows kinect Paper dataset**, frames # 0,# 70,# 130,# 160 and #180. **5th-6th rows Hamlyn 4 (Heart sequence)**, frames # 0,# 12,# 16,# 26 and #40. **7th-8th Hamlyn 21 (Liver sequence)**, frames # 750,# 800,# 850,# 900 and #950.

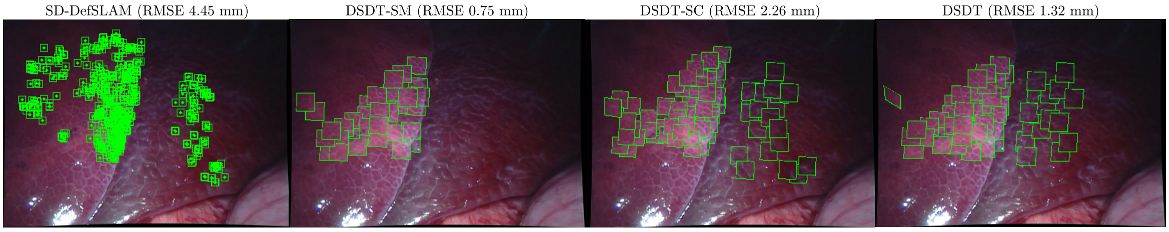

Fig. 5.5 Non-isometric deformation results. Left to right: SD-DefSLAM, DSDT w. static map, w. static camera and DSDT.

sequence consists in a paper deformed isometrically. This sequence has two main challenges for our method: illumination changes and temporal discontinuities (missing frames) in frames #70, #120, #130 and #150. In contrast to the other methods, ours is the only one using photometric error.

As shown in Fig. 5.3, the first notable result is that our method can track individual surfels with similar accuracy to methods that assume smoothness in the surface. We also noticed that optimization-based methods (Salzmann and Fua, 2011, Östlund et al., 2012) get worse results than the rest. These methods assume sequential images and the missing frames break this assumption worsening the results. In our case, something similar happens, but thanks to the multi-scale configuration the convergence gets substantially improved. We also conclude that the local compensation of the illumination presented in Eq. 5.5 is crucial to track a higher number of surfels.

We have seen that assuming smooth surfaces improves the results in the paper area. However, this precludes the usage of this method in discontinuous surfaces. In contrast, as we have not assumed any regularizer between the individual surfels, we can track surfels not only on the paper area, but also on the person's T-Shirt and on the white board (See Fig. 5.4). Discontinuities raise other challenges like occlusions that are successfully managed with the saturation of the photometric error.

5.6.3 Direct and Sparse Deformable Tracking

In this section, we analyse the deformable tracking, and we compare the advantage of a world-centric (DSDT) approach (Sec. 5.5) where the camera can move versus a camera-centric approach with static camera (DSDT-SC) (Sec. 5.4).

We compare both methods against our tracking with a static (i.e. rigid) map (DSDT-SM), the deformable tracking from SD-DefSLAM (Rodríguez et al., 2020) and the tracking of ORBSLAM (Mur-Artal and Tardós, 2017) in the sequences 6, 20 and 21 from Hamlyn dataset (Mountney et al., 2010b). All the methods are initialized with a stereo pair in the

Table 5.2 Comparison of our method against ORBSLAM and SD-DEFSLAM for the Hamlyn Dataset sequences 6, 20 and 21. We report RMSE and # of frames processed.

		Rigid map		Deformable map		
		ORBSLAM	DSDT-SM	SD-DefSLAM	DSDT-SC	DSDT
Dataset 6	RMSE	4.85	3.26	2.72	9.24	3.17
	# Fr.	128	200	286	334	300
Dataset 20	RMSE	1.37	1.37	4.68	3.09	2.9
	# Fr.	220	210	252	350	500
Dataset 21	RMSE	-	-	6.19	1.81	1.30
	# Fr.	-	-	323	321	300

ORB: ORBSLAM (Mur-Artal and Tardós, 2017); **DSDT-SM**: Direct Sparse with static map; **SD**: Semi-direct DefSLAM (Rodríguez et al., 2020); **DSDT-SC**: Direct Sparse Deformable Tracking with static camera; **DSDT**: Direct Sparse Deformable Tracking

same frame and no mapping is allowed, i.e., tracking the initial map without refining or extending it.

Dataset 6 (from frame #50) is an abdominal exploration where the scene remains almost rigid. It has a planar topology in the area where the camera closes up, and a small discontinuity due to a nerve. The texture is minimal except for the veins. The deformable tracking can process 300 frames before tracking loss with an RMS error close to 3mm. On the other hand, DSDT-SC can process a similar number of frames but with a much bigger error. We conclude that the regularizer added in the deformation tracking gives hints to the optimizer yielding better performance. SD-DefSLAM processes a few frames less with similar error, however it only focuses on the planar area.

Dataset 20 (from frame #750) is another abdominal exploration, but in this case the scene contains some global near-isometric deformation keeping a similar shape. Deformable tracking can track 500 frames from the initialization frame, in contrast to the camera-centric approach DSDT-SC that only process 350 frames. Again thanks to the movement of the camera we are able to recover many points that are missed by the DSDT-SC, being able to process a higher number of frames. SD-DefSLAM assumes a global isometry by imposing a mesh, and in this case, as we do not update the mesh, it misses a big part of points when near-isometric deformation happens. Rigid methods focus in a small area of the scene being badly conditioned. In contrast, our direct method tracks almost the double of features in comparison with ORBSLAM.

The last one is Dataset 21 (from frame #750) where the camera images two lobes of a liver moving as independent bodies, one lobe sliding over the other (See Fig.5.5). Thanks to our formulation, the proposed deformable tracking can process global non-isometric deformations. We observe that our system is able to cope with deformations from independent bodies. In this case, SD-DefSLAM can track some of the points but with a high RMSE because its isometric deformation model cannot code the deformation actually observed.

5.7 Conclusion

In this paper we have proposed a novel approach for deformable tracking in deformable SLAM. Each map point is modeled as a 3D surfel that is a local approximation of the scene surface. The deformations of the map are modeled through the movement of these surfels. In contrast to the previous deformable tracking methods we have proposed to remove any connection between 3D map points.

We have proved experimentally that the local model for the deformable tracking can perform similarly to the state-of-the art methods and can perform more robustly and more accurately than the global methods in scenes composed of discontinuous surfaces, or with global non-isometric deformations. In addition, we reassert the potential of the direct methods over the feature-based equivalents.

Future work could extend this deformable tracking into a deformable mapping able to reconstruct scenes composed of discontinuous surfaces, or with global non-isometric deformations. With this two algorithmic components, it will be possible to create a new generation of monocular Deformable SLAM algorithms able to work in a wider range of scenarios.

Chapter 6

Conclusions

In this thesis, we have conceived the first Monocular SLAM able to work in deforming scenarios. We have started a new research branch of SLAM that is able to work in more general scenarios. We have shown that the deformable methods proposed are needed and outperform the traditional ones.

Our first contribution was the deformable tracking presented in Chapter 2. In this first version of the system, we have analysed how a deformable map must be treated. We introduce an optimization method based on physical models, inspired in the template-based methods that usually scope a small object that fits in the camera field-of-view. It recovers the shape of the object by minimizing simultaneously the reprojection error and the deformation energy. The first challenge of our method was to manage only partial views of the map. We saw that trying to recover the entire deformation of the map was not advantageous in any way, instead it is much better to recover just the area observed. The main disadvantage of this method is that it needs a previously known shape-at-rest of the map.

In the Chapter 3, we propose to complement the deformable tracking with a deformable mapping. This deformable mapping was able to map the scene even with deformations from a monocular batch of images. It substitutes the classical rigid methods, like the bundle adjustment, by the NRSfM of Parashar et al. (2017) which reconstructs surfaces assuming isometry, infinitesimal planarity and continuity. The NRSfM method was conceived to deal small objects like pillows, papers or fabrics. In our work, we design the deformable mapping with the same assumptions, but with three main outstanding characteristics: it has automatic data association, it is an incremental method, and it is able to explore new regions of the scene keeping a coherent scale of the map. Bringing together the deformable tracking and deformable mapping in a PTAM-like structure, our method is able to successfully process the Mandala Dataset and medical sequences achieving state-of-the-art results and in real-time. Thus, we can conclude that DefSLAM is the first deformable monocular SLAM system.

In Chapter 4, we improve the DefSLAM to make it more robust in medical scenarios. DefSLAM's main drawback for this sequences was the feature-based matching, specifically ORB method. The ORB method uses a FAST extractor that is not very repeatable, yielding to a considerable number of unobserved features from frame to frame. We substituted it by a Lukas-Kanade tracker that tracks the map points by using photometric error. This method was shown to outperform the previous matching for the Mandala and medical sequences both in terms of accuracy and robustness. Thanks to avoid the extraction of keypoints per frame, the system performs with a similar computational time. In addition, we included a module for relocalization in case of tracking failure and a segmentation module to remove dynamic objects not related with the scene. Thanks to all the improvements proposed we were able to process colonoscopy sequences with deformation.

The main issue of both systems comes with the assumption of continuity and isometry. These assumptions are quite restrictive and although they approximately perform well in near-isometric sequences, this is not the case for many medical sequences. That is why in our last Chapter 5, we tackle the problem of reconstruct non-isometric deformations. To achieve our purpose, we exploit the direct photometric error in order to track the points individually without defining a global deformation model. Due to this new formulation we are able to capture much better the nature of the deformations observed in the medical scene. In our experiments, we proved that we can reconstruct non-isometric deformations or surface with topological changes outperforming the deformable tracking methods proposed in the previous chapters.

We can draw a general conclusion from this thesis by writing that we have proposed the first system capable of work in deformable scenarios. And, in addition, we have also made contributions within the current non-rigid monocular methods. On the first proposal, we have developed a template-based camera tracking method able to work with maps partially observed. On the second set of proposals, we have developed novel deformable mapping inherited from the NRSfM able to perform explorations. We make our method more robust exploding the photometric error and including new techniques for place recognition and segmentation for dynamic SLAM. Finally, we have developed a new deformable tracking method able to process near-isometric and non-isometric deformations more faithfully, yielding in promising results and that serve as base for future deformable monocular SLAM methods.

6.1 Limitations and Future Work

In this thesis, we have explored geometric methods to create a new SLAM method able to tackle the deforming scenes. To be able to reconstruct deforming maps, we had to make many assumptions and incorporate physical models into the equations of the monocular SLAM. This models are more restrictive than in the rigid SLAM, e.g. assuming continuous or isometric surfaces. To have a fully robust and applicable method, those constrains must be relaxed. In Chapter 5, we have seen that it is possible to recover the deformation of the map and the camera pose only with local restrictions. From that point, we consider that the next step is to build up an entire counterpart mapping able to work without assuming continuity. This is crucial to build maps that are not totally smooth or with non-planar topology of the reconstructed surface.

Finally, we have focused in the geometrical formulation leaving the data-driven approaches as a secondary topic. We have seen the improvements that this approaches can bring to the field in many cases, e.g. segmenting or optical flow. In the Bachellor Degree of Diego Royo 2020, we concluded that the optical flow with the Flownet proposed by Dosovitskiy et al. (2015) can be used as a substitute of the Schwarp method used in the first stage of the deformable mapping, achieving irregular and stepped warps. In Chapter 4, we use a segmentation network to ignore the tools that enter in the camera field of view and are not part of the map. The last collaboration with David Recasens in (Recasens et al., 2021) where we explored the usage of Monodepth2 (Godard et al., 2019) as Mapping for the system since it can generate depth from single images given similar results to the obtained with geometrical methods. That is only a small proof of the potential impact that deep learning can bring to this area.

We have focused this manuscript on the idea of conceiving a SLAM algorithm able to tackle deformations of in-body sequences. One important underlying concept that was built is the idea of a changing map. SLAM for applications in non-static scenarios, like a boat in the middle of the ocean or in a street where cars are moving, will need a map with a mathematical model that can represent the evolution of the map to have a robust and faithful performance.

References

Agarwal, Sameer, Keir Mierle, and Others (2010), “Ceres solver.” *http://ceres-solver.org*.

Agudo, Antonio, Lourdes Agapito, Begona Calvo, and Jose Maria Martinez Montiel (2014), “Good vibrations: A modal analysis approach for sequential non-rigid structure from motion.” In *CVPR*.

Agudo, Antonio, Begoña Calvo, and J Montiel (2012), “3d reconstruction of non-rigid surfaces in real-time using wedge elements.” In *ECCV Workshops*, 113–122, Springer.

Agudo, Antonio and Francesc Moreno-Noguer (2015), “Simultaneous pose and non-rigid shape with particle dynamics.” In *CVPR*.

Agudo, Antonio, Francesc Moreno-Noguer, Begoña Calvo, and José María Martínez Montiel (2015), “Sequential non-rigid structure from motion using physical priors.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38, 979–994.

Akhter, Ijaz, Yaser Sheikh, Sohaib Khan, and Takeo Kanade (2011), “Trajectory space: A dual representation for nonrigid structure from motion.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33, 1442–1456.

Alcantarilla, Pablo F, Jesús Nuevo, and Adrien Bartoli (2011), “Fast explicit diffusion for accelerated features in nonlinear scale spaces.” *IEEE Trans. Patt. Anal. Mach. Intell.*, 34, 1281–1298.

Arandjelovic, Relja, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic (2016), “NetVLAD: CNN architecture for weakly supervised place recognition.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5297–5307.

Baker, Simon and Iain Matthews (2004), “Lucas-Kanade 20 years on: A unifying framework.” *International Journal of Computer Vision*, 56, 221–255.

Ballester, Irene, Alejandro Fontan, Javier Civera, Klaus H Strobl, and Rudolph Triebel (2020), “Dot: Dynamic object tracking for visual slam.” *arXiv preprint arXiv:2010.00052*.

Bartoli, A., Y. Gérard, F. Chadebecq, and T. Collins (2012), “On template-based reconstruction from a single view: Analytical solutions and proofs of well-posedness for developable, isometric and conformal surfaces.” In *CVPR*.

Bartoli, Adrien, Vincent Gay-Bellile, Umberto Castellani, Julien Peyras, Søren Olsen, and Patrick Sayd (2008), “Coarse-to-fine low-rank structure-from-motion.” In *Computer Vision and Pattern Recognition, 2008. CVPR 2008.*, 1–8, IEEE.

Bartoli, Adrien, Yan Gérard, François Chadebecq, Toby Collins, and Daniel Pizarro (2015), "Shape-from-template." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37, 2099–2118.

Bescos, Berta, José M Fácil, Javier Civera, and José Neira (2018), "DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes." *IEEE Robotics and Automation Letters*, 3, 4076–4083.

Bouguet, Jean Yves (2001), "Pyramidal implementation of the affine Lucas Kanade feature tracker. Description of the algorithm." Technical report, Intel corporation.

Bradski, G. (2000), "The OpenCV Library." *Dr. Dobb's Journal of Software Tools*.

Bregler, Christoph, Aaron Hertzmann, and Henning Biermann (2000), "Recovering non-rigid 3D shape from image streams." In *CVPR*.

Brunet, Florent, Richard Hartley, Adrien Bartoli, Nassir Navab, and Remy Malgouyres (2010), "Monocular template-based reconstruction of smooth and inextensible surfaces." In *ACCV*, Springer.

Campos, Carlos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós (2021), "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam." *IEEE Transactions on Robotics*.

Casillas-Perez, David, Daniel Pizarro, David Fuentes-Jimenez, Manuel Mazo, and Adrien Bartoli (2019), "Equiareal shape-from-template." *Journal of Mathematical Imaging and Vision*, 61, 607–626.

Chhatkuli, Ajad, Daniel Pizarro, and Adrien Bartoli (2014a), "Non-rigid shape-from-motion for isometric surfaces using infinitesimal planarity." In *BMVC*.

Chhatkuli, Ajad, Daniel Pizarro, and Adrien Bartoli (2014b), "Stable template-based isometric 3d reconstruction in all imaging conditions by linear least-squares." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chhatkuli, Ajad, Daniel Pizarro, Adrien Bartoli, and Toby Collins (2017), "A stable analytical framework for isometric shape-from-template by surface integration." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 833–850.

Chhatkuli, Ajad, Daniel Pizarro, Toby Collins, and Adrien Bartoli (2016), "Inextensible non-rigid shape-from-motion by second-order cone programming." In *CVPR*.

Chiuso, A., P. Favaro, Hailin Jin, and S. Soatto (2002), "Structure from motion causally integrated over time." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24, 523–535.

Civera, Javier, Andrew J Davison, and JM Martinez Montiel (2008), "Inverse depth parametrization for monocular SLAM." *IEEE transactions on robotics*, 24, 932–945.

Collins, T and A Bartoli (2010), "Locally affine and planar deformable surface reconstruction from video." In *International Workshop on Vision, Modeling and Visualization*.

Collins, Toby and Adrien Bartoli (2015), “Realtime shape-from-template: System and applications.” In *ISMAR*, 116–119.

Concha, Alejo and Javier Civera (2015), “DPPTAM: Dense piecewise planar tracking and mapping from a monocular sequence.” In *IEEE/RSJ international conference on intelligent robots and systems*.

Dai, Yuchao, Hongdong Li, and Mingyi He (2014), “A simple prior-free method for non-rigid structure-from-motion factorization.” *International Journal of Computer Vision*, 107, 101–122.

Davison, Andrew J (2003), “Real-time simultaneous localisation and mapping with a single camera.” In *null*, 1403, IEEE.

Davison, Andrew J, Ian D Reid, Nicholas D Molton, and Olivier Stasse (2007), “MonoSLAM: Real-time single camera SLAM.” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29, 1052–1067.

Devernay, Frederic, Diana Mateus, and Matthieu Guilbert (2006), “Multi-camera scene flow by tracking 3-d points and surfels.” In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, volume 2, 2203–2212, IEEE.

Dosovitskiy, Alexey, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox (2015), “Flownet: Learning optical flow with convolutional networks.” In *Proceedings of the International Conference on Computer Vision*, 2758–2766.

Du, Xiaofei, Neil Clancy, Shobhit Arya, George B Hanna, John Kelly, Daniel S Elson, and Danail Stoyanov (2015), “Robust surface tracking combining features, intensity and illumination compensation.” *International Journal of Computer Assisted Radiology and Surgery*, 10, 1915–1926.

Eade, E. and T. Drummond (2006), “Scalable monocular slam.” In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, volume 1, 469–476.

Engel, Jakob, Vladlen Koltun, and Daniel Cremers (2017), “Direct sparse odometry.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Engel, Jakob, Thomas Schöps, and Daniel Cremers (2014), “LSD-SLAM: Large-scale direct monocular SLAM.” In *European Conference on Computer Vision*, 834–849, Springer.

Fayad, Joao, Lourdes Agapito, and Alessio Del Bue (2010), “Piecewise quadratic reconstruction of non-rigid surfaces from monocular sequences.” In *European conference on computer vision*, 297–310, Springer.

Floater, Michael S (2003), “Mean value coordinates.” *Computer Aided Geometric Design*, 20, 19–27.

Forster, Christian, Matia Pizzoli, and Davide Scaramuzza (2014), “SVO: Fast semi-direct monocular visual odometry.” In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 15–22.

Gallardo, Mathias, Toby Collins, and Adrien Bartoli (2016), "Can we jointly register and reconstruct creased surfaces by shape-from-template accurately?" In *ECCV*, 105–120, Springer.

Gallardo, Mathias, Daniel Pizarro, Toby Collins, and Adrien Bartoli (2020), "Shape-from-template with curves." *International Journal of Computer Vision*, 128, 121–165.

Gálvez-López, Dorian and J. D. Tardós (2012), "Bags of binary words for fast place recognition in image sequences." *IEEE Transactions on Robotics*, 28, 1188–1197.

Gálvez-López, Dorian and Juan D Tardos (2012), "Bags of binary words for fast place recognition in image sequences." *IEEE Transactions on Robotics*, 28, 1188–1197.

Gao, Wei and Russ Tedrake (2018), "Surfelwarp: Efficient non-volumetric single view dynamic reconstruction." In *Robotics: Science and System (RSS)*.

Garg, Ravi, Anastasios Roussos, and Lourdes Agapito (2013), "Dense variational reconstruction of non-rigid surfaces from monocular video." In *CVPR*.

Godard, Clément, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow (2019), "Digging into self-supervised monocular depth prediction."

Gotardo, Paulo FU and Aleix M Martinez (2011a), "Kernel non-rigid structure from motion." In *ICCV*.

Gotardo, Paulo FU and Aleix M Martinez (2011b), "Non-rigid structure from motion with complementary rank-3 spaces." In *CVPR*.

Grasa, Oscar G, Ernesto Bernal, Santiago Casado, Ismael Gil, and J.M.M. Montiel (2014), "Visual slam for handheld monocular endoscope." *IEEE Transactions on Medical Imaging*, 33, 135–146.

Grasa, Oscar G, Javier Civera, and JMM Montiel (????), "Ekf monocular slam with relocalization for laparoscopic sequences." In *ICRA*, 4816–4821, IEEE.

Innmann, Matthias, Michael Zollhöfer, Matthias Nießner, Christian Theobalt, and Marc Stamminger (2016), "Volumedeform: Real-time volumetric non-rigid reconstruction." In *ECCV*.

Jianbo Shi and Tomasi (1994), "Good features to track." In *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 593–600.

Kaess, Michael and Frank Dellaert (2009), "Covariance recovery from a square root information matrix for data association." *Robotics and autonomous systems*, 57, 1198–1210.

Kaess, Michael, Viorela Ila, Richard Roberts, and Frank Dellaert (2010), "The bayes tree: An algorithmic foundation for probabilistic robot mapping." In *Algorithmic Foundations of Robotics IX*, 157–173, Springer.

Kazhdan, Michael, Matthew Bolitho, and Hugues Hoppe (2006), "Poisson surface reconstruction." In *Eurographics*, 61–70, Eurographics Association.

Klein, Georg and David Murray (2007), “Parallel tracking and mapping for small AR workspaces.” In *ISMAR*, 225–234.

Kümmerle, Rainer, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard (2011), “g2o: A general framework for graph optimization.” In *ICRA*, IEEE.

Kurmann, Thomas, Pablo Marquez Neila, Xiaofei Du, Pascal Fua, Danail Stoyanov, Sebastian Wolf, and Raphael Sznitman (2017), “Simultaneous recognition and pose estimation of instruments in minimally invasive surgery.” In *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention*, 505–513, Springer.

Laina, Iro, Nicola Rieke, Christian Rupprecht, Josué Page Vizcaíno, Abouzar Eslami, Federico Tombari, and Nassir Navab (2017), “Concurrent segmentation and localization for tracking of surgical instruments.” In *Int. Conf. on medical image computing and computer-assisted intervention*, 664–672, Springer.

Lamarca, Jose and Jose Maria Martinez Montiel (2018a), “Camera tracking for SLAM in deformable maps.” In *European Conference on Computer Vision (ECCV)*.

Lamarca, Jose and Jose Maria Martinez Montiel (2018b), “Camera tracking for SLAM in deformable maps.” In *4th Inter. Workshop on Recovering 6D Object Pose. In ECCVw*.

Lamarca, Jose, Shaifali Parashar, Adrien Bartoli, and JMM Montiel (2020), “Defslam: Tracking and mapping of deforming scenes from monocular sequences.” *IEEE Transactions on robotics*, ?, ?–?

Lamarca, Jose, Juan J. Gomez Rodriguez, Juan D. Tardos, and J. M. M. Montiel (2021), “Direct and sparse deformable tracking.” *arXiv preprint arXiv:2109.07370*.

Lee, Minsik, Chong-Ho Choi, and Songhwai Oh (2014), “A procrustean markov process for non-rigid structure recovery.” In *CVPR*.

Lepetit, Vincent, Francesc Moreno-Noguer, and Pascal Fua (2009), “Epnp: An accurate o (n) solution to the pnp problem.” *International journal of computer vision*, 81, 155–166.

Lin, Bingxiong, Adrian Johnson, Xiaoning Qian, Jaime Sanchez, and Yu Sun (2013), “Simultaneous tracking, 3D reconstruction and deforming point detection for stereoscope guided surgery.” In *Augmented Reality Environments for Medical Imaging and Computer-Assisted Interventions*, 35–44, Springer.

Litany, Or, Emanuele Rodolà, Alexander M Bronstein, Michael M Bronstein, and Daniel Cremers (2016), “Non-rigid puzzles.” In *Computer Graphics Forum*, volume 35, 135–143, Wiley Online Library.

Liu, Xingtong, Yiping Zheng, Benjamin Killeen, Masaru Ishii, Gregory D Hager, Russell H Taylor, and Mathias Unberath (2020), “Extremely dense point correspondences using a learned feature descriptor.” In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4847–4856.

Lowe, David G (2004), “Distinctive image features from scale-invariant keypoints.” *International Journal of Computer Vision*, 60, 91–110.

Lucas, Bruce D and Takeo Kanade (1981), “An iterative image registration technique with an application to stereo vision.” In *Proc. 7th Intl. Joint Conf. on Artificial Intelligence (IJCAI)*.

Mahmoud, N, T Collins, A Hostettler, L Soler, C Doignon, and Jose Maria Martinez Montiel (2018), “Live tracking and dense reconstruction for hand-held monocular endoscopy.” *IEEE Transactions on Medical Imaging*.

Mahmoud, Nader, Iñigo Cirauqui, Alexandre Hostettler, Christophe Doignon, Luc Soler, Jacques Marescaux, and Jose Maria Martinez Montiel (2016), “ORBSLAM-based endoscope tracking and 3D reconstruction.” In *Int. Workshop on Computer-Assisted and Robotic Endoscopy*, 72–83, Springer.

Mahmoud, Nader, Alexandre Hostettler, Toby Collins, Luc Soler, Christophe Doignon, and Jose Maria Martinez Montiel (2017), “SLAM based quasi dense reconstruction for minimally invasive surgery scenes.” *arXiv preprint arXiv:1705.09107*.

Marmol, Andres, Artur Banach, and Thierry Peynot (2019), “Dense-arthroslam: Dense intra-articular 3-d reconstruction with robust localization prior for arthroscopy.” *IEEE Robotics and Automation Letters*, 4, 918–925.

Moreno-Noguer, Francesc and Josep M Porta (2011), “Probabilistic simultaneous pose and non-rigid shape recovery.” In *CVPR*.

Moreno-Noguer, Francesc, Josep M Porta, and Pascal Fua (2010), “Exploring ambiguities for monocular non-rigid shape estimation.” In *European Conference on Computer Vision*, 370–383, Springer.

Moreno-Noguer, Francesc, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua (2009), “Capturing 3d stretchable surfaces from single images in closed form.” In *CVPR*, 1842–1849, IEEE.

Mountney, Peter, Danail Stoyanov, and Guang-Zhong Yang (2010a), “Three-dimensional tissue deformation recovery and tracking.” *IEEE Signal Processing Magazine*, 27, 14–24.

Mountney, Peter, Danail Stoyanov, and Guang-Zhong Yang (2010b), “Three-dimensional tissue deformation recovery and tracking.” *IEEE Signal Processing Magazine*, 27, 14–24.

Mur-Artal, Raul, JMM Montiel, and Juan D Tardos (2015), “ORB-SLAM: a versatile and accurate monocular SLAM system.” *Robotics, IEEE Transactions on*, 31, 1147–1163.

Mur-Artal, Raul and Juan D Tardós (2017), “ORB-SLAM2: An open-source slam system for monocular, stereo, and RGB-D cameras.” *IEEE T-RO*.

Newcombe, Richard A, Dieter Fox, and Steven M Seitz (2015), “Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time.” In *CVPR*.

Newcombe, Richard A, Steven J Lovegrove, and Andrew J Davison (2011), “DTAM: Dense tracking and mapping in real-time.” In *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2320–2327, IEEE.

Ngo, Dat Tien, Jonas Östlund, and Pascal Fua (2016), "Template-based monocular 3D shape recovery using laplacian meshes." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38, 172–187.

Nocedal, Jorge and Stephen Wright (2006), *Numerical optimization*. Springer Science & Business Media.

Östlund, Jonas, Aydin Varol, Dat Tien Ngo, and Pascal Fua (2012), "Laplacian meshes for monocular 3D shape recovery." In *ECCV*, 412–425, Springer.

Özgür, Erol and Adrien Bartoli (2017), "Particle-sft: A provably-convergent, fast shape-from-template algorithm." *International Journal of Computer Vision*, 123, 184–205.

Pakhomov, Daniil, Vittal Premachandran, Max Allan, Mahdi Azizian, and Nassir Navab (2019), "Deep residual learning for instrument segmentation in robotic surgery." In *International Workshop on Machine Learning in Medical Imaging*, 566–573, Springer.

Paladini, Marco, Adrien Bartoli, and Lourdes Agapito (2010), "Sequential non-rigid structure-from-motion with the 3d-implicit low-rank shape model." In *ECCV*.

Paladini, Marco, Alessio Del Bue, Marko Stosic, Marija Dodig, Joao Xavier, and Lourdes Agapito (2009), "Factorization for non-rigid and articulated structure using metric projections." In *CVPR*.

Parashar, Shaifali, Daniel Pizarro, and Adrien Bartoli (2017), "Isometric non-rigid shape-from-motion with riemannian geometry solved in linear time." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40, 2442–2454.

Parashar, Shaifali, Daniel Pizarro, Adrien Bartoli, and Toby Collins (2015), "As-rigid-as-possible volumetric shape-from-template." In *ICCV*.

Perriollat, Mathieu, Richard Hartley, and Adrien Bartoli (2011), "Monocular template-based reconstruction of inextensible surfaces." *International journal of computer vision*, 95, 124–137.

Pizarro, Daniel, Rahat Khan, and Adrien Bartoli (2016), "Schwarps: Locally projective image warps based on 2d schwarzian derivatives." *International Journal of Computer Vision*, 119, 93–109.

Pratt, Philip, Danail Stoyanov, Marco Visentini-Scarzanella, and Guang Zhong Yang (2010), "Dynamic guidance for robotic surgery using image-constrained biomechanical models." In *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention*, 77–85, Springer.

Recasens, David, José Lamarca, José M. Fácil, J. M. M. Montiel, and Javier Civera (2021), "Endo-depth-and-motion: Reconstruction and tracking in endoscopic videos using depth networks and photometric constraints." *IEEE Robotics and Automation Letters*, 6, 7225–7232.

Rodríguez, Juan J Gómez, José Lamarca, Javier Morlana, Juan D Tardós, and José MM Montiel (2020), "Sd-defslam: Semi-direct monocular slam for deformable and intracorporeal scenes." *arXiv preprint arXiv:2010.09409*.

Rublee, Ethan, Vincent Rabaud, Kurt Konolige, and Gary Bradski (2011), "Orb: An efficient alternative to sift or surf." In *Computer Vision (ICCV), 2011 IEEE international conference on*, 2564–2571, IEEE.

Runz, Martin, Maud Buffier, and Lourdes Agapito (2018), "Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects." In *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 10–20, IEEE.

Russell, Chris, Joao Fayad, and Lourdes Agapito (2011), "Energy based multiple model fitting for non-rigid structure from motion." In *CVPR*, IEEE.

Salzmann, Mathieu and Pascal Fua (2011), "Linear local models for monocular reconstruction of deformable surfaces." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33, 931–944.

Salzmann, Mathieu, Francesc Moreno-Noguer, Vincent Lepetit, and Pascal Fua (2008), "Closed-form solution to non-rigid 3d surface registration." In *European conference on computer vision*, 581–594, Springer.

Shvets, Alexey A, Alexander Rakhlin, Alexandr A Kalinin, and Vladimir I Iglovikov (2018), "Automatic instrument segmentation in robot-assisted surgery using deep learning." In *IEEE Int. Conf. on Machine Learning and Applications (ICMLA)*, 624–628.

Song, Jingwei, Jun Wang, Liang Zhao, Shoudong Huang, and Gamini Dissanayake (2017), "Dynamic reconstruction of deformable soft-tissue with stereo scope in minimal invasive surgery." *IEEE Robotics and Automation Letters*, 3, 155–162.

Song, Jingwei, Jun Wang, Liang Zhao, Shoudong Huang, and Gamini Dissanayake (2018), "Mis-slam: Real-time large-scale dense deformable slam system in minimal invasive surgery based on heterogeneous computing." *IEEE Robotics and Automation Letters*, 3, 4068–4075.

Sorkine, Olga and Marc Alexa (2007), "As-rigid-as-possible surface modeling." In *Eurographics*.

Stoyanov, Danail, George P Mylonas, Fani Deligianni, Ara Darzi, and Guang Zhong Yang (2005), "Soft-tissue motion tracking and structure estimation for robotic assisted mis procedures." In *MICAI*.

Stoyanov, Danail, Marco Visentini Scarzanella, Philip Pratt, and Guang-Zhong Yang (2010), "Real-time stereo reconstruction in robotically assisted minimally invasive surgery." In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 275–282, Springer.

Strasdat, Hauke, Andrew J Davison, JM Martìnez Montiel, and Kurt Konolige (2011), "Double window optimisation for constant time visual slam." In *2011 international conference on computer vision*, 2352–2359, IEEE.

Strasdat, Hauke, JMM Montiel, and Andrew J Davison (2010), "Real-time monocular slam: Why filter?" In *2010 IEEE International Conference on Robotics and Automation*, 2657–2664, IEEE.

Strasdat, Hauke, José MM Montiel, and Andrew J Davison (2012), “Visual slam: why filter?” *Image and Vision Computing*, 30, 65–77.

Sumner, Robert W, Johannes Schmid, and Mark Pauly (2007), “Embedded deformation for shape manipulation.” In *ACM SIGGRAPH 2007 papers*, 80–es.

Taylor, Jonathan, Allan D Jepson, and Kiriakos N Kutulakos (2010), “Non-rigid structure from locally-rigid motion.” In *CVPR*.

Torresani, Lorenzo, Aaron Hertzmann, and Chris Bregler (2008), “Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30, 878–892.

Torresani, Lorenzo, Aaron Hertzmann, and Christoph Bregler (2004), “Learning non-rigid 3d shape from 2d motion.” In *Advances in Neural Information Processing Systems*, 1555–1562.

Triggs, Bill, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon (1999), “Bundle adjustment—a modern synthesis.” In *International workshop on vision algorithms*, 298–372, Springer.

Varol, Aydin, Mathieu Salzmann, Pascal Fua, and Raquel Urtasun (2012), “A constrained latent variable model.” In *2012 IEEE conference on computer vision and pattern recognition*, 2248–2255, Ieee.

Varol, Aydin, Mathieu Salzmann, Engin Tola, and Pascal Fua (2009), “Template-free monocular reconstruction of deformable surfaces.” In *ICCV*.

Vicente, Sara and Lourdes Agapito (2012), “Soft inextensibility constraints for template-free non-rigid reconstruction.” In *ECCV*.

Wang, Zhou, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli (2004), “Image quality assessment: from error visibility to structural similarity.” *IEEE Trans. on image processing*, 13, 600–612.

White, Ryan and David A Forsyth (2006), “Combining cues: Shape from shading and texture.” In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, volume 2, 1809–1816, IEEE.

Xiao, Jing, Jin-xiang Chai, and Takeo Kanade (2004), “A closed-form solution to non-rigid shape and motion recovery.” In *ECCV*, Springer.

Yu, Rui, Chris Russell, Neill DF Campbell, and Lourdes Agapito (2015), “Direct, dense, and deformable: template-based non-rigid 3d reconstruction from rgb video.” In *ICCV*, 918–926.

Zubizarreta, Jon, Iker Aguinaga, and Jose Maria Martinez Montiel (2020), “Direct sparse mapping.” *IEEE Transactions on Robotics*, 36, 1363–1370.

Appendix A

DefSLAM: Derivatives of Regularizers of the Deformable Tracking

We show the Jacobian terms of the regularizers to prove that it does not have singularities:

Stretching The streching error $e_s(\mathcal{L}_t^k, \mathcal{T}_k)_e$ for the edge e is:

$$e_s(\mathcal{L}_t^k, \mathcal{T}_k)_e = \left(\frac{l_e^t - l_e^k}{l_e^k} \right), \tag{A.1}$$

being

$$l_e^t = \|(\mathbf{V}_{e^1}^t - \mathbf{V}_{e^2}^t)\|_2 \tag{A.2}$$

where $\mathbf{V}_{e^1}^t$ and $\mathbf{V}_{e^2}^t$ are the two nodes of the edge e in the instant t. Its derivative is

$$\frac{\partial e_s(\mathcal{L}_t^k, \mathcal{T}_k)_e}{\partial \mathbf{V}_{e^i}^t} = \frac{(\mathbf{V}_{e^1}^t - \mathbf{V}_{e^2}^t)}{l_e^k l_e^t} \tag{A.3}$$

Bending The bending error $e_b(\mathcal{L}_k^t, \mathcal{T}_k)_n$ for the node n connected with its neighbours $\mathbf{V}_l \in \mathcal{N}_j$ through the edge e_l is:

$$e_b(\mathcal{L}_k^t, \mathcal{T}_k)_n = \frac{\delta_n^t - \delta_n^k}{l_{e_l}^k}. \tag{A.4}$$

where $\overrightarrow{\delta}_n^t$ is the mean curvature of the surface at the instant t. It is estimated though the neighbours of the node and itself.

$$\overrightarrow{\delta}_n^t = \mathbf{V}_n^t - \frac{1}{\sum_{l \in \mathcal{N}_j} \omega_l} \sum_{l \in \mathcal{N}_j} \omega_l \mathbf{V}_l^i \tag{A.5}$$

$$\delta_n^t = \|\overrightarrow{\delta}_n^t\|_2 \tag{A.6}$$

We assume fixed the values of the weights. Its derivative respect the node $\mathbf{V}_n^t$ is:

$$\frac{\partial e_b(\mathcal{L}_k^t, \mathcal{T}_k)_n}{\partial \mathbf{V}_n^t} = \frac{\overrightarrow{\delta}_n^t}{l_{e_l}^k \delta_n^t} \tag{A.7}$$

with respect to its neighbours $\mathbf{V}_l^t$

$$\frac{\partial e_b(\mathcal{L}_k^t, \mathcal{T}_k)_n}{\partial \mathbf{V}_l^t} = \frac{\omega_l}{\sum_{l \in \mathcal{N}_j} \omega_l} \frac{\overrightarrow{\delta}_t^n}{l_{e_l}^k \delta_t^n} \tag{A.8}$$

In case of being a plane, the mean curvature and its derivative tends to zero.

$$\frac{\partial e_b(\mathcal{L}_k^t, \mathcal{T}_k)_n}{\partial \mathbf{V}_n^t} = 0, \quad \delta_t^n = 0 \tag{A.9}$$

Reference The reference error is

$$e_r(\mathcal{L}_k^t, \mathcal{L}_k^k) = \mathbf{V}_n^t - \mathbf{V}_n^k. \tag{A.10}$$

and its derivative:

$$\frac{\partial e_r(\mathcal{L}_k^t, \mathcal{T}_k)}{\partial \mathbf{V}_n^t} = 1. \tag{A.11}$$

Appendix B

Direct and Sparse Deformable Tracking. Jacobian equations

In this section we develop the Jacobians used for the optimizations in Eq. 5.13 and Eq. 5.21. To estimate them, we apply the chain rule to each of the residual terms: the photometric error, the deformation energy and the equilibrium point constrain.

The photometric error was modeled through forward-compositional illumination-invariant approach as in Eq. 5.5 for each pixel p:

$$\mathcal{P}_{i,p}^{t} = \alpha_i^t T(u_p, v_p) - I(\pi(\mathbf{T}_{cw}\mathbf{S}_i^t(u_p, v_p)) + \beta_i^t. \tag{B.1}$$

It depends on three groups of parameters: the deformation of the surfel, the camera pose, and the initial parameters. This parameters are optimized depending on the problem.

In the Direct and Sparse Deformable Tracking with Static Camera, only the movement of the surfel is optimized through its translation $\mathbf{t}_i^t$, rotation $\mathbf{R}_i^t$ and the deformation. The initial parameters of the surfel, $\mathbf{X}_i^0$ and $\mathbf{J}_i^0$, are known and the pose of the camera is fixed $\mathbf{T}_{cw}$.

$$\mathbf{S}_i^t = (\mathbf{X}_i^0 + \mathbf{t}_i^t) + \mathbf{R}_i^t \mathbf{J}_i^0 \mathbf{F}_i^t \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} \tag{B.2}$$

The surfel is transformed to the camera reference by:

$$\mathbf{S}_{ic}^t = \mathbf{T}_{cw}\mathbf{S}_i^t \tag{B.3}$$

The derivatives of the photometric error *wrt.* the pose of the camera can be defined through the chain rule:

$$\frac{\partial \mathcal{P}_i^t}{\partial \mathbf{X}} = -\nabla I \frac{\partial \pi}{\partial \mathbf{S}_{ic}^t} \mathbf{R}_{cw} \frac{\partial \mathbf{S}_i^t}{\partial \mathbf{X}} \tag{B.4}$$

being ∇I the gradient of the image, it is estimated with a central finite differences kernel $\begin{bmatrix}-1 & 0 & 1\end{bmatrix}$. $\mathbf{R}_{cw}$ is the rotation of the camera which images the surfel. For the SfT, we assume that the fixed camera is in the origin, so $\mathbf{T}_{cw} = \mathbb{I}_4$.

$\frac{\partial \pi}{\partial \mathbf{S}_{ic}^t}$ is the projection derivative. It is estimated by using the position of the point $\begin{bmatrix}X_c Y_c Z_c\end{bmatrix}^\top$ and the intrinsic calibration of the camera -focal length f_x, f_y- in the camera reference as:

$$\frac{\partial \pi}{\partial S_{ic}^t} = \begin{bmatrix} \frac{f_x}{Z_c} & 0 & -\frac{f_x X_c}{Z_c^2} \\ 0 & \frac{f_y}{Z_c} & -\frac{f_y Y_c}{Z_c^2} \end{bmatrix}. \tag{B.5}$$

$\frac{\partial \mathbf{S}_i^t}{\partial \mathbf{X}}$ depends on the translation, rotation and deformation of the surfel. The jacobian obtained for this derivative is a 3×6 or 3×9 matrix.

$$\frac{\partial \mathbf{S}_i^t}{\partial \mathbf{t_i}} = \mathbb{I}_3. \tag{B.6}$$

$$\frac{\partial \mathbf{S}_i^t}{\partial \mathbf{R_i^t}} = -[\mathbf{R}_i^t \mathbf{J}_i^0 \mathbf{F}_i^t \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}]_\times \tag{B.7}$$

.

$$\frac{\partial S_i^t}{\partial \mathbf{F_i^t}} = \mathbf{R}_i^t \mathbf{J}_i^0 \begin{bmatrix} \Delta u & 0 & \Delta u \\ 0 & \Delta v & \Delta v \end{bmatrix}. \tag{B.8}$$

For the deformation tracking presented in Sec.5.5 the camera is not static and its pose is estimated. The derivative of the photometric error *wrt.* the camera motion is:

$$\frac{\partial \mathcal{P}_i}{\partial T_{cw}} = -\nabla I \frac{\partial \pi}{\partial S_{i\,c}^t} \begin{bmatrix} \mathbb{I}_3 & -[S_{i\,c}^t]_\times \end{bmatrix}. \tag{B.9}$$

α is static and only reestimated at the beginning of the optimization with the initial guess of the coarsest scale. It is enough with multiply ∇I by the α gain.

When isometry is soft-constrained, we use a deformation energy to constrain the deformations. Deformation are modeled by the symmetric matrix $\mathbb{F}_i^t$ and the deformation energy is:

$$\mathcal{I}_i^t = \left\| \mathbf{F}_i^t - \mathbb{I}_2 \right\|_2^2. \tag{5.12}$$

The derivatives depending to model the deformation energy are:

$$\frac{\partial \mathcal{I}_i^t}{\partial \mathbb{F}} = \mathbb{I}_3 \tag{B.10}$$

The equilibrium point residuals (Eq. 5.20 for deformation tracking). The Eq. 5.20) derivatives used for the optimization are:

$$\frac{\partial \mathcal{E}_i^t}{\partial \mathbf{t}_i^t} = \mathbb{I}_3. \tag{B.11}$$

Appendix C

Conclusiones en Español

En esta tesis hemos concebido el primer SLAM Monocular capaz de trabajar en escenarios con deformación. Hemos iniciado una nueva rama de investigación de SLAM que puede trabajar en escenarios más generales. Hemos demostrado que los métodos deformables propuestos son necesarios y superan a los tradicionales.

Nuestra primera contribución fue la localización deformable presentada en el Capítulo 2. En esta primera versión del sistema, hemos analizado cómo se debe tratar un mapa deformable. Introducimos un método de optimización basado en modelos físicos, inspirado en los métodos basados en plantillas que normalmente tratan un objeto pequeño que cabe en el campo de visión de la cámara. Conseguimos recuperar la forma del objeto minimizando simultáneamente el error de reproyección y la energía de deformación. El primer desafío de nuestro método fue procesar solo vistas parciales del mapa. Vimos que recuperar toda la deformación del mapa no era ventajoso de ninguna manera, en cambio es mucho mejor recuperar solo el área observada. La principal desventaja de este método es que necesita conocer la forma en reposo del mapa de antemano.

En el Capítulo 3, proponemos complementar la localización deformable con un mapeo deformable. Este mapeo deformable fue capaz de mapear la escena incluso con deformaciones para un conjunto de imágenes monoculares. Sustituye los métodos rígidos clásicos, como el ajuste de haces, por el NRSfM de Parashar et al. (2017) que reconstruye superficies asumiendo isometría, planaridad infinitesimal y continuidad. El método NRSfM fue concebido para tratar objetos pequeños como almohadas, papeles o telas. En nuestro trabajo, diseñamos el mapeo deformable con los mismos supuestos, pero con tres características principales destacadas: tiene asociación automática de datos, es un método incremental y es capaz de explorar nuevas regiones de la escena manteniendo una escala coherente del mapa. Al juntar la localización deformable y el mapeo deformable en una estructura similar a PTAM, nuestro método es capaz de procesar con éxito el Mandala dataset y algunas secuencias médicas logrando resultados estado del arte y en tiempo real. Así, podemos concluir que DefSLAM es el primer sistema SLAM monocular deformable.

En el Capítulo 4, mejoramos DefSLAM para hacerlo más robusto en escenarios médicos. Su principal inconveniente para estas secuencias fue el emparejamiento basada en características, específicamente el método ORB. El método ORB utiliza un extractor FAST que no es muy repetible, lo que da lugar a un número considerable de características no observadas entre imagenes. Lo sustituimos por un rastreador Lukas-Kanade que rastrea los puntos del mapa mediante el uso de un error fotométrico. Se demostró que este método supera al emparejamiento anterior tanto para el Mandala como para las secuencias médicas, tanto en términos de precisión como de robustez. Gracias a evitar la extracción de keypoints por frame, el sistema se comporta con un tiempo computacional similar. Además, incluimos un módulo de relocalización en caso de fallo de seguimiento y un módulo de segmentación para eliminar objetos dinámicos no relacionados con la escena. Gracias a todas las mejoras propuestas pudimos procesar secuencias de colonoscopia con deformación.

El problema principal de ambos sistemas viene con el supuesto de continuidad e isometría. Estos supuestos son bastante restrictivos y, aunque aproximadamente funcionan bien en secuencias casi isométricas, este no es el caso de muchas secuencias médicas. Por eso, en nuestro último Capítulo 5, abordamos el problema de la reconstrucción de deformaciones no isométricas. Para lograr nuestro propósito, aprovechamos el error fotométrico para rastrear los puntos individualmente sin definir un modelo de deformación global. Gracias a esta nueva formulación podemos captar mucho mejor la naturaleza de las deformaciones observadas en la escena médica. En nuestros experimentos demostramos que podemos reconstruir deformaciones no isométricas o superficies con cambios topológicos que superan los métodos de localización deformables propuestos en los capítulos anteriores.

Como conclusión general de esta tesis: hemos propuesto el primer sistema capaz de funcionar en escenarios deformables. Además, también hemos hecho aportaciones dentro de los actuales métodos monoculares no rígidos. En la primera propuesta, hemos desarrollado un método de localización de cámara basado en plantillas capaz de trabajar con mapas parcialmente observados. En el segundo conjunto de propuestas, hemos desarrollado un nuevo mapeo deformable heredado del NRSfM capaz de realizar exploraciones. Hacemos que nuestro método sea más robusto explotando el error fotométrico e incluyendo nuevas técnicas para el reconocimiento de lugares y la segmentación para SLAM dinámico. Finalmente, hemos desarrollado un nuevo método de seguimiento deformable capaz de procesar deformaciones casi isométricas y no isométricas de forma más fiel, dando resultados prometedores y que sirven de base para futuros métodos SLAM monoculares deformables.

C.1 Limitaciones y Trabajo Futuro

En esta tesis, hemos explorado métodos geométricos para crear un nuevo método SLAM capaz de abordar las escenas con deformación. Para poder reconstruir mapas deformables, tuvimos que hacer varias suposiciones e incorporar el modelo físico en las ecuaciones del SLAM monocular. Estos modelos son más restrictivos que en el caso rígido, por ejemplo, asumiendo superficies continuas e isométricas. Para tener un método completamente robusto y aplicable, esas restricciones deben ser relajadas. En el capítulo 5, hemos visto que es posible recuperar la deformación del mapa y la pose de la cámara solo con restricciones locales. A partir de ese punto, consideramos que el siguiente paso es construir un mapeo deformable más versátil capaz de funcionar sin asumir continuidad. Esto es crucial para construir mapas que no sean totalmente uniformes o con topología no plana de la superficie reconstruida.

Finalmente, nos hemos centrado en la formulación geométrica dejando los enfoques basados en aprendizaje como un tema secundario. Hemos visto las mejoras que estos enfoques pueden aportar al campo en varios casos, por ejemplo gracias a la segmentación o al flujo óptico. En el trabajo de fin de grado de Diego Royo 2020, concluimos que el flujo óptico con el Flownet propuesto por Dosovitskiy et al. (2015) se puede utilizar como sustituto del método Schwarp utilizado en la primera etapa del mapeo deformable, logrando deformaciones irregulares y escalonadas. En el Capítulo 4, usamos una red de segmentación para ignorar las herramientas que entran en el campo de visión de la cámara y no son parte del mapa. La última colaboración con David Recasens en (Recasens et al., 2021) exploramos el uso de Monodepth2 (Godard et al., 2019) como Mapeo para el sistema ya que puede generar profundidad a partir de imágenes individuales con resultados similares a los obtenidos con métodos geométricos. Esa es solo una pequeña prueba del impacto potencial que el aprendizaje profundo puede traer a este área.

Hemos centrado este manuscrito en la idea de concebir un algoritmo SLAM capaz de abordar deformaciones de secuencias en el cuerpo. Un concepto subyacente importante presentado es la idea de un mapa cambiante. En SLAM para aplicaciones en escenarios no estáticos, como un bote en medio del océano o en una calle donde se mueven los coches y peatones, se necesitará un mapa con un modelo matemático que pueda representar la evolución del mapa para tener un desempeño robusto y fiable.