Dense RGB-D-Inertial SLAM with Map Deformations

Tristan Laidlow, Michael Bloesch, Wenbin Li and Stefan Leutenegger

***Abstract*— While dense visual SLAM methods are capable of estimating dense reconstructions of the environment, they suffer from a lack of robustness in their tracking step, especially when the optimisation is poorly initialised. Sparse visual SLAM systems have attained high levels of accuracy and robustness through the inclusion of inertial measurements in a tightly-coupled fusion. Inspired by this performance, we propose the first tightly-coupled dense RGB-D-inertial SLAM system.**

Our system has real-time capability while running on a GPU. It jointly optimises for the camera pose, velocity, IMU biases and gravity direction while building up a globally consistent, fully dense surfel-based 3D reconstruction of the environment. Through a series of experiments on both synthetic and real world datasets, we show that our dense visual-inertial SLAM system is more robust to fast motions and periods of low texture and low geometric variation than a related RGB-D-only SLAM system.

I. INTRODUCTION

Visual Simultaneous Localisation and Mapping (SLAM) has achieved a level of maturity that allows for integration into mobile robots. We can split respective algorithms into two broad categories: sparse landmark-based systems and dense or semi-dense systems. While sparse methods may not directly produce a map that is useful for robot navigation, pose estimation quality and robustness of state-of-the-art systems, such as [1] and [2], are typically very high. Even higher accuracy and robustness may be attained by the inclusion of inertial measurements in a tightly-coupled fusion. Inertial Measurement Units (IMUs) have become very cheap and are abundant in today's consumer electronic devices, therefore their use in visual SLAM has been widely adopted. Approaches are formulated either as filters e.g. [3]–[6] or as methods employing iterative minimisation, typically in a sliding window manner, such as [7]–[10]. Loosely-coupled approaches to visual-inertial fusion, such as [11]–[13] that separate out either the visual or inertial estimation part have also been proposed. These methods are popular due to their modularity, but disregard correlations in the state estimates, typically leading to lower accuracy and/or robustness.

Other research has focused on producing denser maps, a development enabled by ever more computational power and specifically the emergence of Graphics Processing Units (GPUs), as well as by novel sensors in the form of depth cameras (RGB-D cameras). Such systems typically employ direct photometric alignment of the image and/or Iterative Closest Point (ICP) alignment of the depth image to the map in a *tracking* step; in a separate *mapping* step, new information is fused into the dense map representation. Examples using solely monocular cameras range from the fully dense DTAM [14] to LSD-SLAM [15], which reconstructs a semi-dense representation. RGB-D SLAM approaches that make use of depth cameras include KinectFusion [16] and other methods employing signed distance function-based volumetric mapping, such as [17]–[19]. Surfel-based mapping presents itself as an alternative enabling easier scalability in space and time: ElasticFusion [20], which our proposed work is based on, focuses on global map consistency by applying elastic map deformations upon loop closure.

Fig. 1: Tightly integrating IMU measurements into a dense RGB-D SLAM system leads to more accurate and robust tracking and 3D reconstruction compared with using visual information alone. The above surface reconstruction was captured despite periods of low texture and geometric variation.

Dense maps offer much more potential for safe robot navigation which we envisage to evolve into very general spatial perception with semantic understanding and tracking of dynamic objects in the future. As of now, however, vision-only SLAM, and dense SLAM using direct image alignment in particular, suffers from a lack of robustness in the tracking step when initialised too far from the "true" solution; in fact, the tracking optimisation may not converge at all in absence of sufficient texture and/or geometric variation in the depth channel. To address these shortcomings, inspired by the success of sparse visual-inertial systems, we advocate the integration of acceleration and rotation rate measurements into the tracking of a dense SLAM system. In principle, the tight integration of these complementary sensing modalities

The authors are with the Dyson Robotics Laboratory, Imperial College London, UK. Corresponding author: Tristan Laidlow, t.laidlow15@imperial.ac.uk

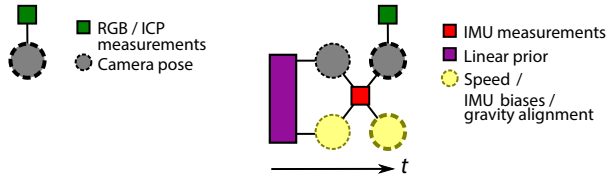
Research presented in this paper has been supported by Dyson Technology Ltd.

should provide robustness in rapid motion, low texture and flat walls. Furthermore, the inclusion of an IMU renders the gravity direction observable, which not only improves map accuracy due to bounded absolute inclination error, but may also be of paramount importance for robot control, most prominently drones.

There have been a few recent examples of dense visual-inertial systems: both [21] and [22] present loosely-coupled approaches, with the former using the integrated IMU data as a prediction step in a filter to estimate the transformation between image pairs, and the latter fusing relative poses generated by inertial and stereo camera measurements in a manner similar to a pose graph. A tightly-coupled semi-dense monocular visual-inertial odometry system is presented in [23]. Unlike other pure monocular odometry systems, it is able to use the inertial data to remove scale ambiguity. Their system uses a semi-dense approach for tracking and, in a separate thread, estimates a fully dense map below frame rate using a piecewise planar prior. Another example of a semi-dense visual-inertial odometry system is described in [24]. This system is implemented within the stereo LSD-SLAM framework [25]. Through a series of experiments, they demonstrate that their tightly-coupled approach outperforms both vision-only and loosely-coupled approaches. While the system is closely related to ours, we propose a more map-centric, fully dense approach that additionally considers a depth channel and performs map optimisation compliant with gravity alignment.

In this paper, we extend the RGB-D SLAM system ElasticFusion [20] with tightly-coupled IMU integration, which is capable of more accurate and robust fully dense mapping. Please see Figure 1 for an example map output. More specifically, we make the following contributions:

- In the tracking step, we simultaneously estimate the camera pose, velocity, IMU biases and gravity direction from an RGB-D camera and IMU by minimising a joint photometric, geometric, and inertial energy functional.
- Concerning the mapping, we propose a system that constructs a globally consistent, fully dense surfel-based 3D reconstruction of the environment. The map is optimised not through a pose graph, but by applying non-rigid space deformations using a sparse deformation graph. We propose an addition to the deformation energy that ensures consistency with the observable gravity direction.
- Through experiments on both synthetic and real world datasets, we demonstrate the benefits of our approach. It performs well under aggressive motion, fast rotations, and under low texture and geometric variation. We demonstrate trajectory and map reconstruction accuracy higher or on-par with an RGB-D-only ElasticFusion.
- We emphasise that the system maintains real-time capability while running on a GPU. Unlike [23] and [24], which achieve real-time performance on a CPU, our system constructs a fully dense map at frame rate.
- To the best of our knowledge, we hereby present the first tightly-coupled dense RGB-D-inertial SLAM system.



(a) RGB-D-only ElasticFusion (b) RGB-D-inertial ElasticFusion

Fig. 2: Tracking optimisation in RGB-D-only vs. RGB-D-inertial ElasticFusion: inertial measurements necessitate the augmentation of the state with speed, IMU biases, and gravity alignment; furthermore, the temporal nature of IMU measurements requires us to marginalise old states, resulting in a linear prior.

We furthermore plan to release our software framework.

The remainder of this paper is organised as follows: we start with an overview of notation employed in Section II and of the approach as such in III. We then describe the method concerning tracking in IV and mapping in V followed by extensive results in VI.

II. NOTATION

Throughout this work we will employ the following notation: a reference frame A is denoted $\underrightarrow{\mathcal{F}}_A$, with vectors expressed in it denoted as ${}_A\mathbf{p}$. The position vector from the origin of $\underrightarrow{\mathcal{F}}_A$ to the origin of $\underrightarrow{\mathcal{F}}_B$, represented in $\underrightarrow{\mathcal{F}}_A$ is written ${}_A\mathbf{r}_B$ and the velocity of the origin of $\underrightarrow{\mathcal{F}}_C$ as observed by $\underrightarrow{\mathcal{F}}_B$ and expressed in $\underrightarrow{\mathcal{F}}_A$ is denoted ${}_A\mathbf{v}_{BC}$. The homogeneous transformation matrix that transforms homogeneous points from $\underrightarrow{\mathcal{F}}_B$ to $\underrightarrow{\mathcal{F}}_A$ is written as $\boldsymbol{T}_{AB}$. The corresponding rotation is represented by a Hamiltonian unit quaternion, $\mathbf{q}_{AB}$. In order to refer to the homogeneous coordinates of a coordinate vector $\mathbf{p}$ we will use the italic notation $\boldsymbol{p}$.

Four different coordinate frames will be used in this work:

- $\underrightarrow{\mathcal{F}}_W$, the world frame in which the global model is expressed. This frame corresponds with the initial camera frame.
- $\underrightarrow{\mathcal{F}}_I$, the inertial frame that is aligned with gravity and shares an origin with $\underrightarrow{\mathcal{F}}_W$.
- $\underrightarrow{\mathcal{F}}_C$, the camera frame in which the RGB-D data is observed.
- $\underrightarrow{\mathcal{F}}_S$, the sensor frame in which the IMU data is observed.

III. SYSTEM OVERVIEW

Our system directly builds upon the vision-only dense RGB-D tracking and mapping approach of ElasticFusion [20]. Like ElasticFusion, our approach performs the tracking and mapping in separate steps. In the tracking step, a joint photometric, geometric and inertial energy functional is constructed. Please see Figure 2 for a comparison of the factor-graph representation of the underlying tracking optimisation problem in RGB-D-only and RGB-D-inertial ElasticFusion. Whereas ElasticFusion combined the photometric and geometric terms based on a tuning parameter λ, we combine the terms based on the covariances associated with the

measurement noise. A nonlinear optimisation formulation is then used to simultaneously estimate the camera pose, velocity, IMU biases and gravity direction. Unlike the original ElasticFusion which only estimated the current camera pose, our system estimates the states associated with both the current and previous camera frames. After the optimisation the states related to the previous frame are marginalised and the remaining current state is used as a prior for the next time step.

In the mapping step, a fully dense surfel-based surface representation is constructed from the camera data and estimated poses obtained from the tracking step. The map is kept globally consistent by applying non-rigid space deformations through a sparse deformation graph. We extend the deformation energy formulation proposed by ElasticFusion to ensure consistency with the observable gravity direction.

IV. TRACKING

A. States & Local Parameterisation

At the arrival of each new camera frame, we estimate the current state, $\mathbf{x}_1$, while simultaneously refining the previous state, $\mathbf{x}_0$. The system state is comprised of the camera position in the world frame ${}_W\mathbf{r}_C$, the camera orientation $\mathbf{q}_{WC}$, the velocity of the IMU in the inertial frame ${}_I\mathbf{v}_{IS}$, the biases of the gyroscopes $\mathbf{b}_g$ and accelerometers $\mathbf{b}_a$, and the orientation of the world frame in the inertial frame $\mathbf{q}_{IW}$. Therefore the system state $\mathbf{x}$ for a specific time instance is given by:

$$\begin{aligned}\mathbf{x} := &\left[{}_W\mathbf{r}_C^T, \mathbf{q}_{WC}^T, {}_I\mathbf{v}_{IS}^T, \mathbf{b}_g^T, \mathbf{b}_a^T, \mathbf{q}_{IW}^T\right]^T \\ &\in \mathbb{R}^3 \times S^3 \times \mathbb{R}^9 \times S^3.\end{aligned} \tag{1}$$

While only two degrees of freedom are required to express the gravity direction in the world frame, for simplicity, we use a 3D implementation with gauge freedom. We did not observe any issues related to this formulation.

The system state exists on a manifold and so is updated by a local perturbation $\delta\mathbf{x}$ in the tangent space through the $\boxplus$ operator, such that $\mathbf{x} = \bar{\mathbf{x}} \boxplus \delta\mathbf{x}$ around a reference $\bar{\mathbf{x}}$. For ${}_W\mathbf{r}_C$, ${}_I\mathbf{v}_{IS}$, $\mathbf{b}_g$ and $\mathbf{b}_a$, the $\boxplus$ operator is equivalent to standard vector addition. For $\mathbf{q}_{WC}$ and $\mathbf{q}_{IW}$, a combination of the group operator (quaternion multiplication) and exponential map is used ($\mathbf{q} \boxplus \delta\boldsymbol{\alpha} = \exp(\delta\boldsymbol{\alpha}) \otimes \mathbf{q}$). This results in the following minimal local coordinate representation:

$$\delta\boldsymbol{x} = \left[\delta\mathbf{r}^T, \delta\boldsymbol{\alpha}^T, \delta\mathbf{v}^T, \delta\mathbf{b}_g^T, \delta\mathbf{b}_a^T, \delta\mathbf{g}^T\right]^T \in \mathbb{R}^{18}. \tag{2}$$

Similarly, a $\boxminus$ operator can be introduced to compute the difference between two systems states. For regular vector space quantities this corresponds to standard subtraction. For orientations an inverse of the above $\boxplus$ can be constructed ($\mathbf{p} \boxminus \mathbf{q} = \exp^{-1}(\mathbf{p} \otimes \mathbf{q}^{-1})$).

Please refer to [26], [27] for further details.

B. Dense Photometric & Geometric Alignment

The RGB-D subsystem of our approach combines dense per-pixel photometric alignment with ICP point-to-plane geometric alignment. The photometric alignment error, $e_{\mathrm{RGB},\mathbf{u}}$ for pixel $\mathbf{u}$ in the current image is the intensity difference between the transformed previous and current images:

$$e_{\mathrm{RGB},\mathbf{u}} = I_0\left(\boldsymbol{\pi}\left(\mathbf{K}\,\boldsymbol{T}_{WC_0}^{-1}\,\boldsymbol{T}_{WC_1}\,\boldsymbol{\rho}(\boldsymbol{u}, d)\right)\right) - I_1(\boldsymbol{u}), \tag{3}$$

where $I_*(\cdot)$ is a scalar function that returns the intensity value of a given pixel, $\boldsymbol{\pi}(\cdot)$ is the projection and dehomogenisation function that maps a 3D point onto the image plane, and $\boldsymbol{\rho}(\mathbf{u}, d)$ is the back-projection function that returns a homogeneous 3D point for pixel $\mathbf{u}$ with a depth d. $\mathbf{K}$ is the camera intrinsics matrix, containing the focal lengths and principal point of the camera.

The geometric alignment error uses a point-to-plane ICP technique and computes the signed distance between a point $\mathbf{p}_k$ projected from the depth measurement k as viewed from the current camera pose and a corresponding point in the global model:

$$e_{\mathrm{ICP},k} = {}_W\mathbf{n}_k^T\left[\boldsymbol{T}_{WC_1}\,{}_{C_1}\boldsymbol{p}_k - {}_W\boldsymbol{p}_k\right]_{1:3}. \tag{4}$$

C. Inertial Integration

For the formulation of the IMU measurement error term we adopt the approach of [9], extending it to include the preintegration technique described by [10]. The IMU measurements are integrated numerically between the previous and current camera frames. The final IMU error term is given by:

$$\mathbf{e}_{\mathrm{IMU}} = \hat{\mathbf{x}}_1(\mathbf{x}_0) \boxminus \mathbf{x}_1, \tag{5}$$

where $\hat{\mathbf{x}}_1$ is the prediction of the current state by integrating the IMU measurements onto the previous state.

D. Optimisation

The RGB-D-inertial tracking problem is solved using a joint cost function c_{track} that contains the weighted photometric alignment, geometric alignment and inertial terms:

$$\begin{aligned}c_{\mathrm{track}}(\mathbf{x}_0, \mathbf{x}_1) = &\sum_{\mathbf{u}} e_{\mathrm{RGB},\mathbf{u}} W_{\mathrm{RGB}} e_{\mathrm{RGB},\mathbf{u}} \\ &+ \sum_k e_{\mathrm{ICP},k} W_{\mathrm{ICP},k} e_{\mathrm{ICP},k} + \mathbf{e}_{\mathrm{IMU}}^T \mathbf{W}_{\mathrm{IMU}} \mathbf{e}_{\mathrm{IMU}} \\ &+ \left(\mathbf{x}_0 \boxminus \bar{\mathbf{x}}_0 - \mathbf{H}^{*-1}\mathbf{b}^*\right)^T \mathbf{H}^* \left(\mathbf{x}_0 \boxminus \bar{\mathbf{x}}_0 - \mathbf{H}^{*-1}\mathbf{b}^*\right),\end{aligned} \tag{6}$$

where W_{RGB}, $W_{\mathrm{ICP},k}$, and $\mathbf{W}_{\mathrm{IMU}}$ are the inverse covariance (matrices) associated with the respective measurement uncertainties, and $\mathbf{H}^*$ and $\mathbf{b}^*$ are priors obtained through the marginalisation step.

The cost function is minimised using a Gauss-Newton iterative method with a three level coarse-to-fine pyramid scheme. We omit the Jacobians for readability and space constraints. After each iteration, the current and previous states are updated using the $\boxplus$ operator.

E. Partial Marginalisation & Fixation of Variables

The equations for the Gauss-Newton system are constructed from the Jacobians, error terms and information relating to the current and previous states, taking the form:

$$\begin{bmatrix} \mathbf{H}_{00} & \mathbf{H}_{01} \\ \mathbf{H}_{10} & \mathbf{H}_{11} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_0 \\ \delta\mathbf{x}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \end{bmatrix}. \tag{7}$$

After the current and previous states are updated, we marginalise out the previous state using the Schur-Complement:

$$\mathbf{H}_{11}^{*} = \mathbf{H}_{11} - \mathbf{H}_{10}\mathbf{H}_{00}^{-1}\mathbf{H}_{01}, \tag{8a}$$

$$\mathbf{b}_{1}^{*} = \mathbf{b}_{1} - \mathbf{H}_{10}\mathbf{H}_{00}^{-1}\mathbf{b}_{0}. \tag{8b}$$

The resulting $\mathbf{H}^{*}$ and $\mathbf{b}^{*}$ information is used as a prior in the next optimisation step. This partial marginalisation fixes the linearisation point, but with each iteration in the subsequent optimisation scheme, the linearisation point changes. Instead of relinearising at each step, we apply a first-order correction, $\Delta\mathbf{x}$, based on the difference between the new and old linearisation points as is commonly done in the literature [9] [24]:

$$\mathbf{H}_{11}^{*'} = \mathbf{H}_{11}^{*}, \tag{9a}$$

$$\mathbf{b}_{1}^{*'} = \mathbf{b}_{1}^{*} + \mathbf{H}_{11}^{*}\Delta\mathbf{x}. \tag{9b}$$

V. MAPPING

Like the original ElasticFusion, the map in our formulation is split into *active* and *inactive* areas [20]. The active map is the area most recently observed and is where the tracking and fusing takes place. If a segment of the active map is not observed for a period of time, δ_t, it becomes inactive. We keep the map globally consistent by attempting to match the currently observed portion of the active map with the inactive map. If a match is detected, the loop is closed by applying non-rigid space deformations through a sparse deformation graph. A deformation graph is a set of nodes, $\mathcal{G}^l$, that are embedded in the global model, each with a position, $\mathcal{G}^l_{\mathbf{g}}$, and a set of neighboring nodes, $\mathcal{G}^n \in \mathcal{N}(\mathcal{G}^l)$. Each deformation node stores a Euclidean transformation as a rotation, $\mathcal{G}^l_{\mathbf{R}}$, and a translation, $\mathcal{G}^l_{\mathbf{t}}$, that is used to elastically deform surfels in the map from a source position $\mathcal{Q}_{\mathbf{s}}$ to a destination position $\mathcal{Q}_{\mathbf{d}}$ through a deformation function, $\phi(\cdot)$, defined in [20]. This affine transformation is determined by minimising a cost function. In the original ElasticFusion, the cost function consists of five terms. The first encourages rigidity in the deformation:

$$E_{rot} = \sum_{l} \left\| {\mathcal{G}^l_{\mathbf{R}}}^{T} \mathcal{G}^l_{\mathbf{R}} - \mathbf{I} \right\|_F^2. \tag{10}$$

The second encourages smoothness in the deformation:

$$E_{\text{reg}} = \sum_{l} \sum_{n \in \mathcal{N}(\mathcal{G}^l)} \left\| \mathcal{G}^l_{\mathbf{R}}(\mathcal{G}^n_{\mathbf{g}} - \mathcal{G}^l_{\mathbf{g}}) + \mathcal{G}^l_{\mathbf{g}} + \mathcal{G}^l_{t} - (\mathcal{G}^n_{\mathbf{g}} + \mathcal{G}^n_{t}) \right\|_2^2. \tag{11}$$

The third minimises the distance of each point from the desired deformation:

$$E_{\text{con}} = \sum_{p} \left\| \phi(\mathcal{Q}^p_s) - \mathcal{Q}^p_d \right\|_2^2. \tag{12}$$

The fourth constrains the inactive areas of the map such that the active map is being deformed into the inactive map:

$$E_{\text{pin}} = \sum_{p} \left\| \phi(\mathcal{Q}^p_d) - \mathcal{Q}^p_d \right\|_2^2. \tag{13}$$

The fifth term is only applied to global deformations, and is used to prevent previous registrations, $\mathcal{R}$, from being pulled apart by future global loop closures:

$$E_{\text{rel}} = \sum_{p} \left\| \phi(\mathcal{R}^p_s) - \phi(\mathcal{R}^p_d) \right\|_2^2. \tag{14}$$

As matches between the active and inactive areas of the map are determined only by the RGB-D subsystem, we include a sixth cost term in our formulation to constrain the graph from deforming the map out of alignment with gravity:

$$E_{\text{imu}} = \sum_{l} \left\| \mathcal{G}^l_{\mathbf{R}} {}_{W}\mathbf{g} - {}_{W}\mathbf{g}_{\cdot} \right\|_2^2, \tag{15}$$

where ${}_{W}\mathbf{g}$ denotes the acceleration due to gravity represented in vision-world frame $\underrightarrow{\mathcal{F}}_W$.

Keeping the parameter choices the same as ElasticFusion, the total cost function for local loop closures is given by:

$$E_{\text{loc}} = \omega_f E_{\text{rot}} + \omega_r E_{\text{reg}} + \omega_c (E_{\text{con}} + E_{\text{pin}}) + \omega_i E_{\text{imu}}, \tag{16}$$

and the total cost for the global loop closures is given by:

$$E_{\text{glo}} = \omega_f E_{\text{rot}} + \omega_r E_{\text{reg}} + \omega_c (E_{\text{con}} + E_{\text{pin}} + E_{\text{rel}}) + \omega_i E_{\text{imu}}, \tag{17}$$

with ω_f = 1, ω_r = 10, and ω_c = ω_i = 100.

VI. RESULTS

We evaluate our system in terms of trajectory estimation and reconstruction accuracy on both synthetic and real world datasets. We adapt the living room sequences of the ICL-NUIM dataset [28] for the experiments on synthetic data. For the real world experiments, we recorded our own datasets along with ground truth poses from a Vicon motion capture system. The synthetic dataset consists of slow, smooth trajectories usually required for dense visual SLAM. The real world dataset contains a mixture of slow trajectories, aggressive motions and sequences with low texture and geometric information where vision-only systems tend to struggle.

We consider two metrics when examining the performance of the system: the absolute trajectory (ATE) root-mean-square error (RMSE) described in [29], and for reconstruction error, the mean distance from each point in the reconstruction to the nearest surface in the aligned ground truth model. The ATE RMSE is calculated for all sequences, but the reconstruction error is only available for the synthetic dataset. As the behavior of the loop closure mechanism is non-deterministic, we ran each test 10 times and took the average result. Tests where either system had lost tracking are denoted by brackets in the tables.

Through these experiments, we show that our dense RGB-D-inertial SLAM system performs at least as well as the RGB-D-only system on "easier" trajectories where the problem is well constrained by the visual data alone, but is much more robust when facing sequences with fast motions or little photometric and geometric variation.

TABLE I: System Parameters

Noise Parameter	Synthetic Dataset	Real World Dataset	Units
Gyr. saturation	7.8	7.8	rad s^{-1}
Acc. saturation	176.0	176.0	m s^{-2}
Gyr. noise density	12.0e-4	12.0e-4	rad s^{-1} $Hz^{-0.5}$
Acc. noise density	8.0e-3	8.0e-2	m s^{-2} $Hz^{-0.5}$
Gyr. bias Prior	0.03	0.03	rad s^{-1}
Acc. bias Prior	0.1	1	m s^{-2}
Gyr. drift noise density	4.0e-6	4.0e-6	rad s^{-2} $Hz^{-0.5}$
Acc. drift noise density	2.0e-5	2.0e-5	m s^{-3} $Hz^{-0.5}$
Acc. due to gravity	9.81	9.81	m s^{-2}
IMU rate	200	200	Hz
Static acc. bias	[0, 0, 0]	[0.060, 0.258, 0.126]	m s^{-2}
Image intensity noise	4.0	1.0	-
Image disparitiy noise	5.5	5.5	pixels

A. *Synthetic Data*

We evaluate both the trajectory estimation and surface reconstruction accuracy of our system on a modified version of the living room sequences in the ICL-NUIM dataset. The ICL-NUIM dataset is a benchmark that provides ground truth poses as well as a 3D model with which to evaluate reconstructions of RGB-D SLAM systems. The dataset does not come with inertial data, however, so in a manner similar to [30], we fit splines to the ground truth poses to simulate continuous trajectories. IMU measurements are then generated along these trajectories using the model described in [31] and the noise parameters given in Table I. However, due to the non-smooth trajectories of the dataset, we needed to sample every 10th frame of the ground truth trajectories when fitting the splines. This resulted in the new ground truth poses being close to but not exactly the same as those in the original dataset. Therefore, we rendered the images at these new poses using POV-Ray and applied the same noise models to the images as those in the original dataset.

Since the entire ground truth states are known for the synthetic dataset, we are able to demonstrate that our system can accurately track the velocity and IMU biases. For example, the error in the velocity and bias estimates for Sequence LR0, provided in Fig. 3, quickly converges to zero.

We compared the performance of the RGB-D-inertial and RGB-D-only versions of our system on each of the four living room sequences of the modified ICL-NUIM dataset. The results for the ATE RMSE are given in Table II and for the reconstruction error in Table III.

Although slow, some of the sequences in the modified ICL-NUIM dataset are still difficult for dense SLAM systems to follow, particularly the last sequence. In this sequence, the camera moves slowly along a wall providing little photometric or geometric variation. The results of the original ElasticFusion were not obtained using the same set of internal parameters for each sequence in the ICL-NUIM dataset. However, in this work, to showcase the robustness of our system and to avoid overfitting to a particular sequence, the default set of parameters was used across all datasets. As a result, the RGB-D-only version of ElasticFusion now fails on this sequence. The RGB-D-inertial system, however, is able to use the inertial data to get through the difficult section of the sequence and successfully reconstructs the scene. The RGB-D-inertial system performs approximately as well as the RGB-D-only system on the three easier sequences.

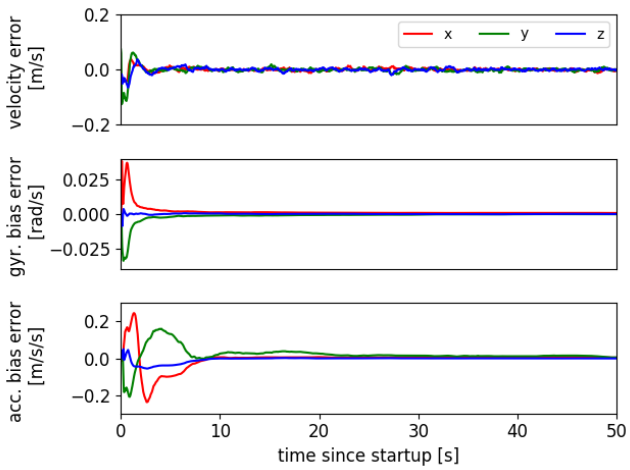


Fig. 3: Error in the velocity and bias estimates when compared to the ground truth values in synthetic dataset LR0. Our system is able to converge to and track the correct values.

TABLE II: ATE RMSE on the synthetic datasets (brackets indicate a tracking failure)

Sequence	RGB-D-Only	RGB-D-Inertial
LR0	0.032	**0.009**
LR1	**0.009**	0.012
LR2	**0.009**	**0.009**
LR3	(0.906)	**0.019**

TABLE III: Surface reconstruction accuracy on the synthetic datasets (brackets indicate a tracking failure)

Sequence	RGB-D-Only	RGB-D-Inertial
LR0	0.014	**0.008**
LR1	**0.007**	0.009
LR2	**0.010**	0.011
LR3	(0.118)	**0.010**

B. *Real World Data*

While the synthetic data showed that the RGB-D-inertial system is capable of performing at least as well as the RGB-D-only system on slow, smooth trajectories, the real strength of visual-inertial systems is their robustness to aggressive motions and sequences with little photometric or geometric information. To test this, a new dataset of 21 sequences

TABLE IV: Comparison of ATE RMSE on the real world datasets (brackets indicate a tracking failure)

Trajectory Type	Sequence	RGB-D-Only	RGB-D-Inertial
slow	1	0.227	**0.066**
	2	0.110	**0.065**
	3	0.225	**0.088**
slow, loop closure	4	0.089	**0.050**
	5	0.106	**0.048**
	6	0.091	**0.051**
medium	7	0.156	**0.077**
	8	0.166	**0.069**
	9	**0.118**	0.124
fast	10	0.098	**0.061**
	11	0.438	**0.354**
	12	0.267	**0.156**
quick rotation	13	0.231	**0.110**
	14	**0.057**	0.063
	15	0.220	**0.064**
low texture	16	(54.238)	**0.682**
	17	(26.306)	**0.498**
	18	(6.536)	(2.141)
long	19	**0.373**	0.560
	20	0.359	**0.216**
	21	0.417	**0.202**

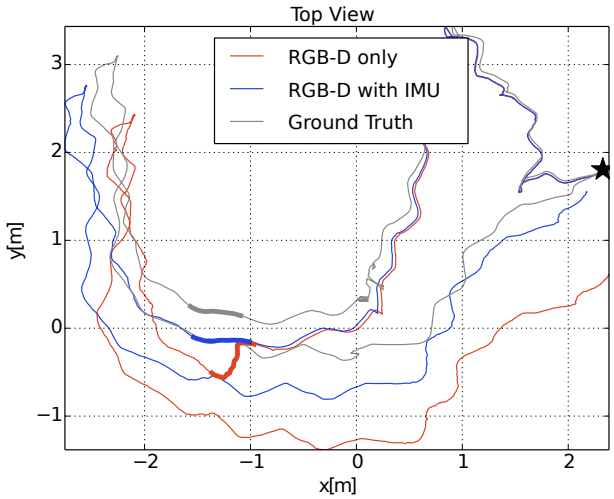


Fig. 4: Top view of the estimated trajectories for Sequence 21. Integrating the IMU data improves the tracking capabilities of the framework. In particular, it increases robustness against visually degenerate situations which pose a significant problem to the RGBD-only framework. Such an event, where the majority of the camera's field of view was filled with a white wall, is highlighted in the above trajectories.

was collected using the Intel RealSense ZR300 visual-inertial sensor. This sensor captures aligned RGB and depth images as well as inertial measurements. The camera intrinsics, as well as the transformation between the camera and IMU, $\boldsymbol{T}_{CS}$, was obtained using the Kalibr calibration system [32].

To see how our system would perform under different scenarios, a number of different types of datasets were captured. Sequences 1-3 are slow, smooth trajectories that typical RGB-D SLAM systems could handle. Sequences 4-6 are also slow and smooth, but with a large loop closure. Sequences 7-9 have slightly faster trajectories, and sequences 10-12 have very aggressive trajectories but continue to map the same area, allowing the SLAM system to keep tracking against a previously built up map. Sequences 13-15 are also aggressive trajectories, but include a quick rotation into an unmapped area of the scene. Sequences 16-18 are slow trajectories, but pass close to a white wall such that the RGB-D data provides little photometric or geometric information. Sequences 19-21 are slow, smooth trajectories, but much longer than the other sequences, on the order of 15-20m.

For each sequence, a ground truth trajectory was captured using the Vicon motion capture system. As explained in [29], these ground truth poses cannot be used to create a reliable ground truth scene reconstruction through depth image projection, as very small errors in the pose can result in very large errors in the reconstruction. For this reason, we do not calculate the reconstruction error for these sequences, only the ATE RMSE.

1) RGB-D-Only vs. RGB-D-Inertial: For each of the 21 sequences we compared the performance of the RGB-D-inertial system with the RGB-D-only system. The results of these experiments are presented in Table IV. In all but 3 of the sequences the RGB-D-inertial system outperformed the RGB-D-only system, often decisively. In particular, the RGB-D-only system was not capable of tracking the sequences where the camera moves across a white wall. The RGB-D-inertial system is able to rely on the IMU measurements to continue tracking despite the lack of photometric or geometric information, but in the final sequence of that group, the camera moves across the wall for too long and even the RGB-D-inertial system fails. Another example of this occurs in Sequence 21, where halfway through the trajectory the RGB-D-only system struggles when the camera goes across a blank wall. This is visualized in Fig. 4.

Qualitatively, we generally achieve a higher degree of map consistency in the RGB-D-inertial system compared with the RGB-D-only system. For example, Fig. 5 shows map reconstructions when the system is run on Sequence 7, a moderately difficult sequence in the real world dataset. The top level views show how the inclusion of inertial terms in tracking significantly reduces the amount of drift, as the map is much better aligned for the RGB-D-inertial system. Keeping the map aligned helps ElasticFusion find potential loop closures, but this will still sometimes fail as shown in the pair of images second from the bottom.

We encourage the reader to view our supplementary video for a better visualisation of our results.

2) Odometry vs. SLAM: To confirm that our formulation of a globally consistent map is improving our trajectory estimation, we examine the performance of our system with an open loop version where the system is restricted to only tracking and fusing against the active map (deformations are not allowed). We compared this on a sample sequence

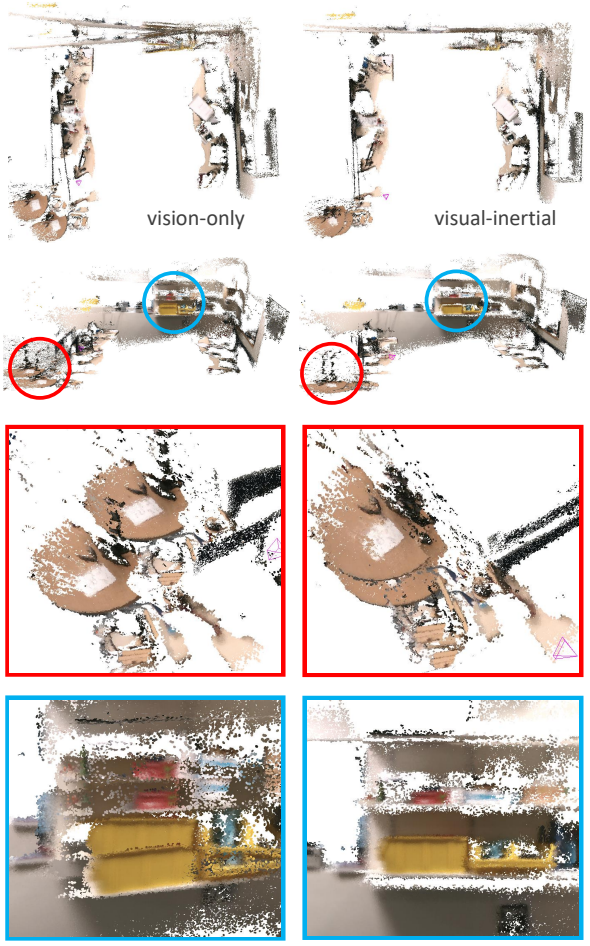


Fig. 5: Qualitative comparison of map reconstructions in RGB-D-only (left) and RGB-D-inertial (right) ElasticFusion: we generally achieve higher degree of map consistency through the inclusion of inertial measurements in the tracking. While loop closure was enabled, the first zoom-in (row second from the bottom) shows that ElasticFusion failed to detect and apply a larger loop closure (in both cases); but it also shows smaller drift as a starting point before potential loop closures. The second zoom-in (bottom row) highlights in more detail the generally higher map consistency with inertial integration.

from five of the different categories (we excluded the quick rotation and low texture scenes due to their difficulty). The results of these tests are shown in Table V. In all cases, the closed loop version performs at least as well as the open loop version.

3) Drift Analysis: In order to examine the relative contributions of the gyroscopes and accelerometers, we test the system on a number of sequences where the accelerometer related residuals are ignored. Fig. 6 shows the position error as a function of the distance traveled for Sequence 20, comparing RGB-D-only to RGB-D-and-gyroscopes-only to the full RGB-D-inertial system. As this figure shows, most of the gain in accuracy comes from the gyroscopes. This is confirmed by the results for the long sequences in Table VI. Over such a long sequence, the gyroscopes-only setup can outperform the full IMU due to the high noise levels of the accelerometers. The necessity of the accelerometers, however, is shown by the low texture sequences in Table VI. As the camera passes over the white wall, the gyroscope-only system fails because without visual input the relative position is no longer constrained.

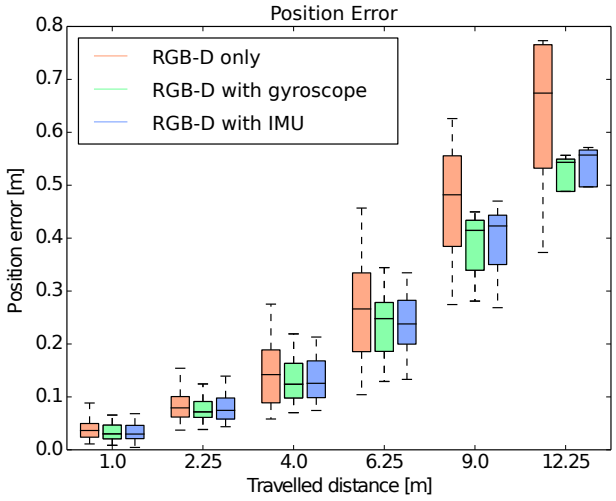


Fig. 6: Relation between accumulated position error and traveled distance for three different setups: no IMU, gyroscope only, and full IMU (gyroscope + accelerometer) for Sequence 20. We observe that most of the accuracy is gained from the integration of the gyroscopes. While the accelerometers do not significantly improve the accuracy of the system, their integration can contribute to the reliability of the system.

TABLE V: Comparison of ATE RMSE between open loop odometry and SLAM on the real world datasets

Trajectory Type	Sequence	Odometry	SLAM
slow	1	0.102	**0.066**
slow with loop closure	4	0.051	**0.050**
medium	7	0.078	**0.077**
fast	10	0.062	**0.061**
long	19	0.541	**0.525**

TABLE VI: Comparison of ATE RMSE between RGB-D-inertial and RGB-D with only gyroscopes on the real world datasets

Trajectory Type	Sequence	Gyro Only	Full IMU
long	19	**0.296**	0.560
	20	0.223	**0.216**
	21	0.203	**0.202**
low texture	16	(7.548)	**0.682**
	17	(3.916)	**0.498**
	18	(0.917)	(2.141)

VII. CONCLUSION

We have presented what is, to the best of our knowledge, the first real-time tightly-coupled dense RGB-D-inertial SLAM system. In the tracking step, it minimises a combined photometric, geometric and inertial energy functional to simultaneously estimate the camera pose, velocity, IMU biases and gravity direction. In the mapping step, our system constructs a fully dense 3D reconstruction of the environment which is not only globally consistent, but gravity aligned due to the addition of an inertial deformation energy applied to the deformation graph.

We show through a series of experiments on both synthetic and real world datasets that our RGB-D-inertial system performs at least as well as the RGB-D-only version of our system on slow, smooth trajectories, but is much more robust to aggressive motions and a lack of photometric and geometric variation.

REFERENCES

[1] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: a Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics (T-RO)*, vol. 31, no. 5, pp. 1147–1163, 2015.

[2] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast Semi-Direct Monocular Visual Odometry," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[3] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 3565–3572.

[4] M. Li and A. Mourikis, "High-precision consistent EKF-based visual-inertial odometry," *International Journal of Robotics Research (IJRR)*, vol. 32, pp. 690–711, 2013.

[5] P. Tanskanen, T. Naegeli, M. Pollefeys, and O. Hilliges, "Semi-direct EKF-based monocular visual-inertial odometry," in *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2015.

[6] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct ekf-based approach," in *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2015.

[7] E. S. Jones and S. Soatto, "Visiual-inertial navigation, mapping and localization: A scalable real-time causal approach," *International Journal of Robotics Research (IJRR)*, vol. 30, no. 4, pp. 407–430, 2011.

[8] N. Keivan, A. Patron-Perez, and G. Sibley, "Asynchronous adaptive conditioning for visual-inertial SLAM," in *Proceedings of the International Symposium on Experimental Robotics (ISER)*, 2014.

[9] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, p. 0278364914554813, 2014.

[10] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," in *Proceedings of Robotics: Science and Systems (RSS)*, 2015.

[11] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 957–964.

[12] J. Engel, J. Sturm, and D. Cremers, "Camera-based navigation of a low-cost quadrocopter," in *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2012.

[13] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "Pixhawk: A system for autonomous flight using onboard computer vision," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[14] R. A. Newcombe, S. Lovegrove, and A. J. Davison, "DTAM: Dense Tracking and Mapping in Real-Time," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[15] J. Engel, T. Schoeps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[16] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-Time Dense Surface Mapping and Tracking," in *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.

[17] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGB-D cameras," in *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2013.

[18] T. Whelan, J. B. McDonald, M. Kaess, M. Fallon, H. Johannsson, and J. J. Leonard, "Kintinuous: Spatially Extended KinectFusion," in *Workshop on RGB-D: Advanced Reasoning with Depth Cameras, in conjunction with Robotics: Science and Systems*, 2012.

[19] O. Kahler, V. A. Prisacariu, C. Y. Ren, X. Sun, P. H. S. Torr, and D. W. Murray, "Very High Frame Rate Volumetric Integration of Depth Images on Mobile Device," in *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2015.

[20] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "ElasticFusion: Real-time dense SLAM and light source estimation," *International Journal of Robotics Research (IJRR)*, vol. 35, no. 14, pp. 1697–1716, 2016.

[21] S. Omari, M. Bloesch, P. Gohl, and R. Siegwart, "Dense visual-inertial navigation system for mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[22] L. Ma, J. M. Falquez, S. McGuire, and G. Sibley, "Large scale dense visual inertial SLAM," in *Proceedings of the International Symposium on Experimental Robotics (ISER)*, 2015.

[23] A. Concha, G. Loianna, V. Kumar, and J. Civera, "Visual-inertial direct SLAM," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[24] V. Usenko, J. Engel, J. Stückler, and D. Cremers, "Direct visual-inertial odometry with stereo cameras," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[25] J. Engel, J. Stückler, and D. Cremers, "Large-scale direct slam with stereo cameras," in *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2015.

[26] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder, "Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds," *Information Fusion*, vol. 14, no. 1, pp. 57–77, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1566253511000571

[27] M. Bloesch, H. Sommer, T. Laidlow, M. Burri, G. Nützi, P. Fankhauser, D. Bellicoso, C. Gehring, S. Leutenegger, M. Hutter, and R. Siegwart, "A Primer on the Differential Calculus of 3D Orientations," *CoRR*, vol. abs/1606.0, 2016. [Online]. Available: http://arxiv.org/abs/1606.05285

[28] A. Handa, T. Whelan, J. B. McDonald, and A. J. Davison, "A Benchmark for RGB-D Visual Odometry, 3D Reconstruction and SLAM," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2014. [Online]. Available: http://www.doc.ic.ac.uk/ ahanda/VaFRIC/iclnuim.html

[29] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A Benchmark for the Evaluation of RGB-D SLAM Systems," in *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2012.

[30] C. Kerl, J. Stückler, and D. Cremers, "Dense continuous-time tracking and mapping with rolling shutter RGB-D cameras," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.

[31] J. Nikolic, P. Furgale, A. Melzer, and R. Siegwart, "Maximum likelihood identification of inertial sensor noise model parameters," *IEEE Sensors Journal*, vol. 16, no. 1, pp. 163–176, 2016.

[32] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 1280–1286.