2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 24–28, 2017, Vancouver, BC, Canada

Model-Aided Monocular Visual-Inertial State Estimation and Dense Mapping

Kejie Qiu and Shaojie Shen

Abstract— **Robust state estimation and real-time dense mapping are two core capabilities for autonomous navigation of mobile robots. Global Navigation Satellite System (GNSS) and visual odometry/SLAM are popular methods for state estimation. However, when working between tall buildings or in indoor environments, GNSS fails due to limited sky view or obstruction from buildings. Visual odometry/SLAM are prone to long-term drifting in the absence of reliable loop closure detection. A state estimation method with global-consistent guarantee is desirable for navigation applications. As for real-time mapping, SLAM methods usually get a sparse map that is not good enough for obstacle avoidance and path-planning, and high-quality dense mapping is often computationally too demanding for mobile devices. Realizing the availability of city-scale 3D models, in this work, we improve our previous work on model-based global localization, and propose a model-aided monocular visual-inertial state estimation and dense mapping solution. We first develop a global-consistent state estimator by fusing visual-inertial odometry with the model-based localization results. Utilizing depth prior from the model, we perform motion stereo with semi-global disparity smoothing. Our dense mapping pipeline is capable of online detection of obstacles that are originally not included in the offline 3D model. Our method runs onboard an embedded computer in real-time. We validate both the state estimation and mapping accuracy in real-world experiments.**

I. Introduction

We are interested in a universal solution to global-consistent state estimation and real-time dense mapping, which can be easily implemented on a light-weight mobile agent, e.g., a Micro Aerial Vehicle (MAV), in multiple kinds of scenarios, ranging from indoor environment to large-scale urban environments. For example, aerial robots with autonomous navigation capabilities are particularly suitable for delivery and infrastructure inspection applications. One prerequisite of such an autonomous system is reliable global localization, GNSS-based solutions have been widely used but they can easily be disrupted or disabled due to obstructed sky view, which often occurs in urban and indoor environments. And global localization solutions based on place recognition [1], [2] can only obtain topological localization which is not accurate enough for closed-loop control. Another solution is using incremental localization methods including visual odometry and SLAM-based approaches using monocular camera [3], [4], [5]. Among these methods, multi-state constraint Kalman filter (MSCKF) [6] is a light-weight filter-based solution and our previous work on visual-inertial system (VINS) [7] adopts a sliding window optimization with online initialization and camera-IMU extrinsic calibration. But odometry-based methods suffer from long-term drifting while SLAM-based approaches can not guarantee global consistency before a major loop closures detection. Fusion of odometry, SLAM and GNSS may resolve the localization problem in most cases, but it still does not guarantee drift-free localization at all times.

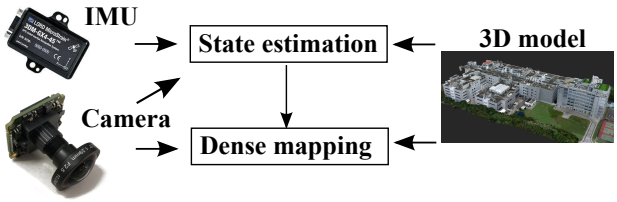


Fig. 1: The minimal sensing for the proposed model-aided visual-inertial state estimation and mapping system.

On the other side, real-time dense mapping is desired for obstacle avoidance and path-planning purposes of MAV, and occlusion detection consideration in augmented reality (AR) community. Sensors like depth sensor and stereo cameras are appropriate for certain applications. KinectFusion utilizes an RGBD sensor to solve the simultaneously localization and mapping problem, and with the aid of truncated signed distance field (TSDF) fusion, the constructed map is suitable for both obstacle avoidance and visualization [8]. However, the intrinsic detection limitation impedes outdoor applications. Similar methods using stereo camera are also studied. [9] proposes to use semi-global matching (SGM) for depth map optimization. Both data term and regularized term are involved in the optimization object using a simplified smoothness scheme. Besides that limited baseline constrains detection range, extrinsic calibration is another issue for easy use. All these shortcomings drive us to work on monocular dense mapping using only a monocular camera as the extrinsic sensor.

Actually, spatial stereo can be extended to temporal stereo or motion stereo with precise relative pose estimation provided. Accordingly, the drawbacks of traditional stereo like calibration and baseline limitation are all gone. Our previous work [10] has realized motion stereo for real-time mapping using SGM, which is parallelly implemented on an embedded GPU and used for online MAV path-planning. However, the major contradiction of this solution is painfully obvious, that high-quality dense mapping is always time demanding and computational resources consuming. Semi-dense mapping based on LSD-SLAM [3] is proposed by

This work was supported by the WeChat-HKUST Joint Laboratory on Artificial Intelligence Technology (WHAT LAB).

All authors are with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, China. kqiuaa@connect.ust.hk, eeshaojie@ust.hk

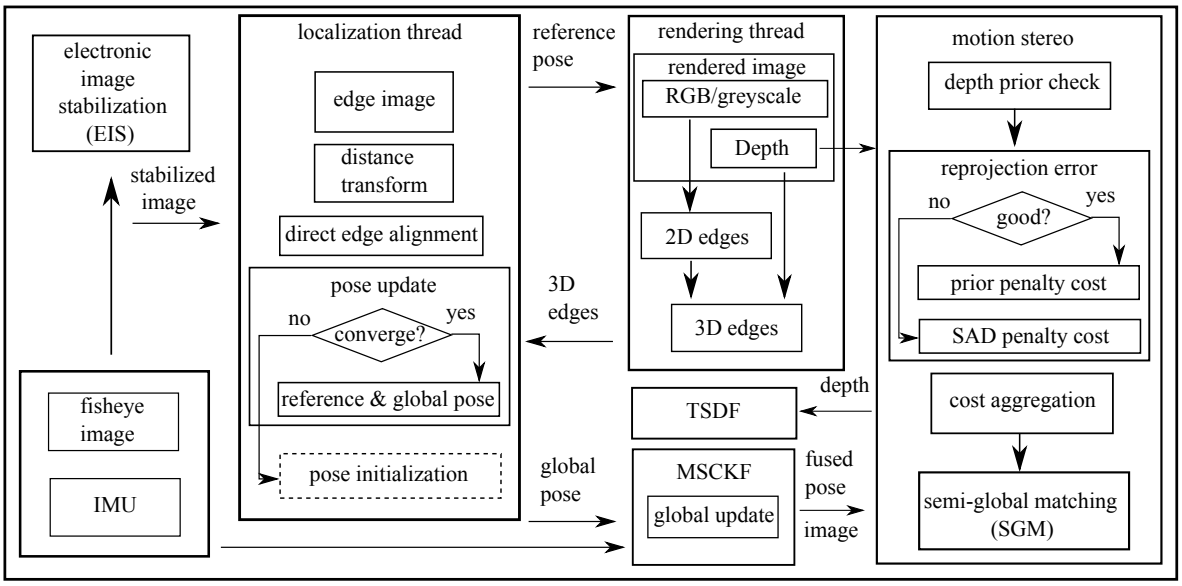


Fig. 2: The overall structure of the proposed model-based global localization and mapping system.

Engel et al., which is different from both feature-based sparse mapping and dense mapping. Only high-gradient pixels are chosen for real-time mapping on CPU which may not be good enough for safe navigation. DTAM efficiently solve the depth estimation problem with total variation by using a primal-dual method [11]. REMODE [12] is another monocular solution using a similar method. Whereas, both of them are computationally demanding such that a desktop-level GPU is needed for real-time performance.

Fortunately, today we can easily obtain 3D models of major urban areas through multiple online resources or online model construction services, such as Altizure, Skycatch[1], etc. These dense 3D models are constructed using well-studied offline structure from motion (SfM) techniques, in which all the camera poses and pixels are optimized jointly to achieve globally dense consistency. Utilizing these 3D models to achieve global localization becomes a possible solution, and global localization is essentially an image registration problem which aims to align the image capture by the onboard camera and a virtual image rendered from the 3D model. Also, today's high-performance mobile devices for graphics computation make real-time rendering a reality. The rendering process provides both a virtual RGB image and a depth image with nearly unlimited range, from which localization can be done by using only the sensory input from one camera. Related work on 3D edge-based tracking with multiple hypotheses are proposed years ago [13], [14], but only Computer-Aided Design (CAD) model or simple geometric model is used for edge extraction. The most relevant work of utilizing a 3D model is a direct alignment-based tracking system on a known mesh model constructed by Kinect fusion for indoor environments [15].

In our previous work, a light-weight global localization system for aerial robots using only a monocular fisheye camera and an IMU based on a rough 3D model is proposed [16]. A robust edge alignment-based method [17] under strong changes in lighting conditions and camera characteristics is adopted for image-model registration. We further improve the localization performance with electronic image stabilization (EIS) for robust tracking, and Extended Kalman Filter (EKF)-based IMU fusion for closed-loop control realization. In that way, all-the-time global-consistent localization that is accurate enough for closed-loop flight control using minimal sensing without detection distance limitation and onboard computer is fulfilled, as shown in Fig. 1. This global localizer can work independently or work with any other state estimation methods to correct any possible drift at all times. In addition, the rendered virtual view is a strong prior for dense mapping, which means high-quality real-time monocular dense mapping using limited onboard computational resources can also be solved with the combination of the view prior and original motion stereo. In this paper, we integrate the model-based global localization with a well studied visual-inertial fusion method to get all-the-time global localization with good local accuracy, which improves the robustness and localization accuracy of state estimation for better motion stereo implementation. The rendered depth maps from the 3D model serve as prior knowledge for final depth estimation because the realistic scene may be different from the model. Given estimated depth maps of different camera poses, the last problem is how to fuse them together to obtain optimized dense 3D reconstruction. Instead of using simple occupancy grid mapping, we follow the open-source CHISEL [18], i.e., dynamic spatially-hashed TSDF-based map fusion approach for 3D map reconstruction.

The closed-loop flight control and online dense mapping and trajectory planning have been shown in our previous

[1]www.altizure.com, www.skycatch.com

work [16], [10], thus we mainly focus on the localization accuracy and mapping quality in this paper, and identify our contributions as follows:

- We handle simultaneously global localization and real-time dense mapping problem with minimum sensing and a rough prior 3D model.
- We integrate the model-based global localization method with a tightly-coupled visual-inertial fusion method to get all-the-time global localization with high local accuracy.
- We implement motion stereo with depth prior rendered from a prior 3D model to realize accurate environment awareness.

II. Overview

The overall structure of the proposed approach is shown in Fig. 2. The rendering thread generates a virtual image and its depth map accordingly. For every fisheye image captured by the camera, it first goes through the EIS module to obtain a cropped and stabilized pinhole image. Edges are extracted from the stabilized image to eliminate impacts from different illumination conditions, shadows, etc. This edge image is further turned into a distance transform for constructing the cost function for edge alignment. At the same time, edges of the rendered RGB image are also extracted and further converted into 3D edges since the depth values of the virtual image are known. The 3D edges can be reprojected onto the distance transform field, and relative translation and rotation between the actual camera view and the rendered view (from initial guess) can be derived by minimizing the reprojection error. The global pose is computed following standard pose compounding of relative transformations. The updated global pose is fed back to the rendering thread for the next rendering and registration. The rendered depth map is also used as motion stereo prior, which is first checked by the reprojection error. For the pixels which are expected to be located on the prior depth value, a Huber-norm cost is applied as the prior penalty, while for the other pixels, an original sum of absolute difference penalty is calculated for different depths. A semi-global matching optimization is applied on the aggregated cost with considering the smoothness penalty, and the generated depth maps are organized in a TSDF framework for mapping.

The rest of the paper is structured as follows. Section III introduces the global pose fusion with an arbitrary visual-inertial odometry as a global measurement update. In section IV, the fused state estimates are used for monocular dense mapping, including depth prior-based motion stereo and 3D reconstruction by using a TSDF framework. Section V gives the experimental results with comparison against ground truth and the original model. Conclusion and directions for future work are presented in Section VI.

III. Global pose update for visual-inertial odometry

The original visual-inertial odometry (VIO) can already handle local area autonomous navigation for both indoor and outdoor environment. With the global reference generated by a 3D model, we can even equip an arbitrary visual-odometry method with all-the-time global-consistent property, which is significantly different from that of loop closure where global localization cannot be obtained until a good loop closure is detected. Thus our global localization solution is particularly appropriate for being deployed in large-scale outdoor urban environments. We adopt MSCKF as the VIO implementation that is based on Kalman filter and treat the global pose update as an additional EKF update. In order to follow the error state representation way used in MSCKF, we modify the original model-based global observation accordingly. The EKF error-state vector including N camera frames is defined as:

$$\tilde{\mathbf{X}}_k = [\tilde{\mathbf{X}}_{IMU_k}^T \ \delta\theta_{C_1}^T \ \tilde{\mathbf{p}}_{C_1}^T \dots \delta\theta_{C_N}^T \ \tilde{\mathbf{p}}_{C_N}^T], \tag{1}$$

where the IMU error-state vector consists of orientation, position, velocity and biases is defined as:

$$\tilde{\mathbf{X}}_{IMU} = [\delta\theta_I^T \ \tilde{\mathbf{b}}_g^T \ \tilde{\mathbf{v}}_I{}^T \tilde{\mathbf{b}}_a^T \ \tilde{\mathbf{p}}_I^T]. \tag{2}$$

Followed by the regular IMU state propagation and camera state augmentation, the EKF update using relative visual measurements is derived as:

$$\Delta\mathbf{X} = \mathbf{K}_r\mathbf{r}_n, \tag{3}$$

and the state covariance matrix is updated according to:

$$\mathbf{P}_{k+1|k+1} = (\mathbf{I}-\mathbf{K}_r\mathbf{T_H})\mathbf{P_{k+1|k}}(\mathbf{I}-\mathbf{K}_r\mathbf{T_H})^T+\mathbf{K}_r\mathbf{R}_n\mathbf{K}_r^T, \tag{4}$$

where the Kalman gain of relative measurements is:

$$\mathbf{K}_r = \mathbf{P}\mathbf{T}_H^T(\mathbf{T}_H\mathbf{P}\mathbf{T}_\mathbf{H}^\mathbf{T}+\mathbf{R_n})^{-1}. \tag{5}$$

Here $\mathbf{r}_n$ and $\mathbf{R}_n$ are modified residual and covariance, more details can be found in this paper [6].

In order to apply global state update using the absolute measurements from the model-based localizer to MSCKF, we align the MSCKF state including the IMU state and camera states into the model frame when the first global observation comes. Then every time a new global observation comes ($\mathbf{p_z}$ $\mathbf{q_z}$), which is attached to a certain camera frame (camera index $C_* \in [C_1, C_N]$), a correction to the corresponding camera state in MSCKF ($\hat{\mathbf{p}}$ $\hat{\mathbf{q}}$) is calculated according to an EKF update step and applied to the whole MSCKF state, such that the drift caused by MSCKF is corrected meanwhile the relative state transformations between camera frames and IMU frame in MSCKF state are preserved.

We define a new model observation as:

$$\mathbf{z}' = \begin{bmatrix}\mathbf{p}_{\mathbf{z}'}\\ \mathbf{q}_{\mathbf{z}'}\end{bmatrix} = \begin{bmatrix}\mathbf{p_z}-\hat{\mathbf{p}}\\ \hat{\mathbf{q}}^{-1}\otimes\mathbf{q_z}\end{bmatrix}, \tag{6}$$

where

$$\mathbf{p_z} = \mathbf{p}+\mathbf{n_p}, \mathbf{p} = \hat{\mathbf{p}}+\delta\mathbf{p}, \mathbf{q_z} = \mathbf{q}\otimes\mathbf{n_q}, \mathbf{q} = \hat{\mathbf{q}}\otimes\delta\mathbf{q}.$$

We convert the quaternions caused by small rotations to minimal representation:

$$\mathbf{n_q} \approx \begin{bmatrix}\frac{1}{2}n_\theta\\ 1\end{bmatrix}, \quad \delta\mathbf{q} \approx \begin{bmatrix}\frac{1}{2}\delta\theta\\ 1\end{bmatrix}, \quad \mathbf{q}_{\mathbf{z}'} \approx \begin{bmatrix}\frac{1}{2}\theta_{z'}\\ 1\end{bmatrix}.$$

The position and orientation correction to the corresponding camera frame state is derived from the EKF update step:

$$\Delta_p = \mathbf{K}_{a_1} p'_z,\ \Delta_\theta = \mathbf{K}_{a_2} \theta'_z, \tag{7}$$

and the covariance matrix block is updated according to:

$$\mathbf{P}_{p_{C*}} = \mathbf{P}_{p_{C*}} - \mathbf{K}_{a_1}\mathbf{P}_{p_{C*}},\ \mathbf{P}_{\theta_{C*}} = \mathbf{P}_{\theta_{C*}} - \mathbf{K}_{a_2}\mathbf{P}_{\theta_{C*}}, \tag{8}$$

where

$$\mathbf{K}_{a_1} = \mathbf{P}_{p_{C*}}(\mathbf{P}_{p_{C*}} + \mathbf{R}_1)^{-1}, \mathbf{K}_{a_2} = \mathbf{P}_{\theta_{C*}}(\mathbf{P}_{\theta_{C*}} + \mathbf{R}_2)^{-1}, \tag{9}$$

and $\mathbf{R}_1$, $\mathbf{R}_2$ are the position and orientation estimation covariances of the model-based localizer.

IV. Monocular dense mapping with depth prior constrains

A. Multi-view cost aggregation

Different from spatial stereo where only two calibrated views are used for depth estimation, multiple temporal camera views are used for depth estimation with precise pose estimation for every camera frame. One key advantage is that there is no baseline limitation any longer as long as enough camera motion is accumulated during flight, and the same depth estimation scheme can be used both small indoor environments and large outdoor cases. We uniformly sample l planes with inverse depth λ_k where $k \in [0, l-1]$. The depth of k^{th} enumerated plane is $d_k = \frac{1}{k \cdot \lambda_{min}}$, where $d_{min} = \frac{1}{(l-1)\cdot\lambda_{min}}$ is the nearest plane and $d_{max} = \frac{1}{0\cdot\lambda_{min}} = \infty$ is the plane at infinity. The back-projection procedure from a 2D pixel $\mathbf{u} = [u\ v]^T$ to the corresponding 3D point $\mathcal{P} = [x\ y\ z]^T$ in world frame can be expressed as:

$$\mathcal{P}^w = \mathbf{R}_r^w(d \cdot K^{-1} \cdot \dot{\mathbf{u}}) + \mathbf{p}_r^w, \dot{\mathbf{u}} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \tag{10}$$

where $\dot{\mathbf{u}} = [u, v, 1]^T$ is the pixel $\mathbf{u}$ in homogeneous coordinates. The corresponding 2D pixel $\mathbf{u}'$ in measurement frame I_m is computed by perspective projection:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}^m = K\mathbf{R}_m^{w^T}(\mathcal{P}^w - \mathbf{p}_m^w), \mathbf{u}' = \begin{bmatrix} \frac{x}{z} \\ \frac{y}{z} \end{bmatrix}. \tag{11}$$

For better performance, a 3×3 patch is applied for sum of absolute difference (SAD) computation to evaluate similarity. The cost will be:

$$E_{SAD}(\mathbf{u}, k) = \sum_{m \in \mathcal{M}} \sum_{\mathbf{u}_r \in \mathcal{N}(\mathbf{u})} |I_r(\mathbf{u}_r) - I_m(\mathbf{u}')|. \tag{12}$$

To fully make use of the model prior, we reuse the depth map rendered from the 3D model as a strong prior knowledge for online depth estimation. In fact, we first check the prior depth values to create an estimation mask for further similarity evaluation.

$$M_D(\mathbf{u}) = \begin{cases} 1 & \text{if } E_{SAD}(\mathbf{u}, d_{prior}) \le E_{thresh} \\ 0 & \text{otherwise} \end{cases}, \tag{13}$$

An image eroding process is also applied to the mask image for outlier rejection. For the pixels with mask value 1, we treat the corresponding cost values as a Huber-norm penalty centered at the prior depth value.

$$E_{prior}(\mathbf{u}, k) = \begin{cases} \frac{1}{2}a^2 & \text{if } |a| \le \Delta \\ \Delta(|a| - \frac{1}{2}\Delta) & \text{otherwise} \end{cases}, \tag{14}$$

where $a = \frac{1}{k \cdot \lambda_{min}} - d_{prior}$.

While the other pixels are processed with the regular similarity evaluation step shown in Equation 12, the similarity cost or the data term can then be summarized as:

$$E_{data}(\mathbf{u}, k) = \begin{cases} E_{prior} & M_D(\mathbf{u}) = 1 \\ E_{SAD} & M_D(\mathbf{u}) = 0 \end{cases} \tag{15}$$

For parallel realization on the GPU, Eq.(10) and Eq.(11) can be combined as:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}^m = K\mathbf{R}_m^{w^T}(\mathbf{R}_r^w(d \cdot K^{-1} \cdot \dot{\mathbf{u}}) + \mathbf{p}_r^w - \mathbf{p}_m^w) \tag{16}$$

$$= \frac{1}{k\lambda_{min}} K\mathbf{R}_m^{w^T}\mathbf{R}_r^w K^{-1}\dot{\mathbf{u}} + K\mathbf{R}_m^{w^T}(\mathbf{p}_r^w - \mathbf{p}_m^w) \tag{17}$$

$$= \frac{1}{k\lambda_{min}}\mathbf{H}\dot{\mathbf{u}} + \mathbf{J} \tag{18}$$

$$\mathbf{u}' = \begin{bmatrix} \left(\frac{\mathbf{h}_1\dot{\mathbf{u}}}{k\lambda_{min}} + J_1\right) / \left(\frac{\mathbf{h}_3\dot{\mathbf{u}}}{k\lambda_{min}} + J_3\right) \\ \left(\frac{\mathbf{h}_2\dot{\mathbf{u}}}{k\lambda_{min}} + J_2\right) / \left(\frac{\mathbf{h}_3\dot{\mathbf{u}}}{k\lambda_{min}} + J_3\right) \end{bmatrix}. \tag{19}$$

Every re-projection pixel is found by back-projecting a pixel in the reference frame to a 3D point and re-projecting this 3D point into the measurement frame. For each pixel $\mathbf{u}$ in the reference frame, we adopt this procedure for its neighbor pixel $\mathbf{u}_r$ repeatedly. The cost volume E_{SAD} is computed and aggregated temporally from multiple image measurements at different instants of time.

B. Semi-global matching

Denote D as the depth map we try to optimize. We define energy $E(D)$ which combines the pixel-wise cost and smoothness constraints as:

$$\begin{aligned} E(D) = \sum_{\mathbf{u}} (&E_{data}(\mathbf{u}, k) \\ &+ P_1 \cdot \sum_{\mathbf{u}_q \in \mathcal{N}(\mathbf{u})} T[|k - k_{\mathbf{u}_q}| = 1] \\ &+ P_2 \cdot \sum_{\mathbf{u}_q \in \mathcal{N}(\mathbf{u})} T[|k - k_{\mathbf{u}_q}| > 1]), \end{aligned} \tag{20}$$

where $\mathcal{N}(\mathbf{u})$ are the neighbor pixels around $\mathbf{u}$, and the energy function for each pixel consists of three terms: one data term directly from the cost volume and two regularization terms. The first regularization term penalizes the energy by neighbor pixels with which the enumerated depth difference is 1, and the second penalty caused by neighbor pixels with which enumerated depth difference is large than 1.

Although the two kinds of data penalty costs may have different scale values, they may only influence the cost along

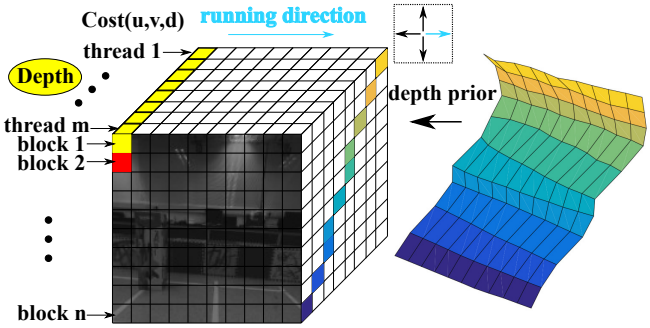


Fig. 3: Semi-global approximation process on GPU. n blocks denote the n rows of the image and m threads denote the corresponding sample depths, which runs parallelly by using CUDA. One direction from left to right is shown here.

different depth values rather than the smoothness effect, because as Equation 20 shows, the optimization results is only affected by the neighbor depth values instead of the specific penalty cost of that depth value.

Since the global minimization is an NP-complete problem which can not be solved in polynomial time, a semi-global approximation is proposed in [9]:

$$S(\mathbf{u},k)=\sum_{\mathbf{u}}\sum_{\mathbf{r}}L_{\mathbf{r}}(\mathbf{u},D_{\mathbf{u}}), \tag{21}$$

$$\begin{aligned}L_{\mathbf{r}}(\mathbf{u},k)=&E_{SAD}(\mathbf{u},k)+\min\begin{Bmatrix}L_{\mathbf{r}}(\mathbf{u}-\mathbf{r},k)\\ L_{\mathbf{r}}(\mathbf{u}-\mathbf{r},k-1)+P_1\\ L_{\mathbf{r}}(\mathbf{u}-\mathbf{r},k+1)+P_1\\ \min_i\{L_{\mathbf{r}}(\mathbf{u}-\mathbf{r},i)+P_2\}\end{Bmatrix}\\ &-\min_j L_{\mathbf{r}}(\mathbf{u}-\mathbf{r},j),\end{aligned} \tag{22}$$

where the global minimization problem is simplified as a combination of several 1D problems along different directions. The approximation cost function is good enough for fast depth estimation and it can be parallelized by on GPU. An intuitive illustration of a depth prior-based 4-direction SGM is shown in Fig. 3, where only the direction from left to right is presented in the figure.

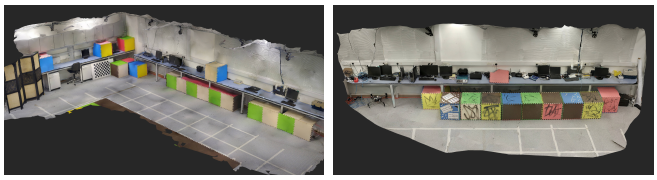
C. 3D reconstruction

For dense 3D mapping using the discrete depth maps, we apply TSDF to get an uncertainty-aware representation. The 3D environment is modeled as a volumetric signed distance field $\phi(x):\mathbb{R}^3\rightarrow\mathbb{R}$. For any point x in the world, $\Phi(x)$ is the distance to the nearest surface, signed positive if the point is outside of obstacles and negative otherwise. The scene surface is encoded by the unique zero isocontour and the surface ambiguity is eliminated.

Since we are only interested in constructing the surface, we use truncated signed distance field:

$$\Phi_\tau(x)=\begin{cases}\phi(x) & \text{if } |\phi(x)|\leq\tau\\ \text{undefined} & \text{otherwise}\end{cases}, \tag{23}$$

where $\tau\in\mathbb{R}$ is the truncation distance.



(a) For localization comparison (b) For mapping comparison

Fig. 4: The 3D models we use for virtual view and depth prior rendering. They are constructed offline using the service from Altizure.com.

Besides truncated signed distance field, we store two more values at each 3D voxel. $C(x):\mathbb{R}^3\rightarrow\mathbb{R}$ is the photometric intensity which is maintained like $\Phi_\tau(x)$. $W(x):\mathbb{R}^3\rightarrow\mathbb{R}$ is the confidence weight of the measurements.

V. Experimental Results

A. Implementation details

The proposed method is mainly designed for autonomous navigation in large-scale outdoor environments where no ground truth data is provided (GPS is not accurate enough and it's also error-prone). As what our previous work does, we first construct a 3D model of an indoor environment where a motion capture system [2] is deployed to provide ground truth references for numerical analysis, as shown in Fig. 4 (a). We create another environment with textured objects inside it for mapping comparison, as shown in Fig. 4 (b). We use the service by Altizure.com for 3D model construction of the experimental scenes. The onboard IMU runs at 100 Hz, while the camera captures 752 by 480 images at 12 Hz. The size of the stabilized image after EIS is 160 by 108, which is also used for 4-direction SGM depth estimation. All the data is collected with an MAV mounted with a fisheye camera [16]. Onboard computation is a NVIDIA Tegra X1, including a 4-core CPU and a 256-CUDA-core GPU. Our implementation is able to process the stabilized images at 12 Hz, including simultaneously global localization and mapping.

B. State estimation results

Limited by space and safety consideration, we move the drone manually in the indoor environment with different motion patterns. The total trajectory length is about 80 meters. The global-consistent of the model-based method has been validated in our previous work, here we focus on the localization accuracy by fusion with an incremental localization method. In this way, all-the-time high local accuracy and global consistency can both be satisfied, which can be used for precise control feedback and motion stereo. Independently, MSCKF itself has marginal performance before fusion with the global reference state, while the fused results have much better performance.

The position, orientation comparison and error terms are shown in Fig. 5. Compared with MSCKF, the fused

[2]http://www.optitrack.com/

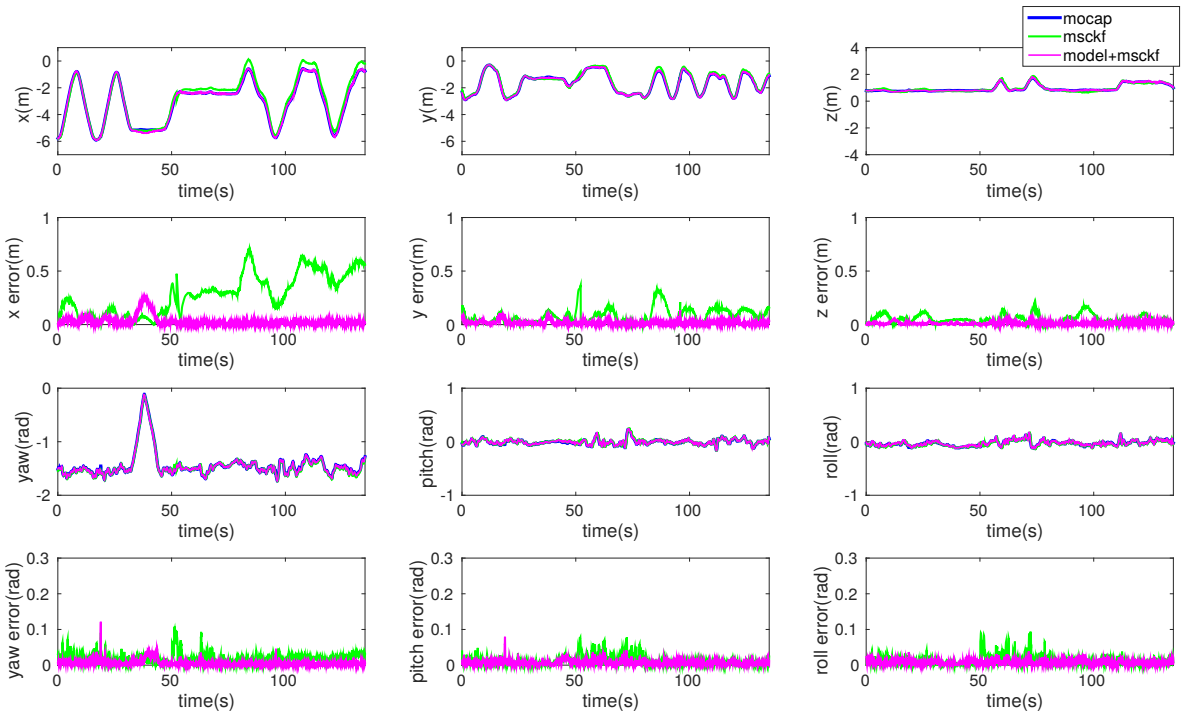


Fig. 5: The comparison of position, orientation of MSCKF and MSCKF+Model. The corresponding error is calculated from the absolute difference between different localization methods and the ground truth.

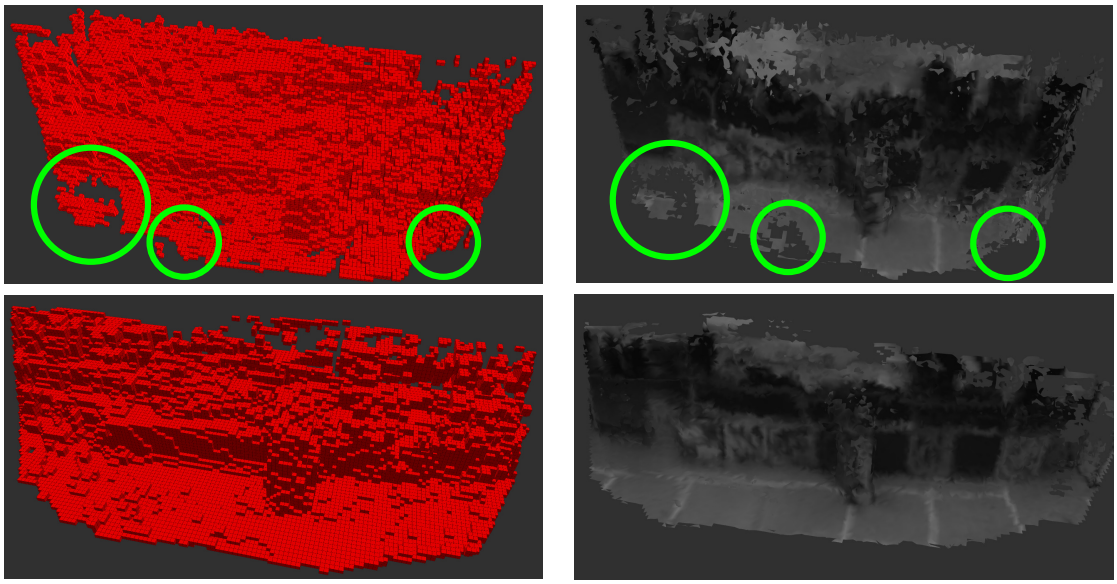


Fig. 7: Online mapping comparison, the first row shows the grid and mesh map by using original motion stereo and the second row shows the results by using depth prior-based motion stereo. Obviously, the depth prior-based one has better mapping performance at the textureless areas such as the green circular areas.

method (model+MSCKF) leads to drift-free global localization results. We first give out the quantitative comparison between MSCKF and Ground Truth, with a standard deviation of $\{0.0220, 0.0156, 0.0162\}$ (rad) in yaw, pitch and roll, a standard deviation of $\{0.2384, 0.0895, 0.0644\}$ (m) in the x, y and z positions. The whole trajectory is 73.9 m with 0.59 m final drift, the percentage of drift is 0.81%. We then present quantitative comparison between model+MSCKF and Ground Truth, as we can see that results from the proposed approach matches well with the ground truth, with a standard deviation of $\{0.0098, 0.0096, 0.0089\}$ (rad) in yaw, pitch and roll, a standard deviation of $\{0.0498, 0.0230, 0.0175\}$ (m) in the x, y and z positions, which is also better than our previous work by using a simple

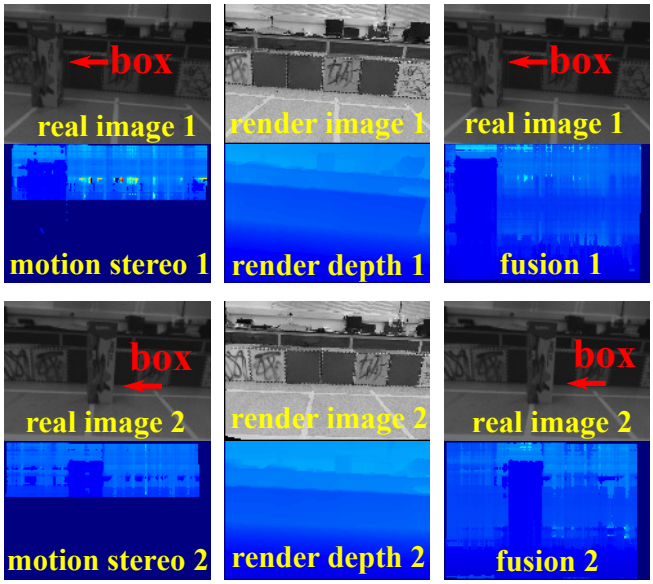


Fig. 6: Depth estimation comparison that the deep blue background denotes invalid estimation. the first column shows the depth estimation results by using original motion stereo, the second column shows the depth prior rendered from the 3D model, and the third column shows the depth estimated from our depth prior-based method. Note that original motion stereo has unstable performance especially at the textureless areas, such as the ground. Two timestamps are selected for comparison, also we can see the online motion stereo can detect the obstacle that was not modeled previously (the box) to adapt to dynamic environments.

EKF framework fused with IMU data. Note that part of the environment is missing in the 3D model, thus the localization accuracy will be affected when the rendered view is not complete. More details can be found in the attached video.

C. Mapping results

To test the online mapping results in dynamic environment, we construct a 3D model of another indoor environment with textured boxes for better visualization effect and do experiments with an additional box inside it. Compared with the depth prior, the online mapping can detect new obstacles for further path planning. And compared with purely motion stereo, the model-aided motion stereo has much better estimation quality at the textureless areas, such as ground and wall. The detailed comparison is shown in Fig. 6, and TSDF result shown in Fig. 7. The real-time mapping process is also shown in the attached video.

VI. Conclusion and Future Work

In this paper, we proposed a global localization and dense mapping solution utilizing a known 3D textured model. We integrate fast model view rendering, image stabilization, edge-based image alignment, global pose fusion, monocular dense mapping to solve the challenging real-time state estimation and environment mapping problem in complex environments. Experimental results show high local accuracy and global consistency of our system. Our solution runs real-time onboard a computationally-constrained platform. In the future, we will improve the mapping quality by utilizing larger cropped image size. Besides obstacle avoidance, We may also refine the 3D model online using onboard visual information to account for the differences between the model and the environment, and the refined model will immediately be used in the rendering thread.

References

[1] G. Schindler, M. Brown, and R. Szeliski, "City-scale location recognition," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–7.

[2] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[3] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 834–849.

[4] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 6, pp. 1052–1067, 2007.

[5] R. Mur-Artal, J. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *Robotics, IEEE Transactions on*, vol. 31, no. 5, pp. 1147–1163, 2015.

[6] 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.

[7] Z. Yang and S. Shen, "Monocular visual-inertial state estimation with online initialization and camera-imu extrinsic calibration," *IEEE Transactions on Automation Science and Engineering*, no. 99, p. 1, 2016.

[8] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*. IEEE, 2011, pp. 127–136.

[9] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 328–341, 2008.

[10] S. S. Zhenfei Yang, Fei Gao, "Real-time monocular dense mapping on aerial robots using visual-inertial fusion," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017.

[11] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2320–2327.

[12] M. Pizzoli, C. Forster, and D. Scaramuzza, "Remode: Probabilistic, monocular dense reconstruction in real time," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2609–2616.

[13] C. Teuliere, E. Marchand, and L. Eck, "3-d model-based tracking for uav indoor localization," *IEEE transactions on cybernetics*, vol. 45, no. 5, pp. 869–879, 2015.

[14] G. Klein and D. W. Murray, "Full-3d edge tracking with a particle filter." in *BMVC*, 2006, pp. 1119–1128.

[15] K. Ok, W. N. Greene, and N. Roy, "Simultaneous tracking and rendering: Real-time monocular localization for mavs," in *IEEE International Conference on Robotics and Automation (ICRA-2016), Stockholm, Sweden*, 2016, pp. 4522–4529.

[16] K. Qiu, T. Liu, and S. Shen, "Model-based global localization for aerial robots using edge alignment," *IEEE Robotics and Automation Letters*, 2017.

[17] M. P. Kuse and S. Shen, "Robust camera motion estimation using direct edge alignment and sub-gradient method," in *IEEE International Conference on Robotics and Automation (ICRA-2016), Stockholm, Sweden*, 2016.

[18] M. Klingensmith, I. Dryanovski, S. Srinivasa, and J. Xiao, "Chisel: Real time large scale 3d reconstruction onboard a mobile device using spatially hashed signed distance fields." in *Robotics: Science and Systems*, 2015.