This paper has been accepted for publication at the
IEEE International Conference on Robotics and Automation (ICRA), Paris, 2020. ©IEEE

Redesigning SLAM for Arbitrary Multi-Camera Systems

Juichung Kuo, Manasi Muglikar, Zichao Zhang, Davide Scaramuzza

***Abstract*— Adding more cameras to SLAM systems improves robustness and accuracy but complicates the design of the visual front-end significantly. Thus, most systems in the literature are tailored for specific camera configurations. In this work, we aim at an adaptive SLAM system that works for arbitrary multi-camera setups. To this end, we revisit several common building blocks in visual SLAM. In particular, we propose an adaptive initialization scheme, a sensor-agnostic, information-theoretic keyframe selection algorithm, and a scalable voxel-based map. These techniques make little assumption about the actual camera setups and prefer theoretically grounded methods over heuristics. We adapt a state-of-the-art visual-inertial odometry with these modifications, and experimental results show that the modified pipeline can adapt to a wide range of camera setups (e.g., 2 to 6 cameras in one experiment) without the need of sensor-specific modifications or tuning.**

Supplementary Material

Video: https://youtu.be/JGL4H93BiNw

I. Introduction

As an important building block in robotics, visual(-inertial) odometry (VO/VIO), or more general, simultaneous localization and mapping (SLAM) has received high research interest. Modern SLAM systems are able to estimate the local motion accurately as well as build a consistent map for other applications. One of the remaining challenges for vision-based systems is the lack of robustness in challenging environments, such as high dynamic range (HDR) and motion blur [1]. Among different approaches that have been explored for better robustness (e.g., [2] [3]), adding more cameras in SLAM systems proves to be effective and is already exploited in successful commercial products, such as Oculus Quest [4] and Skydio [5].

As the workhorse for modern (keyframe-based) SLAM systems, bundle adjustment (BA) like nonlinear optimization naturally generalizes to multiple sensors, including visual-inertial and multi-camera systems, as long as the measurement process is modeled correctly. On the other hand, the design of the so-called front-ends is much less theoretically grounded. Many details, such as initialization, keyframe selection, and map management, are designed heuristically. Moreover, such designs are often tailored to specific sensor setups, and it is not clear to what extent they can be applied to more general sensor configurations. For example, one popular method for selecting keyframes is to consider commonly visible features in the current frame with respect to the last keyframe. While this works well for monocular setups or stereo pairs with highly overlapping field-of-views (FoV), it quickly becomes complicated as more cameras are added, as different cameras may have drastically different view conditions (e.g., the number of features).

The authors are with the Robotics and Perception Group, Dep. of Informatics, University of Zurich, and Dep. of Neuroinformatics, University of Zurich and ETH Zurich, Switzerland— http://rpg.ifi.uzh.ch.
This research was supported by the National Centre of Competence in Research (NCCR) Robotics, through the Swiss National Science Foundation, the SNSF-ERC Starting Grant and Sony R&D Center Europe.

Fig. 1: Multi-camera systems achieve superior performance in perception algorithms and are widely used in real-world applications, such as omni-directional mapping [6], autonomous drones [5], and VR headsets [4]. To facilitate the use of such systems in SLAM, we propose several generic designs that adapt to arbitray multi-camera systems automatically.

To remove the dependence on sensor-specific assumptions and heuristics, we resort to adaptive and more principled solutions. First, instead of using hard-coded rules, we propose an adaptive initialization scheme that analyzes the geometric relation among all cameras and selects the most suitable initialization method online. Second, instead of engineering heuristics, we choose to characterize the uncertainty of the current pose estimate with respect to the local map using the information from all cameras, and use it as an indicator of the need for a new keyframe. Third, instead of relying on the covisiblity graph, we organize all the landmarks in a voxel grid and sample the camera frustums via an efficient voxel hashing algorithm, which directly gives the landmarks within the FoVs of the cameras. These methods generalize well to arbitrary camera setups without compromising the performance for more standard configurations (e.g., stereo).

Contributions: To summarize, the contribution of this work is an adaptive design for general multi-camera VO/VIO/SLAM systems, including

- an adaptive initialization scheme,
- a sensor-agnostic, information-theoretic keyframe selection algorithm,
- a scalable, voxel-based map management method.

Since the proposed method is not limited to specific implementations or sensing modalities, we will use the term

SLAM in general for the rest of the paper.

The paper is structured as follows. In Section II, we review the common methods for initialization, keyframe selection, and map management in visual SLAM. In Section III, we describe our adaptive initialization process based on overlap check. In Section IV, we detail our entropy-based keyframe selection algorithm. In Section V, we introduce our voxel-based map representation for visible points retrieval. Finally, we apply our method to a state-of-the-art VIO system and present the experimental results in Section VI and conclude our work in Section VII.

II. Related Work

A. Initialization

The initialization in SLAM typically incorporates assumptions that are specific to camera configurations. For monocular systems, homography and 5-point relative pose algorithm from [7] are popular ways to obtain the poses of the first two keyframes and the initial map (e.g., [8]), which usually requires the camera to undergo certain motion, such as strong translation and no pure rotation. In contrast, stereo cameras can recover the depth information and initialize the map directly [9], [10]. In multi-camera setups, there could be various ways of combining different cameras depending on the extrinsic parameters. For example, MCPTAM [11] initializes the monocular cameras individually with a known target. The pipeline in Liu *et al.* [12] performs initialization with stereo matching from predefined stereo pairs. While the possible ways for initialization inevitably depend on sensor configurations, we would like a system to be able to select the proper method automatically, instead of hard-coded rules.

B. Keyframe Selection

It is common to maintain a fixed number of keyframes in the front-end as the local map, against which new frames are localized. Hence, the selection of keyframes is crucial for the performance of SLAM systems. In general, the keyframe selection criteria can be categorized into heuristic-based methods and information-theoretic methods.

1) Heuristics-based methods: In many SLAM systems, the keyframe selection criteria are the combinations of different heuristics. We list the most common ones below.

Camera motion: In ORB-SLAM [8], one of the criteria is to check whether the current frame is a certain number of frames away from the last keyframe. Similarly, DSO [13] and SVO [10] select a new keyframe if the current pose is away from the last keyframe by a certain amount of motion.

Number of tracked features: A new keyframe is selected if the number or the percentage of tracked features in the current frame falls below a certain threshold. However, the specific threshold usually varies greatly between different scenarios. This criterion in used in [8], [10], [11], and [14].

Optical flow: The Euclidean norm between the corresponding features from the current frame and the last keyframe. This criterion, for example, is used in [13] and [12].

Brightness change: For direct methods, changes in image brightness caused by camera exposure time and lighting condition makes the tracking against old keyframes difficult. Hence, [13] also uses the relative brightness as a criterion.

Using the combination of different heuristics usually relies on certain assumptions of the sensor configurations and scenes, which makes parameter tuning as well as the application to general multi-camera setups complicated.

2) Information-theoretic methods: These methods rely on more principled metrics and are less common in literature. Das *et al.* [15] chose the keyframes to be included in the BA. Their method favors the frames that bring the most entropy reduction in the map points and essentially selects the most informative keyframes for BA. The criterion from DVO [16] is the most related to ours: it selects keyframes based on an entropy ratio that reflects the uncertainty of the camera pose with respect to the last keyframe. Our method follows a similar idea, but considers all the current keyframes, which reflects the information contained in the entire local map.

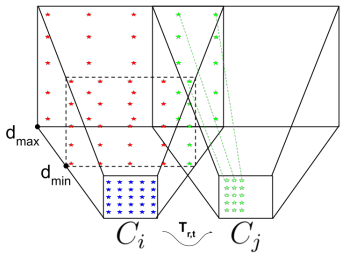

Fig. 2: An illustration of the stereo overlapping check between two cameras, C_i and C_j. The blue stars are the sampled points on the image plane of camera i. The green stars are the 3D points that are successfully projected to camera j, and the red ones are the points that fall out of the image plane.

C. Map Management and Query

To estimate the pose of newly coming frames, the front-end usually needs to find the 2D-3D correspondences between the observations in the new images and the map. A common method is the covisibility check: only search for the matches of the 3D points in the keyframes that reproject onto the new images, such as in [8], [13], [17], [18]. As more cameras are added, the complexity of the covisiblity check increases quadratically, and keyframes from cameras with large common FoVs introduce high redundancy. For example, for stereo pairs with highly overlapping FoVs, it is usually sufficient to keep one of the two frames as keyframes. Obviously, it is not clear how this strategy can generalize to arbitrary camera configurations. To the best of of our knowledge, there is no previous study about efficiently querying map points in a general multi-camera setup.

III. Adaptive Initialization

Our initialization method has no hard-coded assumptions regarding the camera configuration. For any multi-camera setups with known intrinsic and extrinsic calibrations, it is able to select the proper initialization method accordingly, without the need to change the algorithm settings manually. Specifically, it utilizes an overlapping check between the camera frustums to identify all the possible stereo camera pairs. If there exists stereo pairs, the initial 3D points are created from the stereo matching of these stereo pairs. Otherwise, the 5-point algorithm is run on every camera as

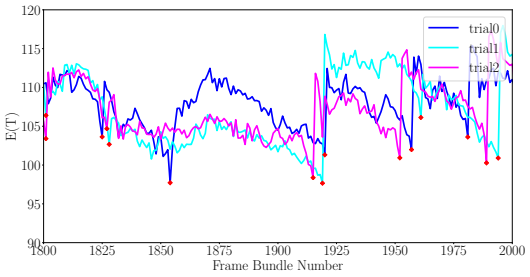

Fig. 3: Negative entropy evolution of 3 runs in EuRoC *MH_01*. $E(\mathtt{T})$ for each run is shown in a different color, and the red dots indicates where a frame is selected as a keyframe. $E(\mathtt{T})$ increases after a keyframe insertion and decreases as the sensor moves away from the map.

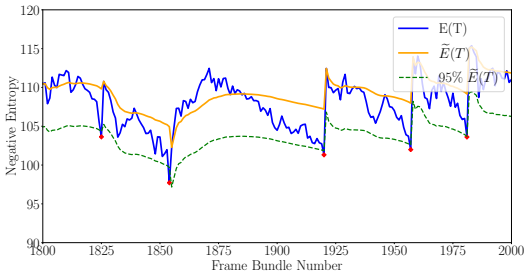

Fig. 4: Running average $\tilde{E}(\mathtt{T})$ and keyframe selection. The running average filter (yellow) tracks the localization quality since the last keyframe. When the negative entropy of the current frame (blue) falls below a certain percentage of the running average (green dash), a new keyframe is selected (red dots) and the running average filter is reset.

in a standard monocular setup, and the map is initialized whenever there exists a camera that triangulates the initial map successfully (i.e., enough parallax, and the camera is not undergoing strong rotation).

The core part of the aforementioned initialization scheme is the overlapping check. The overlapping checking algorithm checks all the possible pairs in a multi-camera configuration, denoted as C_{ij}, where $i, j \in 1 \dots n,\ i \neq j$, and n is the total number of cameras in the system, and finds all possible stereo pairs. For each pair C_{ij}, the algorithm is illustrated in Fig. 2. We denote a 3D point in homogeneous coordinates as $(x, y, z, 1)^\top$. With the camera projection function π, a 2D point $\mathbf{u}$ in the image plane can be back-projected to a 3D point in the camera frame for a depth value z as $\mathbf{p} = \pi^{-1}(\mathbf{u}, z)$. We also know the corresponding relative transformation $\mathtt{T}_{ij}$ from the extrinsic calibration of the camera system. In detail, the overlapping check first uniformly samples (or possibly using different sampling methods) a set of points U_i on the image plane of camera i. Then the points in U_i are back-projected to the minimal and maximal depths d_{min} and d_{max} as $P_{i,max}$ and $P_{i,min}$ respectively. These depths are specified by users and encloses the effective depth range of the initialization process. Given the $\mathtt{T}_{ij}$ and the intrinsics of camera j, we then project the 3D points $P_{i,min}$ and $P_{i,max}$ to camera j as $U_{j,min}$ and $U_{j,min}$ and check whether these points fall in the image plane of camera j. The projection from $\mathbf{u}_i$ in U_i to camera j is considered successful only if both of the backprojected 3D points at d_{min} and d_{max} are within the image plane of camera j. A pair of cameras is considered as a stereo pair if the overlapping ratio, $\frac{\text{\# of Successful Projection}}{\text{\# of Total Samples}}$, is above a user-defined threshold.

The proposed sampling-based method is generic. By using the camera projection/backprojection directly, we can find all stereo pairs across different types of camera models without the need to explicitly calculate the overlapping volume of possibly very different frustums (e.g., between pinhole and fisheye cameras), which can be non-trivial to compute analytically. Moreover, the check can be computed offline, and the valid stereo pairs be directly used at runtime.

IV. Entropy-based Keyframe Selection

The concept of keyframe naturally generalizes to a keyframe bundle for a multi-camera setup, as in [11]. A keyframe bundle contains the frames from all the cameras at the same time. In the following, we will use the terms keyframe and keyframe bundle interchangeably. To determine when a keyframe should be added, we design an entropy-based mechanism. In particular, the local map contains 3D points (organized as keyframes or voxels as in Section V) against which new frames can localize. Intuitively, a keyframe should be selected when the current map is not sufficient for tracking, since new points will be initialized at the insertion of a keyframe. Therefore, we select keyframes based on the uncertainty of the keyframe bundle pose with respect to the current map. Compared with heuristics, our method is more principled, has less parameters (*only 1*) and generalizes to arbitrary camera configurations. In this section, we first provide necessary background on the uncertainties in estimation problems and then describe our keyframe selection method.

A. Uncertainties Estimation in Nonlinear Least Squares

For a parameter estimation problem of estimating $\mathbf{x}$ from measurement $\mathbf{z}$ with normally distributed noise, a common method is to cast the problem as a nonlinear least squares (NLLS) problem. In iterative algorithms of solving NLLS problems, such as Gauss-Newton, the uncertainties of the estimated parameters can be obtained as a side product in each iterative step. Specifically, the normal equation at step i is $(\mathtt{J}^\top \Sigma_{\mathbf{z}}^{-1} \mathtt{J}) \delta \mathbf{x}_i = \mathtt{J}^\top \mathbf{r}(\mathbf{x}_i)$, where $\mathbf{r}(\mathbf{x}_i)$ is the residual given the current estimate $\mathbf{x}_i$, $\delta \mathbf{x}_i$ the optimal update, and $\mathtt{J}$ the Jacobian of $\mathbf{z}$ with respect to $\mathbf{x}$. With first-order approximation, the covariance of the estimate can be obtained by backward propagating the measurement noise to the parameters, which is simply:

$$\Sigma_{\mathbf{x}} = (\mathtt{J}^\top \Sigma_{\mathbf{z}}^{-1} \mathtt{J})^{-1}, \tag{1}$$

which is an important tool to quantify the estimation quality of NLLS solutions [19, Chapter 5, App. 3]. $\mathtt{I}_{\mathbf{x}} = \mathtt{J}^\top \Sigma_{\mathbf{z}}^{-1} \mathtt{J}$ is also known as the Fisher information.

B. Negative Pose Entropy in SLAM

In keyframe-based SLAM, the pose of the current camera is usually obtained by solving a NLLS problem. For example, one common method is to solve a Perspective-n-Points (PnP) problem using the Gauss-Newton method. In this case, the

Algorithm 1: Running average filter.

Input: newest entropy value $E(\mathtt{T})$
Result: Returns the current running average, $\widetilde{E}(T)$
initialization: $n = 0,\ \widetilde{E}(T) = 0$
for *each incoming* $E(T)$ **do**
  $n = n + 1$
  $\widetilde{E}(T) = \widetilde{E}(T) + (E(T) - \widetilde{E}(T))/n$
  $return\ \widetilde{E}(T)$

Fisher information and the covariance of the camera pose can be directly obtained as

$$\mathtt{I}_\mathtt{T} = \mathtt{J}_\mathtt{T}^\top \Sigma_\mathbf{u}^{-1} \mathtt{J}_\mathtt{T},\ \Sigma_\mathtt{T} = (\mathtt{J}_\mathtt{T}^\top \Sigma_\mathbf{u}^{-1} \mathtt{J}_\mathtt{T})^{-1}, \tag{2}$$

where $\mathbf{u}$ is the observation, and $\mathtt{J}_\mathtt{T}$ is Jacobian of $\mathbf{u}$ with respect to the camera pose $\mathtt{T}$. [1] Note that in different NLLS problems, the Fisher information and covariance may be obtained differently (e.g., marginalization in a BA setup).

As mentioned above, our goal is to use the estimate uncertainty of the current pose to indicate whether a new keyframe should be inserted. While (2) provides a principled tool, it is more desirable to have a scalar metric as keyframe selection criteria. Therefore, we utilize the concept of the differential entropy for a multivariate Gaussian distribution, which is $H(\mathbf{x}) = \frac{1}{2}m(1 + \ln(2\pi)) + \frac{1}{2}\ln(|\Sigma|)$ for a m-dimensional distribution with covariance Σ. Note that the magnitude of the entropy only depends on $\ln(|\Sigma|)$. Moreover, in the context of NLLS for pose estimation, from (2), we have $\ln(|\Sigma_\mathtt{T}|) = -\ln(|\mathtt{I}_\mathtt{T}|)$. Since that the Fisher information $\mathtt{I}_\mathtt{T}$ comes for free in the process of solving NLLS problems, we can actually avoid the matrix inversion and use

$$E(\mathtt{T}) \triangleq \ln(|\mathtt{I}_\mathtt{T}|) \tag{3}$$

to indicate how well the camera can localize in the current map. We refer to (3) as *negative entropy*. Since (2) is simply the sum of individual measurements, it is straightforward to incorporate the observations from all the cameras into one single scalar (3) in an arbitrary multi-camera setup.

C. Running Average Filter for Keyframe Selection

Examples of the negative entropy $E(\mathtt{T})$ evolution on the same dataset (*MH_01*) with our multi-camera pipeline (see Section VI) are shown in Fig. 3. We can see that $E(\mathtt{T})$ indeed reflects the localization uncertainty of the pose with respect to the current map. After inserting a new keyframe to the map (red dots on the curves), the negative entropy increases, due to the triangulation of new points; and as the camera moves away from the last keyframe/local map, $E(\mathtt{T})$ decreases until another keyframe is selected. On the other hand, even for the same environment, the absolute value of $E(\mathtt{T})$ varies from run to run. This indicates that using an absolute threshold for $E(\mathtt{T})$ as the keyframe selection criterion is not feasible.

Instead, we propose to track the negative entropy value using a running average filter (see Algo. 1) in the local map and selects a keyframe when $E(\mathtt{T})$ of a frame is below certain percentage of the tracked average $\tilde{E}(\mathtt{T})$. Since we localize the camera with respect to the latest map, and the map remains the same until a new keyframe is added, $\tilde{E}(\mathtt{T})$ essentially tracks the average pose estimation quality with respect to the local map up to the current time. Note that the running average filter is reinitialized every time the map is updated with a new keyframe, since the local map changes as a new keyframe is inserted. Moreover, we use a relative threshold with respect to the running average $\tilde{E}(\mathtt{T})$ so that the selection is adaptive to different environments. This threshold is the only parameter in our keyframe selection method, and it is intuitive to tune. A higher value means more frequent keyframe insertion, and vice versa (see Table II). An example of the running average filter is shown in Fig. 4.

[1] Technically, the Jacobian is with respect to a minimal parameterization of 6 DoF poses, which is omitted here for easy presentation.

V. Voxel-Map Query

For new incoming images, the tracking process in SLAM is responsible to find the correspondences between the observations in the new images and the 3D points in the map. For monocular and stereo setup, this can be efficiently done by searching only for matches of the points in the keyframes that overlap with the new frames. For a general multi-camera setup, since keyframes from different cameras can have high overlap, this method can introduce considerable redundancy. Therefore, we organize the map points in a voxel grid, and directly sample the camera frustums for possible 3D points to match, as proposed in [20].

Map representation: Our voxel-map is a hash table using the voxel hashing technique described in [21]. Each entry in the hash table is a voxel of a user-defined size at a certain position, and it contains the 3D points (from SLAM pipeline) that fall in this voxel. The voxels in the hash table are accessed via a hashing function on the integer world coordinates. Therefore, to get the 3D points around a location of interest, we can directly get the corresponding voxel in constant time. In addition, the map only allocates voxels where there are 3D points and does not store empty voxels. The voxel hash table is synchronized with the map points in the SLAM pipeline.

Map query: To get the map points to match for a multi-camera system, we sample a fixed number of points in the camera frustums and find corresponding voxels in the voxel-map. The points inside these voxels are then used to match the observations in the new images. In this way, it is guaranteed that *all and only* the 3D points within the FoVs of all cameras are retrieved from the map. Moreover, we avoid the process of checking overlapping keyframes from different cameras, which may have many points in common and introduce redundant computation.

Note that we only use voxel-map for querying visible landmarks. Keyframes are still selected for triangulation and potentially bundle adjustment.

VI. Experiments

To validate the proposed method, we applied the aforementioned adaptations to a state-of-the-art keyframe-based visual-inertial odometry pipeline that consists of an efficient visual front-end [10] and an optimization-based backend similar to [22]. We performed experiments on both simulated

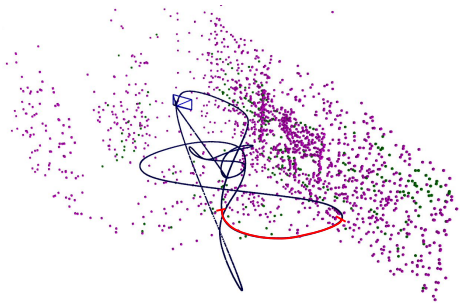

Fig. 5: Simulated figure 8 trajectory in the simulation environment. The trajectory was estimated by running the adapted VIO pipeline with 5 cameras. The segment where the monocular setup lost track is marked in red. The magenta dots are the tracked landmarks by SLAM systems.

and real-world data. In simulation, we verified the robustness and analyzed several properties of the pipeline with different multi-camera configurations. For real-world data, we first tested the stereo setup with the EuRoC dataset [23] to show that the proposed method performs on par with standard methods but is much easier to tune. We then tested the multi-camera setup with the AutoVision dataset [24]. For quantitative evaluation of accuracy, we follow the evaluation protocol in [25]. We repeated the experiment on each sequence for 5 runs using the same setting unless specified otherwise. In each of the experiment, we kept the parameters *the same* for different camera configurations.

A. Simulation Experiment

We tested the pipeline on a drone with various camera configurations: 2 cameras (a front mono; a side mono), 3 cameras (a front stereo; a side mono), 4 cameras (a front stereo; a side stereo), and 5 cameras (a front stereo; a side stereo; a down mono). We refer the reader to the accompanying video for the visualization of our setup. We set the drone to fly a figure 8 trajectory in the environment (Fig. 5). Note that a monocular setup from either the front or side stereo failed when the drone went around a textureless pillar (marked in red in Fig. 5), and the corresponding quantitative results are omitted. Next, we analyzed the accuracy and timing, and the performance of voxel-map and keyframes.

Accuracy: The relative pose error of different camera configurations is shown in Fig. 6. Adding more cameras to the system improved the trajectory estimation accuracy, but the improvement became marginal after the 3-camera configuration. This is because adding the third camera formed a stereo pair (front stereo) compared with the 2-camera configuration, which made direct triangulation possible.

Timing: The total front-end time for different configurations is shown in Fig. 7 (left). As we increased the number of cameras in the configuration, we observe an increase in the total processing time of the front-end. The increase in time is not as significant between the 4 and 5 camera configurations, as the 5th camera (downlooking) did not produce as many landmarks as the other cameras.

Voxel-map vs. Keyframes: In general, the voxel-map method retrieved more landmarks (Fig. 7 middle) than the keyframe based method, because some of the visible landmarks in the current frame may not be stored in nearby

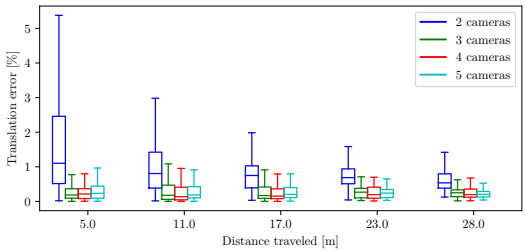

Fig. 6: Overall relative translation error in simulation for 5 runs.

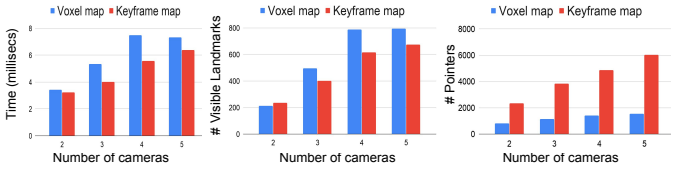

Fig. 7: Comparison of the proposed voxel-map with standard keyframes for different camera configurations (2 to 5 cameras). **Left**: total time for the front-end in VIO. **Middle**: retrieved landmarks for matching from the map. **Right**: number of references/pointers to landmark positions.

keyframes. However, the front-end consumed more time in our experiment, and we assume that it can be further reduced by optimizing our voxel-map implementation. In terms of memory footprint (Fig. 7 right), the voxel-map increased much slower than keyframes. The reason is that the keyframe-based map stores landmark observations in each keyframe. For a multi-camera setup with large FoV overlap, it is very likely different cameras observe the same landmarks, resulting in redundant copies in keyframes. In contrast, the voxel-map stores the references only once.

B. Real-world Experiment

1) EuRoC Dataset: We tested the multi-camera VIO pipeline on EuRoC dataset for the stereo setup. The number of keyframes in the sliding window was set to 10. To show the effect of the relative negative entropy for keyframe selection, we also experimented with different relative entropy thresholds. We use the notation "e**r**-**m**" to denote our experimental configurations, where **r** is the entropy threshold in percentage, and **m** the map representation used (i.e., voxel or keyframes). The default pipeline that is carefully tuned for stereo setups is denoted as "default-kf".

The median values of the absolute trajectory error in 5 runs are shown in Table I. While there is no definite winner, the adapted pipeline in general performed similar or better than the default pipeline. This can also be confirmed from the odometry errors in Fig. 8 (we select three sequences only due to the limit of space). The adapted pipeline has lower estimate error in 10 out of 11 sequences and the entropy ratio of 98% has the most. Regarding the number of keyframes, it is clearly seen in Table II that increasing the relative entropy ratio resulted in more keyframes. In addition, for relative entropy ratio of 95%, fewer keyframes were selected in general but the accuracy was still similar to the default pipeline according to Table I. This indicates that the proposed method selected keyframes more effectively and introduced less redundancy than the default pipeline.

To summarize, as a generic pipeline, our method performed at least similarly good compared with a carefully

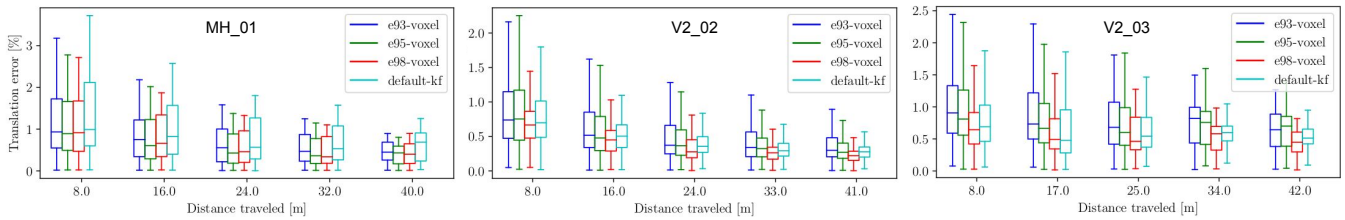

Fig. 8: Relative translation error percentages from the EuRoC dataset with BA.

TABLE I: Median RMSE (meter) on EuRoC dataset over 5 runs. Lowest error highlighted in **bold**.

Algorithm	MH_01	MH_02	MH_03	MH_04	MH_05	V1_01	V1_02	V1_03	V2_01	V2_02	V2_03
default-kf	0.140	0.078	0.091	**0.119**	0.330	0.042	0.070	0.047	0.056	0.066	0.127
e93-voxel	0.104	0.390	0.107	0.177	0.262	**0.038**	**0.036**	0.043	0.080	0.103	0.169
e95-voxel	**0.078**	0.084	0.093	0.182	0.237	0.040	0.047	0.049	0.056	0.087	0.171
e98-voxel	0.095	**0.074**	**0.088**	0.128	**0.180**	0.039	0.053	**0.041**	**0.046**	**0.057**	**0.111**

TABLE II: Average number of keyframes for 5 runs in EuRoC sequences.

Algorithm	MH_01	MH_02	MH_03	MH_04	MH_05	V1_01	V1_02	V1_03	V2_01	V2_02	V2_03
default-kf	64.00	57.80	91.40	76.00	70.20	70.60	119.60	238.80	74.80	172.00	281.40
e93-voxel	46.00	46.30	67.80	58.80	61.80	52.80	56.20	120.40	30.80	63.40	86.80
e95-voxel	71.20	66.00	87.00	74.40	75.80	76.40	86.80	160.00	39.80	85.00	107.80
e98-voxel	154.20	137.20	181.20	138.60	143.60	176.80	177.80	305.20	84.40	169.00	203.60

TABLE III: The average number of keyframes by different keyframe selection criteria for monocular and stereo setups.

Algorithm	MH_01	MH_02	V2_01	V2_02
heuristic, mono	202.75	190.75	150.75	379.75
heuristic, stereo	90.00	117.25	84.75	204.5
entropy, mono	129.5	128.25	100.00	193.5
entropy, stereo	122.25	125.25	98.5	195

TABLE IV: Different trajectory error metrics from the multi-camera pipeline on the Science Park day sequence. The first row contains the absolute RMSE of the full trajectory (547.488 m)

Metric	F	FR	FRB
Abs. Trajectory error (meter)	1.184	2.366	1.766
Rel. Trans. Percentage @ 200m	0.582	1.808	1.320
Rel. Trans. Percentage @ 400m	0.642	1.07	0.760

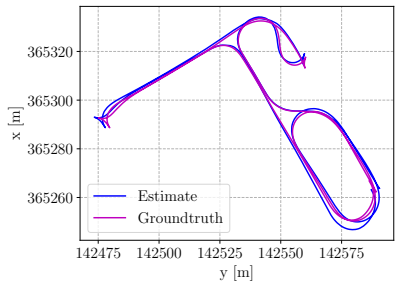

Fig. 9: Top view of the estimated and groundtruth trajectory of the FRB configuration from the Science Park day sequence.

tuned stereo pipeline, and our method was able to achieve similar accuracy with fewer keyframes. More importantly, we would like to emphasize that our method has *only one* parameter for keyframe selection, which makes the task of parameter tuning much easier.

Sensor Agnostic We also performed an experiment comparing the number of selected keyframes between monocular and stereo configurations. We only ran the visual front-end in this case to remove the influence of the optimization back-end, which caused the different keyframe numbers between Table II and III. The average number of keyframes on some sequences in EuRoC is shown in Table III. The heuristic method selected drastically different numbers of keyframes between monocular and stereo configurations because they had to be tuned differently for these configurations. In contrast, our entropy-based method selected very similar numbers of keyframes. This is due to fact that our method essentially summarizes the information in the map instead of relying on camera-dependent quantities. In particular, the stereo pair in EuRoC dataset has largely overlapping FoVs, and thus the visible areas of the environment were similar for monocular and stereo setups, leading to similar information for our keyframe selection method.

2) AutoVision Dataset: We evaluated our pipeline on the Science Park day sequence, which is a large-scale outdoor sequence in a autonomous driving scenario. The trajectory is 547.448 m long and the maximum speed is 3.941 m/s. Following [12], we tested our pipeline with F, FR, and FRB configurations. The trajectory errors, computed in the same way as in [12], are shown in Table IV, and the estimated trajectory (FRB) in Fig. 9. While the estimation accuracy is acceptable and proves the effectiveness of our method, we indeed observed that the accuracy of the trajectory estimates does not necessarily increase as we add more cameras to the pipeline. We suspect that the reason to be the inaccurate extrinsic (similar behavior can be observed in [12]).

VII. Conclusion

In this work, we introduced several novel designs for common building blocks in SLAM to make an adaptive system for arbitrary camera configurations. In particular, we proposed an adaptive initialization scheme that is able to automatically find the suitable initialization method, an information-theoretic keyframe selection method that incorporates the information from all cameras elegantly and a voxel-map representation from which we can directly retrieve the landmarks in the camera FoVs. We applied these techniques to a state-of-the-art VIO pipeline, and extensive experimental results showed that the resulting pipeline was able to adapt to various camera configurations with minimum parameter tuning.

REFERENCES

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

[2] Z. Zhang, C. Forster, and D. Scaramuzza, "Active exposure control for robust visual odometry in hdr environments," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2017. 1

[3] A. Rosinol Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate SLAM? combining events, images, and IMU for robust visual SLAM in HDR and high speed scenarios," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 994–1001, Apr. 2018. 1

[4] "Oculus Quest," https://www.oculus.com/quest/. 1

[5] "Skydio R1," https://robots.ieee.org/robots/skydior1/. 1

[6] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, and J. Weaver, "Google street view: Capturing the world at street level," *Computer*, 2010. 1

[7] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 756–777, 2004. 2

[8] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015. 2

[9] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017. 2

[10] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE Trans. Robot.*, vol. 33, no. 2, pp. 249–265, 2017. 2, 4

[11] A. Harmat, I. Sharf, and M. Trentini, "Parallel tracking and mapping with multiple cameras on an unmanned aerial vehicle," in *Intelligent Robotics and Applications*, 2012. 2, 3

[12] P. Liu, M. Geppert, L. Heng, T. Sattler, A. Geiger, and M. Pollefeys, "Towards robust visual odometry with a multi-camera system," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2018. 2, 6

[13] J. Engel, V. Koltun, and D. Cremers, "Direct Sparse Odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018. 2

[14] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, pp. 1004–1020, 2018. 2

[15] A. Das and S. Waslander, "Entropy based keyframe selection for multi-camera visual slam," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2015. 2

[16] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGB-D cameras," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2013. 2

[17] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *IEEE ACM Int. Sym. Mixed and Augmented Reality (ISMAR)*, Nara, Japan, Nov. 2007, pp. 225–234. 2

[18] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2014, pp. 15–22. 2

[19] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003, 2nd Edition. 3

[20] M. Muglikar, Z. Zhang, and D. Scaramuzza, "Voxel map for visual slam," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2020. 4

[21] M. Niessner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3d reconstruction at scale using voxel hashing," *ACM Trans. Graph.*, 2013. 4

[22] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial SLAM using nonlinear optimization," *Int. J. Robot. Research*, 2015. 4

[23] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *Int. J. Robot. Research*, vol. 35, pp. 1157–1163, 2015. 5

[24] L. Heng, B. Choi, Z. Cui, M. Geppert, S. Hu, B. Kuan, P. Liu, R. Nguyen, Y. C. Yeo, A. Geiger, G. H. Lee, M. Pollefeys, and T. Sattler, "Project autovision: Localization and 3d scene perception for an autonomous vehicle with a multi-camera system," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2019. 5

[25] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2018. 5