ABSTRACT

The basis for most vision based applications like robotics, self-driving cars and potentially augmented and virtual reality is a robust, continuous estimation of the position and orientation of a camera system w.r.t the observed environment (scene). In recent years many vision based systems that perform simultaneous localization and mapping (SLAM) have been presented and released as open source. In this paper, we extend and improve upon a state-of-the-art SLAM to make it applicable to arbitrary, rigidly coupled multi-camera systems (MCS) using the MultiCol model. In addition, we include a performance evaluation on accurate ground truth and compare the robustness of the proposed method to a single camera version of the SLAM system. An open source implementation of the proposed multi-fisheye camera SLAM system can be found on-line `https://github.com/urbste/MultiCol-SLAM`.

- 利用 MultiCol model 扩展出的sota slam 用于MCS
- ground truth的性能评估
- 和单目进行鲁棒性比较
- 开源该原理下的的一个实现multi-fisheye camera SLAM system

INTRODUCTION

The accurate reconstruction of an observed scene from sets of ordered images has a long history in aerial (Kraus, 2004) and close-range photogrammetry (Luhmann et al., 2006). Usually, the object and reconstruction setup is well defined and the scene observations using high-resolution cameras are well planed. Thus, the connectivity between multiple camera positions is known or easily established and off-line bundle adjustment over the cameras and scene structure is performed yielding accurate results. In addition, initial values for the exterior and interior camera orientations are mostly available from external sensors, passpoints and accurate calibration.

- 航拍和近焦摄影(aerial and close-range photogrammetry)领域中SfM的历史，高分辨率相机和well-define的观测对象以及三维重建步骤
- MCS 结构中很容易解算离线的BA，从而进行SfM
- 内外参初值可以通过sensors、passpoints和标定给定

In the computer vision community, the direction of research is called SfM and relaxes many constrains about scene and camera geometry that are assumed in classical photogrammetry. Advances in projective geometry (Hartley and Zisserman, 2008) and visual feature research (Weinmann, 2013) enabled the off-line reconstruction of large scenes from unordered sets of images and photo collections (Wu, 2013, Snavely et al., 2006, Agarwal et al., 2009, Wu et al., 2011, Szeliski, 2010, Sweeney et al., 2015b, Triggs et al., 2000). But essentially, the SfM methods solve the exact same problem as in aerial and (close-range) photogrammetry, i.e. reconstruction of scene and cameras. The main difference lies in the initialization of the bundle adjustment through direct relative orientation methods for calibrated (Stewenius et al., 2006, Hartley and Zisserman, 2008, Kneip et al., 2012) or uncalibrated (Barreto and Daniilidis, 2005, Kukelova et al., 2015) cameras and a simultaneous connectivity estimation using only natural image features.

- 介绍SfM的几个经典研究，并强调与航拍和近焦摄影的photogrammetry是一回事，但主要区别在于和SfM系列的研究相比，photogrammetry在BA的初始化上采用相对定位并仅仅依赖自然图像特征进行连通性估计（连通性估计用于局部图结构的成对关系判断，即判断两个节点是否属于同一类别时要考虑节点和周围节点的连通性）

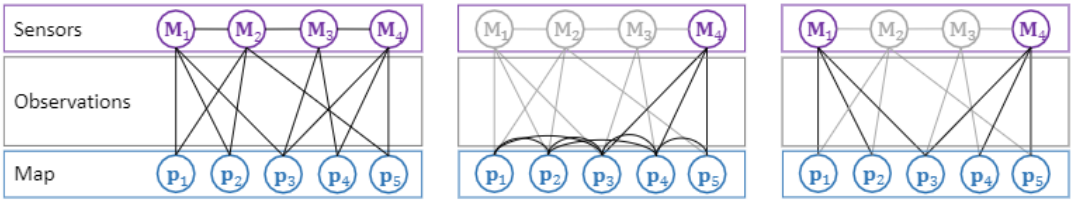
In both communities, the scene is basically assumed to be static and the reconstruction is done off-line doing batch optimization over the entire scene. Hence, correspondence information (features) can be exhaustively extracted and matched a-priori and has no temporal coherence. Latter however, is the case for live video frames from a moving camera. In addition, the pose change (baseline and orientation) between subsequent frames can be very small.

- 二者共同点是静态场景假设和离线批优化重建；通信特征可以先验提取和匹配，不需要考虑时间一致性；后续帧（subsequent frames）的改变可以非常小

methods (Davison et al., 2004, Montemerlo et al., 2002, Davison et al., 2007, Montemerlo and Thrun, 2007) to estimate the sensor pose from continuous sensor updates (live video). In recent years, basically two streams emerged from this line of research, namely filter- and keyframe-based SLAM techniques that will be described in more detail in the following. The basic idea behind both approaches is that not all poses, observations and uncertainties from a continuous video stream can (and have to) be integrated into the solution of the SLAM problem. Outstanding work of (Dellaert and Kaess, 2006) and (Strasdat et al., 2012) give a detailed analysis of both methods and close the gap between SLAM, SfM and classical bundle adjustment by generalization of the entire reconstruction problem using graphical models.

- 在机器人社区出现了基于滤波和关键帧匹配的两种slam技术流派，但基本思想都是并非所有的pose，观测信息和噪声都要参与整合。

Figure 1 depicts the structure in an undirected graph (Markov Random Field) for a toy example. In total, four frames were recorded that observe a scene consisting of five map points. The poses are connected to map points by edges (observations) and the poses themselves are connected by state transitions (e.g. inertial sensor). An optimal BA solution is given (learned) by estimating the ML solution for the depicted graph and could be computed efficiently in this example. Speaking in terms of graphical models, the ML solution corresponds to the joined probability distribution over all parameters (poses and map points). Now, lets grow he graph by adding poses and map points. With each frame, the computational complexity increases and quickly exceeds the runtime constrains for real-time applications. Thus, we need a way of thinning out the graph. Using a filter-based approach, historic poses are marginalized out of the joint distribution by integration over all other parameters. The resulting graph is depicted in Figure 1b. The issue with filter-based SLAM becomes visible. To marginalize the current pose from the joint distribution, all potentials between connected variables need to be stored and updated with every frame and thus the number of map points will be very limited.



(a) SLAM as Markov Random Field (b) Filter-based SLAM (c) Keyframe-based SLAM

- a：pose通过惯导传感器和两帧之间观测得到的匹配点来给定；通过机器学习方法可以求解出BA的最优解
- b：滤波法排除了历史姿态带来的影响
- c：关键帧法相较于滤波法的计算更高效，其每次和关键帧进行比照，删除了其他点，而不是在概率中稀疏图形使其边缘化

Thus in this paper, a keyframe-based SLAM pipeline will be used as the basis for the continuous estimation of map and MCS pose. An additional benefit of using keyframes is, that the map points and observations that are connected to certain keyframes can be easily updated when the MCS revisits a place that changed.

- 该文章基于关键帧法对map和MCS的pose进行连续估计，并指出这样做还有一个好处:当MCS重返一个发生了变化的位置时，可以轻松更新相关信息

SOTA(related work)

In recent years, plenty of keyframe-based SLAM systems were proposed and many of them are publicly available. Probably the first real-time SLAM system that was based on performing local and global BA over keyframes is PTAM, by Klein and Murray(Klein and Murray, 2007). One key to success was to split the tracking and mapping components of the system into separate threads running in parallel on a dual-core CPU. In this way, the mapping thread decides which camera poses to keep and store as keyframes and performs local and global BA asynchronously to the tracking thread. Latter runs at frame rate and performs matching of map point and camera pose estimation. Subsequently, PTAM was improved by adding edge features (Klein and Murray, 2008) and, with the increase in computing power, was implemented on a mobile phone (Klein and Murray, 2009). The use of image patches as features and the lack of loop closing mechanisms already suggests that large scale operation could be an issue using PTAM as storing, updating and indexing of image patches is costly. In addition, the global BA is performed over the entire scene, limiting its applicability to smaller workspaces.

- PTAM(Klein and Murray, 2007)是第一个关键帧法SLAM框架，其一个成功点在于利用一个双核CPU并行执行两个线程来进行tracking和mapping；mapping线程决定使用并存储哪一个相机的pose而tracking线程进行全局或局部的异步BA。继而执行地图点匹配和相机位姿估计
- 介绍了一些PTAM的应用和改进，但缺少回环而是采用全局BA，同时使用贴图作为特征，带来了昂贵的性能开销，因此只能限制在较小的工作空间使用

To extend the range of PTAM, the authors of (Strasdat et al., 2010) propose a so-called scale-drift aware SLAM system. First, the keyframe decisions and feature initialization is moved to the tracking thread. Despite having a higher computational burden, the tracking becomes more stable, as features and keyframes are not initialized asynchronously and are immediately available for pose estimation. Due to the incremental nature of SLAM, the trajectory estimates start to drift over time, i.e. if the camera visits the same place twice, it will have a different exterior orientation and the reprojected map points exhibit large residuals to the measured image features. Still, if one is able to automatically detect similar places using place recognition methods (Garcia-Fidalgo and Ortiz, 2015), loop closing can be performed. The loop closing mechanism in (Strasdat et al., 2010) is based on SURF features and a dense surface model. Then, the trajectory is corrected

- 尺度漂移感知SLAM (scale-drift aware SLAM)(Strasdat et al.,2010) 用来拓展PTAM的范围，解决了上述两个影响工作空间大小的问题。对线程功能进行了修改；并基于SURF增加了闭环机制；以尺度作为一个明确的优化变量来克服增量式带来的轨迹漂移问题。

Seminal work (Strasdat et al., 2011) focused on keyframe optimization, selection and constraints. Figure 2 depicts the double window approach. The inner window of active keyframes models the local area. Poses and map points are optimized using local BA. The outer window stabilizes the inner window and connects it to the rest of the trajectory. The question remains how to determine the connectivity between keyframes. In (Strasdat et al., 2011), the *co-visibility* is introduced. Instead of creating connections between keyframes based on geodesic or euclidean distance or temporal constraints, image features are used. By projecting map points to adjacent keyframes and matching their corresponding descriptors, a co-visibility weight can be calculated, that expresses how many equal features are visible in both keyframes. Apart from being able to update the connectivity if the scene changes, occlusions can be handled a lot better.

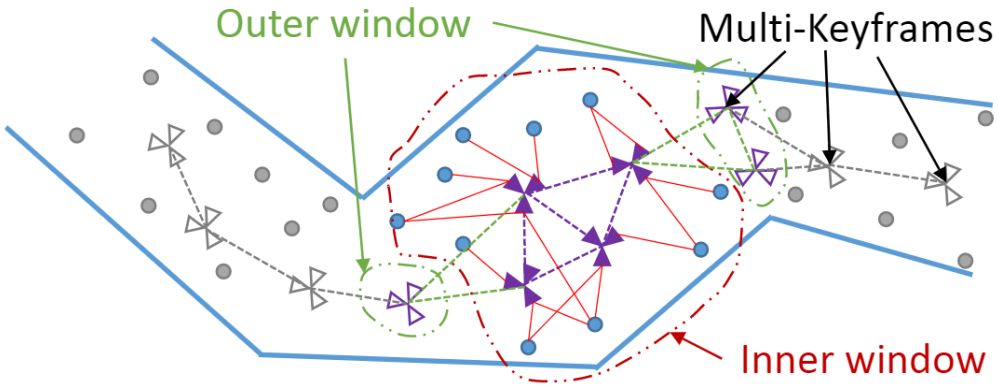


Figure 2. Principle of double window SLAM.

- double window（活动窗口）(Strasdat et al., 2011) 是一项开创性的工作，内联窗口对局部的活跃的关键帧进行建模，其中的pose和匹配点通过局部BA优化；外联窗口用于稳定内窗并连接其余轨迹。
- 遗留的问题是关键帧之间的连通性（相似度）如何确定，co-visibility算法 (Strasdat et al.,2011)与传统的基于几何计算连通性的方案不同，通过投影点到相邻关键帧并匹配描述子计算co-visibility权重，来描述两个关键帧有多少等价的可见特征（通常论文中提到的co-visibility的概念指一个物体在多帧中共同可见，这里指代的是引文中提到的co-visibility points的计算，该算法适用于double windows的优化：以setA为回环中的潜在点集，Vi为每个关键帧，算法的设计目的为检查A中有多少点在Vi中可见。需注意该算法在实现上不同于后面所介绍的从orbslam就沿用的co-visibility graph概念，二者仅在概念上有关联性），被证实这样的改进能克服该框架因为环境变化和遮挡带来的干扰。

In (Pirker et al., 2011), a complete SLAM system was proposed including loop detection, re-localization and handling of long-term dynamics. Co-visibility and local map querying was not only constrained by image to image matches, but also on the level of map points. A Histogram of Cameras (HoC) descriptor (Pirker et al., 2010) is used to determine which map points are visible from the current camera pose. SIFT features are used and extraction and matching is performed on the GPU to achieve real-time performance, which limits its applicability on mobile platforms having only limited computing power.

- (Lim et al., 2014)介绍了一个完整的SLAM系统，包括回环检测、重定位、long-term dynamics，并采用了HoC描述子和SIFT特征

In (Lim et al., 2014), the authors also build on the double-window optimization and co-visibility ideas but use the FAST detector coupled with BRIEF to exploit the performance of binary features. Using BRIEF and FAST decreases the time for feature extraction and matching significantly, however, this detector descriptor combination is not rotation and scale invariant and thus limited to in-plane trajectories (like driving cars). In addition, points are only queried and tracked from the last keyframe. Thus, the local map structure is not fully exploited. ORB-SLAM (Mur-Artal and Tardos, 2014, Mur-Artal et al., 2015, Mur-Artal and Tardos, 2016) is the latest state-of-the-art feature- and keyframe-based SLAM system, that is available on-line. As the name suggests, ORB features are used, being rotation and scale invariant to some degree. The map is reused and queried efficiently using co-visibility computed on ORB features. Place recognition is based on a binary BoW method (Gálvez-López and Tardós, 2012) and a new map initialization heuristic is introduced, that dynamically switches between fundamental and homography estimation. Robust keyframe and map-point culling heuristics ensure a high

- 该研究(Lim et al., 2014),结合了double window和co-visibility的思想，但使用了FAST和BRIEF抽象二进制特征来有效优化性能
- 介绍了ORB-SLAM(Mur-Artal and Tardos, 2014, Mur-Artal et al., 2015, Mur-Artal andTardos, 2016)采用ORB特征通过尺度和旋转不变性有效优化了地图查询和重用的效率；并采用了BoW method进行地点识别；引入了一种新的地图初始化的启发式算法；基本矩阵和单应性矩阵估计动态切换，鲁棒关键帧和匹配点去除算法保证了地图的高质量

The methods described so far either use a single camera (Monocular SLAM) or a stereo configuration (Stereo SLAM). CoSLAM (Collaborative SLAM) (Zou and Tan, 2013) aims at combining the maps build by multiple cameras moving around in dynamic environments independently. The authors introduce inter-camera tracking and mapping and methods to distinguish static background points and dynamically moving foreground objects. In (Heng et al., 2014), four cameras are rigidly coupled on a MAV. Two cameras are paired in a stereo configuration respectively and self-calibrated to an IMU on-line. The mapping pipeline is similar to ORB-SLAM and also uses ORB descriptors for map point assignment. Additionally, the authors propose a novel 3-Point algorithm to estimate the relative motion of the MAV including IMU measurements. Most recent work on multi-camera SLAM is dubbed MC-PTAM (Multi-Camera PTAM) (Harmat et al., 2012, Harmat et al., 2015) and is build upon PTAM. In a first step (Harmat et al., 2012), the authors changed the perspective camera model to the generic polynomial model that is also used in this paper. This induces further changes, e.g. relating the epipolar correspondence search that now has to be performed on great circles on the unit sphere instead of point to line distances in the plane. In addition, significant changes concerning the tracking and mapping pipeline had to be made to include multiple rigidly coupled cameras. Keyframes are extended to MKFs as they now hold more than one camera. As PTAM, their system uses patches as image features and warps them prior to matching. Still, the system lacks a mapping pipeline that is capable to perform in large-scale environments. Subsequent work (Harmat et al., 2015) improved upon (Harmat et al., 2012) and is partly similar to the SLAM system developed in this thesis in that it uses the same camera model and g2o to perform graph optimization. On top, the authors integrated an automated calibration pipeline to estimate the relative orientation of each camera in the MCS. Still the system uses the relatively simple mapping back-end of PTAM instead of double-window optimization that is used in this thesis and has proven to be superior. In addition, image patches are used as features making place recognition, loop closing and the exploration and storage of large environments critical.

- 介绍了CoSLAM(Zou and Tan, 2013)，和之前的技术路线不同，利用了多个独立移动的相机在动态环境中建图，并能进行相机的跟踪和映射与区分前后景
- (Heng et al., 2014)利用四个紧耦合相机两两配对结合在线IMU用于MAV(微型飞行器)，使用类似ORB描述子并提出一种新的3-points算法计算MAV的相对运动
- 最新的多相机研究是MC-PTAM(Harmat et al., 2012,Harmat et al., 2015)，在PTAM基础上进行了较大修改，将透视相机模型更换为了本文中的通用多项式模型，几何计算上发生了较大的变化以至于必须使用多相机视图；沿用了PTAM中的贴图特征但进行了扭曲。但即使如此该方案依然不便于大规模的环境中执行。但后续工作中其进行了较大的改进，用了和本文相似的思路，相同的摄像机模型和g2o图形优化；并集成自动校准的pipeline估计每个相机的相对方位；取缔double window而使用简单的后端，被证明有效。其贴图特征对于地点识别和环境探索很重要。

Thus far, all approaches were based on local point image features. Hence, the reconstructed environment will stay relatively sparse even if hundreds of features are extracted in each keyframe. This makes it difficult for autonomous vehicles or robots that explore their surrounding to automatically analyze and extract object structure or texture information. Thus, most of the time, camera localization is coupled with laser scanners (Lin et al., 2012), structured light (Kerl et al., 2013), yielding structured object information. Recent work on semi-dense (Forster et al., 2014, Engel et al., 2014) and dense (Newcombe et al., 2011, Concha and Civera, 2015) camera-based SLAM systems make use of a single camera to estimate dense scene structure instead of reconstructing only point features.

- 上述算法都是局部特征，所重构的环境相对稀疏，这使得机器人难以探索环境的纹理和结构
- 引用了几个最近的半稠密和稠密的工作

LSD-SLAM (Engel et al., 2014) is a semi-dense approach that runs on a single CPU in real-time, in contrast to dense methods (Newcombe et al., 2011) that need heavy GPU support. Using direct image-alignment by minimizing the photometric error between image discontinuities, the method skips the costly feature extraction and matching stage of all feature-based SLAM systems. The time saved compensates for the increased BA runtime, as a huge number of observations is included. In addition, all scale-drift aware loop closing and large scale double window optimizations are included, making LSD-SLAM a state-of-the-art approach that also runs in real-time. However, loop closing uses FAB-MAP (Cummins and Newman, 2010) for place recognition and thus requires SURF features to be extracted. Subsequent work extended the method to mobile phones (Schöps et al., 2014), stereo (Engel et al., 2015) as well as omnidirectional cameras (Caruso et al., 2015). Instead of coupling camera pose estimation and semi-dense mapping, in (Mur-Artal and Tardós, 2015) a semi-dense extension to ORB-SLAM is presented. The semi-dense map is reconstructed from feature-based keyframes using depth consistency tests and a novel correspondence search. The semi-dense reconstruction is not obtained in real-time but is calculated in a CPU thread running in parallel to tracking and mapping. The methods yields superior performance compared to LSD-SLAM and it seems that the decoupling is advantageous, especially in dynamic scenes.

- LSD-SLAM (Engel et al., 2014) 是直接法的SOTA，可以在单CPU上实时运行，其跳过了所有的特征处理与匹配阶段，基于大量的观察节省了时间和开销
- (Mur-Artal and Tard´os,2015)对ORB-SLAM进行了半密集扩展，被证明相较于LSD-SLAM在动态环境中表现更好

CONTRIBUTIONS

3. CONTRIBUTIONS

We will extend the state-of-the-art ORB-SLAM to multi-fisheye camera systems using MultiCol (Urban et al., 2016b). Our contributions to ORB-SLAM (and ORB-SLAM2 respectively) are the following:

1. The introduction of Multi-Keyframes (MKFs).
2. A hyper-graph formulation of MultiCol.
3. Multi-Camera loop closing.
4. Minimal non-central absolute pose estimation methods for re-localization (Kneip et al., 2013).
5. Different initialization method, based on the essential matrix.
6. Several performance improvements.

- 描述了文章贡献

In order to use MultiCol, the concept of Multi-Keyframes (MKFs) is introduced. Employing a generic camera model (Scaramuzza et al., 2006) allows to couple arbitrary central cameras to the MCS. Instead of employing image patches as features (cf. MC-PTAM), compact binary descriptors proved to be the state-of-the-art when it comes to efficient large-scale re-localization, tracking and loop closing.

- 提出了MKFs多关键帧的概念；使用通用相机模型；允许MCS将任意相机作为中央相机；不使用贴图特征而采用二进制描述子（SOTA）

FRAMEWORK

4. FRAMEWORK

In this chapter, the proposed SLAM system is introduced. As mentioned previously, the basic structure of our system is build upon ORB-SLAM (Mur-Artal and Tardos, 2014, Mur-Artal and Tardos, 2016). The proposed tracking and mapping system is dubbed MultiCol-SLAM. Figure 5 depicts an overview of the system. In general it is divided into multiple threads running in parallel and taking care of different aspects. For the sake of clarity, the loop detection thread is omitted in this figure. Each adaption to the original ORB-SLAM system is highlighted in red and will be explained in more detail in the following. Two of the most profound adjustments in MultiCol-SLAM compared to ORB-SLAM are the introduction of Multi-Keyframes (MKFs), i.e. a keyframe consists of multiple images and the use of fisheye cameras. Both novelties involve some significant changes to the basic design, e.g. bundle adjustment, pose estimation, map point triangulation and relative orientation computation.

- 该文章中框架是在基础架构ORB-SLAM上提出，所提出的用于跟踪和匹配的系统是MultiCol-SLAM，和ORB-SLAM相比，主要有两个创新：鱼眼镜头和MKFs，图5（下）是框架

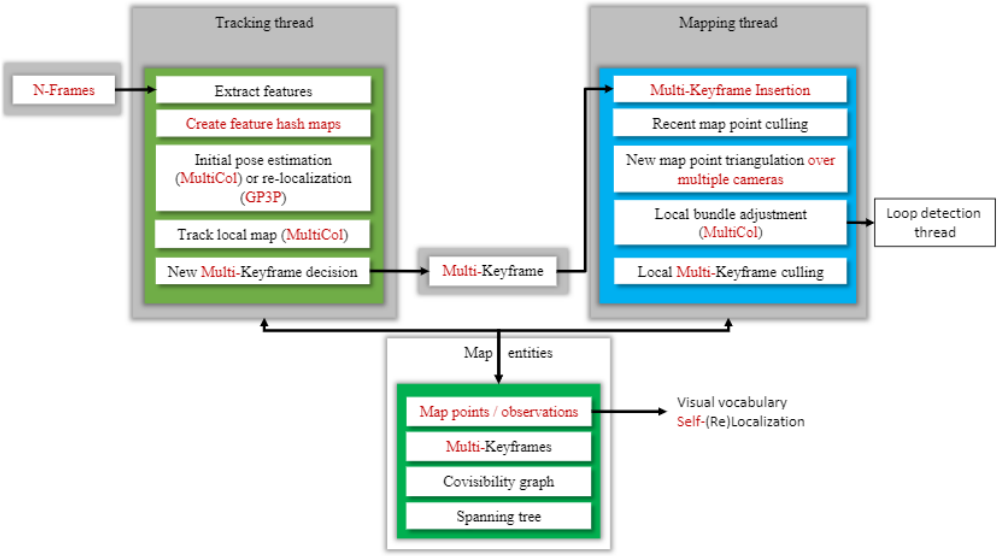


Figure 5. Depicts the MultiCol-SLAM framework without the loop closing thread. Red text depicts the modules where significant difference between our method and ORB-SLAM exist.

With every new set of incoming camera images, the tracking thread extracts point features from every image. Then, they are stored in a continuous vector, that will later be used to identify and match points across MKFs and mask outliers. To ensure a fast indexing and querying of feature to camera mappings, we use hash maps (unordered maps in C++) that provide constant time $O(1)$ search. Like ORB-SLAM, we use the relative orientation between the last two frames to predict the current position of the system. The local map points are projected to the MCS and matched to the extracted features from the current frame. If enough matches are retained from the set of putative correspondences after an initial robust pose optimization using MultiCol, the tracking thread starts to search for more matches, assigns the reference MKF and decides if a new MKF should be added and passed over to the mapping thread. If the initial pose estimation fails, GP3P (Kneip et al., 2013) and RANSAC are used to perform re-localization of the MKS using the map points assigned to a set of recent MKFs. This is different compared to ORB-SLAM where a single camera and the non-minimal PnP solver EPnP (Lepetit et al., 2009) is used in a RANSAC loop to find potential map point matches and estimate the current camera pose after tracking failure. The tracking thread is detailed in Section 4.3.

- 为了快速匹配特征（储存在连续向量中）采用哈希搜索（线性复杂度）
- 解释MKF的重定位机制，跟踪线程分发匹配项给映射线程，映射线程参考最邻近一组的MKFs来进行重定位

Each time the tracking thread passes a new MKF to the mapping thread, recently created map points that do not fulfill certain conditions are deleted from the map. Then, new map points are triangulated over MKFs that are in the vicinity of the added MKF.

Here, the vicinity is determined by the co-visibility graph. In contrast to ORB-SLAM, where features are only triangulated between images of the same camera, the reconstruction is now performed over images of different cameras as well. Subsequently, local bundle adjustment is performed to adjust the poses of the MKFs that are part of the local map, as well as all map points. In addition, the mapping thread decides which MKFs are redundant and deletes them from the map. The mapping thread is detailed in Section 4.4.

- 进一步解释匹配点从MKFs中三角化而来(这里的triangulated指三角测量和三角化方法，而非三角剖分，类似orbslam中的算法，不过可以由多相机的MKFs实现，而orbslam只能在同一相机的图像下进行triangulated)

The loop closing thread searches for potential loop closures with every new MKF that is added. To decide if a place was already visited before and to identify MKFs as loop candidates, the system uses a Bag-of-Binary-Words framework (Gálvez-López and Tardós, 2012). If a loop is detected, the essential graph (a sparse version of the co-visibility graph) is optimized. To correct for the scale-drift, the optimization is carried out over similarity transformations that connect the MKFs.

Like ORB-SLAM, we use the graph optimization framework g2o (Kummerle et al., 2011) for all optimizations. The difference to ORB-SLAM is how we model the tracking and mapping pipeline. As multiple cameras observe the scene from each pose (Figure 1) and also one map point can be observed by multiple cameras at the same time, the graph can not be represented by binary edges anymore (edges that connect two vertexes). Instead, we extend the graph to a hyper graph that we used to model MultiCol (cf. Figure 4) where edges can connect to an arbitrary number of vertexes.

- 介绍Bag-of-Binary-Words framework（视觉词袋）进行回环检测及利用g2o优化
  参考: [ORB-SLAM2] 回环&DBoW视觉词袋

4.1 The MultiCol Model

In this short subsection, we will briefly recapitulate the MultiCol model. For a more in-depth introduction to MultiCol and the involved camera model the reader is referred to (Urban et al., 2015, Urban et al., 2016b).

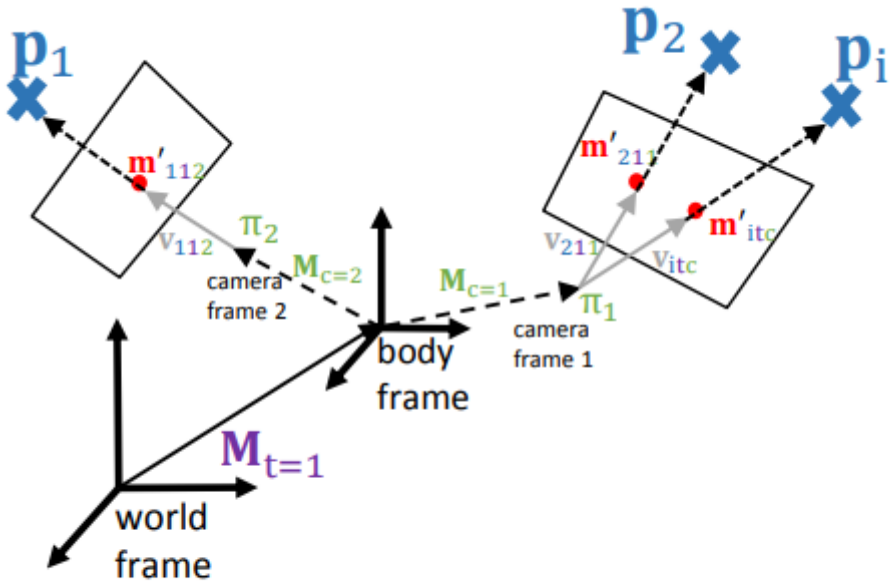


Figure 3. Depicted is the body frame concept and all involved parameters.

Given a homogeneous transformation matrix $\mathbf{M}$ the transformation and projection of a world point $\mathbf{p}_i$ to its corresponding image point $\mathbf{m}_{itc}$ in camera c at time t is given by:

$$\boxed{\mathbf{m}_{itc} = \pi_c^g(\mathbf{p}_{itc}) = \pi_c^g(\mathbf{M}_c^{-1}\mathbf{M}_t^{-1}\mathbf{p}_i)} \quad (1)$$

where π_c^g is the generic camera model presented in (Scaramuzza et al., 2006, Urban et al., 2015) modeling all types of central cameras. If only perspective cameras are used, this could be exchanged with the calibration matrix $\mathbf{K}$ and some additional distortion coefficients.

- 该部分介绍MultiCol Model
- 不同坐标系的几何关系图
- 在该图下由变换矩阵M_c和M_t求解观测的p到world frame下的m_{itc}
- generic camera model 定义参考

4.2 Map Entities

Map Points The map point is the most basic entity of the framework. Each map point $\mathbf{p}_i$ has the following attributes, properties and variables:

- The 3D position $\mathbf{p}_i = [X_i, Y_i, Z_i]^T$ in world coordinates.
- A maximum d_{max} and minimum d_{min} distance at which the point can be observed. This distance is used to reduce the number of points that are queried for local map tracking.
- The viewing direction $\mathbf{n}_i = [n_x, n_y, n_z]^T$.

- 定义map points 的坐标和range，以及观测方向n_i

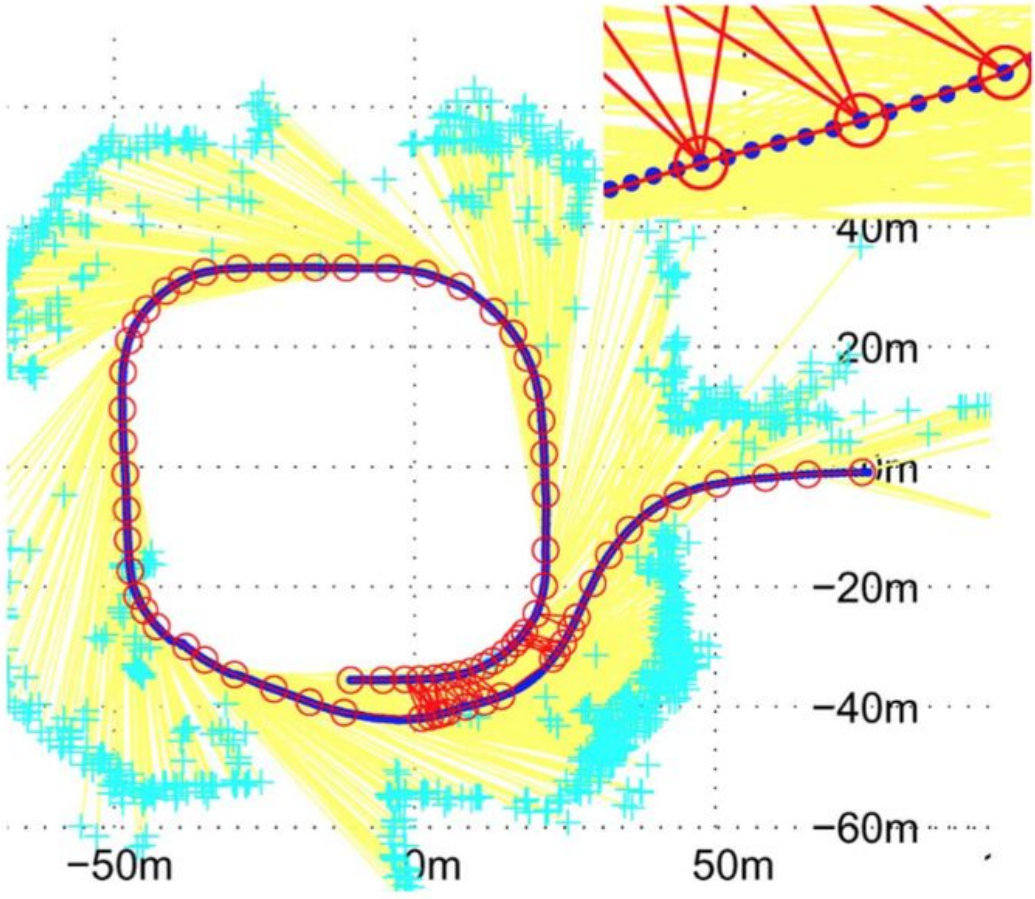
Multi-Keyframes In contrast to ORB-SLAM, each keyframe stores multiple images and is thus called Multi-Keyframe. Again, each Multi-Keyframe MKF_t created at time t has a number of attributes and variables:

- The MCS pose $\mathbf{M}_t$.
- A MCS object that stores the intrinsics of each involved camera and the extrinsics ($\mathbf{M}_c$) of the MCS. This object also performs the forward and back projection of world and image points.
- All features that are extracted from each camera. The features are stored in continuous vectors and thus a fast feature to camera search is needed. For each image point, we store two representations. One is its 2D image coordinate $\mathbf{m}'$ that we use extensively in MultiCol bundle adjustment and pose estimation. In addition, we store the corresponding 3D bearing vector $\mathbf{v}$. Latter will be used in various geometry related algorithms, e.g. essential matrix estimation, epipolar search and absolute pose estimation (GP3P).
- The Bag-of-Binary-Words representation.

- MKF和相关属性的定义

Co-Visibility Graph As in ORB-SLAM, the co-visibility graph is represented as a weighted undirected acyclic graph. The weight χ of each edge that connects two nodes (MKFs) in the graph is calculated as the number of map points the two MKFs share. In (Strasdat et al., 2011), a minimum weight χ_{min} between 15 and 30 was used to insert a connection. In contrast, the authors of ORB-SLAM do not impose a constraint on the minimal weight. When needed, the co-visibility graph is queried with a threshold, to only return nodes with a weight larger than χ_{min}, but the connectivity is kept very dense.

- 基于无向无环图的co-visibility graph，相比于orbslam，该框架中增加了最小约束 χ_{min}
- 根据调研，这里的Covisilibilty Graph的顶点是相机的Pose，而边是Pose-Pose的变换关系
  ，当两个相机看到相似的空间点时，它们对应的Pose就会产生联系（我们就可以根据这些空间点在照片上的投影计算两个相机间的运动）。根据观测到的空间点的数量，给这个边加上一个权值，度量这个边的可信程度。这个概念最早来自2010的文章 *Closing Loops Without Places*(参考回答)



Map Initialization The initialization of the system comprises the estimation of an initial map and the corresponding camera poses. In general, the initialization of the map is the first crucial step in camera based SLAM system. The accuracy and robustness of the initial reconstruction has a significant impact on the overall performance of the system.

The authors of ORB-SLAM introduce an algorithm that switches between scene reconstruction using the fundamental matrix $\mathbf{F}$ or estimating a homography $\mathbf{H}$ using a heuristic. In general, this is a challenging task and often ignored, assuming the first camera motion introduces enough parallax. If a homography describes the current scene better, i.e. if the camera is only rotating or observes a completely planar scene, initialization is suppressed. If a proper fundamental matrix is found, the scene is initialized, by reconstructing camera poses and scene points, followed by bundle adjustment eliminating the gauge freedom by fixing the first camera.

With MultiCol-SLAM, two issues arise that require some adaption. On the one hand, both $\mathbf{F}$ and $\mathbf{H}$ matrices contain the perspective camera matrix $\mathbf{K}$. For omnidirectional or fisheye cameras and especially the camera model employed in this work, a camera matrix does not exist. Still, if images points would be undistorted to their corresponding location in a perspective image, an application would be possible, although points at the image border would be lost. On the other hand, we are actually looking for the relative motion between two MCSs. Thus a map has to be initialized for each camera separately and then fused somehow. A different approach is to directly estimate the relative orientation between two MCS poses which is equivalent to computing the relative pose between two generalized cameras (Yu and McMillan, 2004). We experimented with the linear 17-pt (Yu and McMillan, 2004), a 6-pt (Stewénius et al., 2005) and a 8-pt algorithm (Kneip and Li, 2014) all part of OpenGV. The two polynomial solvers are relatively slow and the linear 17-pt algorithms is numerically very unstable. Recently, a new method was published (Ventura et al., 2015) but we leave the investigation for future work.

- 描述了地图初始化的过程，并根据MultiCol中的多相机模型改进了orbslam：先初始化每个相机而后融合

As the initial reconstruction of the SLAM trajectory is not at the core of this work, we propose a rather practical than generic methodology. We estimate an essential matrix $\mathbf{E}$ in a RANSAC loop between the same camera from different MCS poses and choose the one with the most inliers and the highest translational magnitude. Then, we exploit a slight overlap between the FoVs and search for the map point projection in all other cameras. Finally, we perform bundle adjustment over all observations and the two MCS poses. This routine, however, only works robustly if small overlap exists.

- 该框架中提出的一种实用思路进行trajectory的初始化

4.3 Tracking Thread

In this section, the tracking thread is detailed. It is the core of our multi-camera tracking system as it handles not only the current state but performs feature extraction, matching and pose estimation. At the same time, it is the only thread that has to run in real-time, i.e. at frame rate. If this is not the case, incoming camera images will be dropped and tracking will suffer. Thus, an efficient implementation of all methods is essential. In addition, the tracking thread handles the MKF insertion and takes care of distributing work to all other threads.

All optimizations are carried out using iterative re-weighted least squares (IRLS) using Levenberg-Marquardt regularization and a robust Huber kernel. Huber suggested to calculate the tuning constant e as $e = 1.345\sigma$, where σ is some estimate of the standard deviation of the residuals, that we set to $\sigma = 2$. After each optimization, outlier edges (measurements) are found and eliminated by testing the residual against the Huber value.

- 详细介绍了跟踪线程（tracking thread）。用于进行特征提取、匹配、姿态估计，处理MKF的插入并分配工作给其他线程
- 优化的正则化方法和损失函数

Feature extraction The standard ORB detector extracts FAST corners at multiple scale levels (usually 8) and retains a certain number of corners per level that fulfill the Harris cornerness measure. In ORB-SLAM each image is additionally divided into several cells on each pyramid level. Then, the extractor tries to find at least 5 corners per cell to ensure a homogeneous distribution of feature points in the image. If this is not the case the cornerness threshold is adapted. Finally, the feature orientation and a ORB descriptor is computed.

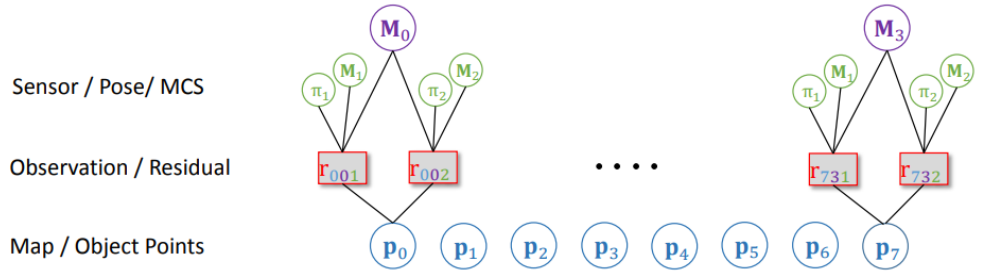


Figure 4. Depicted is the hyper graph model of MultiCol. Parameters are denoted as circles (vertices) and measurements as boxes (edges). In such a hyper graph edges can be connected to multiple vertices. In this example the $i = 1, .., 7$ map points p_i are observed by a MCS at a particular time from pose $\mathbf{M}_t$, with $t = 1..3$. The MCS consists of c=1..2 cameras that have a relative orientation $\mathbf{M}_c$ w.r.t the body frame and an interior orientation π_c.

- 特征金字塔进行特征提取。8个尺度，每个cell上提取5个corner，并计算orb描述子
- 图中展示了不同参数概念之间的函数关系

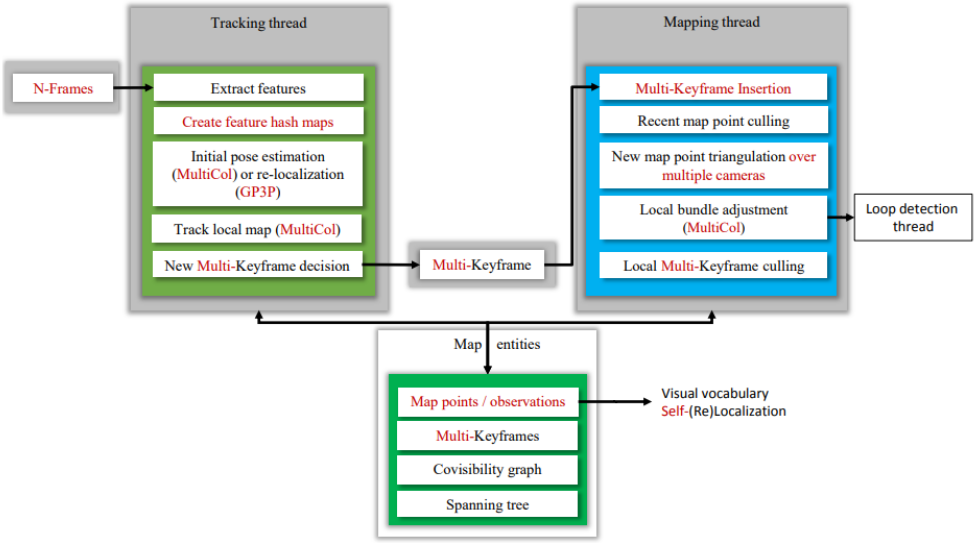


Figure 5. Depicts the MultiCol-SLAM framework without the loop closing thread. Red text depicts the modules where significant difference between our method and ORB-SLAM exist.

- 没有回环的MultiCol框架，红色的是和orbslam有明显区别的部分

As FAST corners usually need a re-training in new environments, we chose to utilize AGAST (Mair et al., 2010) corners instead. Note that, in theory, none of the efficient corner extractors is suited for highly distorted (fisheye) images. To achieve a higher repeatability, the pixel positions of the Bresenham circles used for intensity testing would have to be interpolated, hence completely destroying the efficiency. Some detectors for omnidirectional images were presented in (Hansen et al., 2008, Lourenço et al., 2012). These methods report high repeatability scores, however, are to slow for real-time applications. An interesting adaption of FAST and ORB to spherical images was proposed in (Zhao et al., 2015) called SPHORB, however, without providing the corresponding source code.

In MultiCol-SLAM, we extract ORB descriptors on each AGAST feature that is extracted in different cells over multiple scale levels keeping in mind that neither the detector nor the descriptor is particularly suited for distorted images. It shows, however, that the robust tracking and mapping back-end as well as the restriction of feature matching to local image areas (guided search) is able to compensate for the drawbacks and and weaknesses of the feature extraction stage.

- 用AGAST取代FAST角点，提供计算orb描述子的先验条件。该方法不适用于失真的图像，但一定程度上弥补了FAST的不足

Tracking from the Previous Pose This step is similar to ORB-SLAM. First, the current pose is estimated by using a constant velocity motion model. Then, local map points assigned to the last pose are projected to each camera of the current MCS and a guided search is performed around the projected location. With this initial set of matches, the MCS pose is optimized using MultiCol on fixed map points and outlier measurements are marked by identifying edges with residuals over e. With the optimized pose, the guided search is repeated, to identify more potential matches and the pose is optimized again.

- 该部分和orbslam同理

Re-Localization As soon as the tracking thread indicates a tracking failure, re-localization is carried out. This happens if not enough points are retained after the initial pose tracking. Then, the images are converted into their corresponding Bag-of-Words representation, and the recognition database is queried for potential MKF candidates. We iterate through each MKF and match all associated map points to the keypoints detected in the current frame yielding a set of putative correspondences. In ORB-SLAM, the initial pose estimate is found using EPnP in a RANSAC loop.

- 通过查询词袋和遍历MKF进行重定位

We exchange EPnP for two reasons. On the one hand, EPnP requires more than three points to estimate the current pose of the camera and is thus not a minimal solution. The EPnP estimate is furthermore rather unstable for only few points (Urban et al., 2016a). On the other hand, as we build our system on a MCS, we try to solve for the six degrees of freedom of the MCS pose using observations from multiple cameras. Thus, GP3P+RANSAC (Kneip et al., 2013) is used to find a putative set of inliers for all MKF candidates. If enough inliers are retained and RANSAC did not exceed a predefined number of iterations, we refine the initial pose estimate obtained by GP3P for each MKF over all inliers using UPnP (Kneip et al., 2013). Finally, the pose is optimized with MultiCol, again suppressing map point correspondences with high residuals. If more than a predefined number of points (we set this to 15) is retained after final pose optimization, re-localization was successful and the tracking thread goes back into its usual behaviour.

- 解释了重定位算法修改的原因（orbslam中使用的EPnP）

Tracking the Local Map The final step of the tracking loop is important, as it reinforces the visual connectivity and builds a densely connected co-visibility graph. Having estimated an initial pose either from tracking the previous pose or after re-localization, the local map is projected to the MCS to find more matches. To identify which map points are contained in the local map, a reference MKF to the current pose has to be found first. Thus, we take the list of map points that are currently assigned to the MCS pose entity and from which the current pose was estimated. As each map point stores a list of MKFs it has been observed in, we can then iterate through all map points and count their occurrences. Care has to be taken, as each map point can be observed multiple times from each MKF. The MKF with the most occurrences is taken as the reference MKF and a set of local MKFs is queried from the co-visibility graph.

- 通过匹配出现最多的MKF实现tracking的最后一步。

Then, the following steps are performed consecutively over each point $\mathbf{p}_i$ in the local map:

(1) Project $\mathbf{p}_i$ into each camera of the MKF. Discard if projection is not inside the mirror boundary of a specific camera. Otherwise add the point to potential candidates.

(2) Compute the angle between the current bearing vector $\mathbf{v}_i$ and the map point viewing direction $\mathbf{n}_i$ and discard if angle is larger than $50°$.

(3) Compute the distance d_i from the current MCS pose to the map point. If this distance is outside the interval d_{min} and d_{max} that are defined by the scale invariance region of the image pyramid, discard the point.

(4) Get all descriptors around the projected location of the map point at a given scale and match them to the map point descriptor.

Finally, the pose is optimized for the last time. If not enough matches are retained, the tracking thread goes into re-localization mode.

- 接下来用4.2中定义的变量来筛选（丢弃）p_i

New Multi-Keyframe Decision From a robustly estimated MKF pose that is tightly connected to the local map and co-visible MKFs the tracking thread decides if it is time to add a new MKF to the map. The insertion takes place if the following conditions are met. All thresholds are set according to the FPS of the camera system (our MCS runs at FPS $= 25$):

(1) More than $0.5 \cdot$ FPS frames have passed from last MKF insertion and the local mapping thread is idle.

(2) A certain amount of poses must be successfully tracked from the last re-localization. In our case this threshold is set to the current frame rate.

(3) At least 50 points are tracked from the current MCS pose.

(4) Less than 90% of the current map points are assigned to the reference MKF. Thus, MKFs are only inserted if the visual change is big enough.

- 插入新MKF的规则

4.4 Mapping Thread

This section details the mapping thread (cf. Figure 5). Asynchronously to the tracking thread, it extends the map by triangulating new points, performs local bundle adjustment and deletes redundant map points and MKFs from the map. Every time it finishes one loop, the entities are fed back to the tracking thread and become available.

- 映射线程（mapping thread）主要进行三角测量和map points && MKF的删除

Multi-Keyframe Insertion As soon as the tracking thread decides to insert a new MKF into the map, the mapping thread starts to update the co-visibility graph. The last step in the tracking thread ensured a tight connectivity. Thus a new node is added to the graph and the edge weights are updated with the number of map points each MKF shares with the new MKF. In addition, a Bag-of-Words representation of the new MKF is computed and saved in the recognition database.

- tracking thread决定插入新的MKF时mapping thread会更新co-visibility graph；还会计算新MKF的词袋并储存在数据库中

Recent Map Point Deletion The mapping threads stores a list of recently added map points from the last three MKFs. All map points under consideration have to pass the following conditions to remain in the map. The conditions do not only take visibility in subsequent MKFs into account but also the appearance between them.

(1) A map point has to be found in at least 25% of its predicted MCS poses.

(2) The map point must be observed from at least three MKFs.

As in ORB-SLAM, a map point can only be removed from the map if it is visible in less than three MKFs. This can happen if MKFs are removed from the map.

- 三角化计算map points的详细思路

Map point fusion As the triangulation step might have yielded redundant points that were already in the map, the next step is to fuse those point duplications. Therefore, the map points connected to the current MKF are first projected to all MKFs that are connected in the co-visibility graph. Within each MKF the map points are projected to each camera. Then, all features in a local area around the projected point are queried and matched. If a match is found, that also lies on the epipolar great circle, the map points are fused. If the matched image point is not connected to a map point yet, it is added as an observation. After projecting from the current to all connected MKFs, the same procedure is carried out vice versa.

- 而后进行map points的融合

Local Bundle Adjustment The local bundle adjustment optimizes over poses and map points in the inner window (cf. Figure 2). Therefore, we query the co-visibility graph from the current reference MKF to get a set of MKFs. Then, all map points that are seen from this set of MKFs are added to the local map. In addition, the outer window is found, by looping over the current set of local map points and identifying MKFs that are not yet part of the local set of MKFs. This stabilizes the trajectory and connects it to the rest of the map.

Again the MultiCol equations are used, but this time we optimize over the local map points $\mathbf{p}_i$ and poses $\mathbf{M}_t$. Although hundreds of points and dozens of poses are subject to optimization, the bundle adjustment problem can be efficiently solved by exploiting the special sparsity structure of the Jacobian and Hessian respectively. The special structure comes from the fact that only point-pose but no point-point or pose-pose constrains exist. The resulting normal equations can the be efficiently solved using the Schur complement trick. For more details and insights on the subject, the reader is referred to (Engels et al., 2006). Subsequently, the local map is updated and passed back to the tracking thread.

- 执行局部BA

Local Multi-Keyframe Culling Like in ORB-SLAM, we delete a MKF from the map if any pair of MKFs shares more than 90% of the same map points. Instead of counting map point observations for each camera of the MCS, we count only the occurrence per MKF.

- 通过统计MKF的出现次数进行局部关键帧剔除

4.5 Loop Detection and Closing Thread

In this section, the loop detection and correction procedures are detailed. They are similar to the methodology proposed in (Mur-Artal and Tardos, 2014) but extended and adapted to multi-fisheye camera systems. As measurement errors accumulate over time, the estimated trajectory starts to drift in seven degrees of freedom, i.e. translation, rotation and scale. This effect is visualized in a toy example in Figure 6. Although the MCS visits the same place, the current local map (depicted in blue) does not coincide with the historic map (depicted in gray and orange). Although the map does not spatially align with the start of the trajectory, its local map structure is very similar assuming that the system reconstructs a large amount of map points at the same location, i.e. the reconstruction is repeatable. The goal of the loop detection thread is to identify the situation depicted in Figure 6. It detects the loop candidates from the historic trajectory and corrects the loop by propagating the loop closing error along the trajectory.

The loop detection and correction procedure, as well as the place recognition database and visual vocabulary components are depicted in Figure 7. Each step is detailed in the following sections.

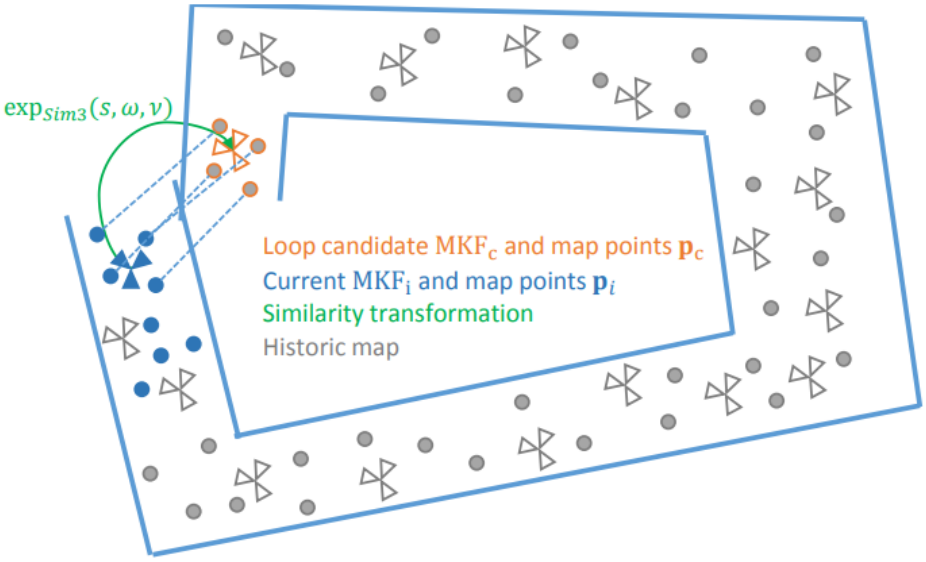
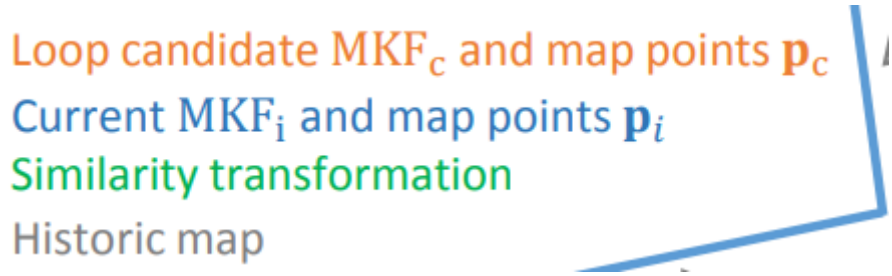


Figure 6. Depicts the loop closing problem. If the SLAM trajectory was estimated without drift, the orange and blue map points should coincide. As this is in general not the case, a similarity transformation can be estimated that aligns both parts of the trajectory over the map points. Then, the alignment error can be used to correct the remaining MKF poses and map points by projecting it back through the map.

- 介绍回环检测和累计误差的修正
- 闭环线程（closing thread）用于检测图中的下列情况



Candidate detection As soon as the system revisits a place or parts of a scene it has seen and reconstructed before, it should load the associated local map points and start tracking from them instead of starting to reconstruct the scene again. This, however, is a challenging task, as the local map is solely queried from the co-visibility graph. One possible way would be to query the map points spatially, i.e. that we query the nearest neighbor map points in a specified radius from the current MKF. This method could work for smaller loops, but as soon as larger loops occur, the corresponding local maps could lie dozens of meters apart, which is sketched in Figure 6. A more favorable solution is to use visual cues to identify possible loop candidates.

- 候选检测（candidate detection）指如果检测为重新访问的场景，则加载关联的局部地图进行tracking而不是再次重建
- 实现该算法有两种思路：基于邻域半径搜索MKF和基于视觉信息。显然视觉信息更好。这一步在局部BA之后进行

somehow. This could either be achieved by taking a fixed number of MKFs sorted by their similarity score. Latter, however, is an absolute measure whose magnitude is unknown, e.g. we queried 10 MKFs, but all have a very bad similarity score. Thus a way of getting a relative measure of the quality of the current similarity score is needed.

Hence, the BoW vectors of all MKFs that are connected in the co-visibility graph of the current MKF$_i$ are queried. Then, the similarity score between all BoW vectors and the current MKF BoW vector are calculated. The lowest score s_{sim} is taken as a similarity threshold, i.e. we only take MKFs from the database as candidates if their score is higher than s_{sim} and thus more similar, than the most dissimilar MKF in the co-visibility graph.

To further reduce the number of possible false candidates, MKF candidates are only accepted if they are part of a consistent group of connected MKFs in the co-visibility graph after several consecutive MKF insertions. Each loop candidate is connected to a number of MKFs (a group) in the co-visibility graph. A group is accepted to be consistent with the previous group if it they share a MKF.

- 详细解释候选检测的思路。利用词袋产生的词向量进行相似度打分，来帮助co-visibility graph的MKF查询（基于相似度的查询）

Transformation estimation If one or more candidates are accepted, the similarity transformation between the current MKF_i and all candidates can be estimated. Lets assume for now, that we have one candidate MKF_c. Looking back at Figure 6, the goal is to find the similarity transformation $\mathbf{S}$ between the map points $\mathbf{p}^i$ assigned to MKF_i and the map points $\mathbf{p}^c$ assigned to MKF_c:

$$\mathbf{p}^i_{mcs} = \mathbf{S}\mathbf{p}^c_{mkf} = \begin{bmatrix} s\mathbf{R} & \mathbf{T} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{p}^c_{mkf} \tag{2}$$

Instead of taking all map points that are connected to both frames, the descriptors assigned to each map point by the MKF are matched in advance. This leaves us with a subset of possible map point correspondences. Yet, this set contains outliers that are either caused by wrong descriptor matches or the distance ratios between reconstructed points is too big, caused by bad reconstruction. Hence, RANSAC is used to find a similarity transformation using Horn's quaternion ((Horn et al., 1988)) method as a model and thus 3D-3D [1] correspondences.

First the set of map points matches is transformed to the respective MKF:

$$\mathbf{p}^c_{mkf} = \mathbf{M}^c_t \mathbf{p}^c \quad \mathbf{p}^i_{mkf} = \mathbf{M}^i_t \mathbf{p}^i \tag{3}$$

where $\mathbf{M}_t$ is the respective pose of the MKF. Obviously, the points can not be transformed to each camera of the MKF, as map points can be observed from multiple cameras and the only common frame is the MKF frame.

Subsequently RANSAC iterations are performed. Three points are selected from each point cloud, and the transformation is estimated. To decide whether the transformation is accepted, the map points are transformed from MKF_c to MKF_i and vice versa using the estimated similarity:

$$\hat{\mathbf{p}}^i_{mkf} = \mathbf{S}\mathbf{p}^c_{mkf} \quad \hat{\mathbf{p}}^c_{mkf} = \mathbf{S}^{-1}\mathbf{p}^i_{mkf} \tag{4}$$

- RANSAC是通过反复选择数据集去估计出模型，一直迭代到估计出认为比较好的模型。
  具体的实现步骤可以分为以下几步：选择出可以估计出模型的最小数据集；(对于直线拟合来说就是两个点，对于计算Homography矩阵就是4个点)使用这个数据集来计算出数据模型；将所有数据带入这个模型，计算出“内点”的数目；(累加在一定误差范围内的适合当前迭代推出模型的数据)比较当前模型和之前推出的最好的模型的“内

点“的数量，记录最大“内点”数的模型参数和“内点”数；重复1-4步，直到迭代结束或者当前模型已经足够好了(“内点数目大于一定数量”)。参考

- RANSAC具体使用一种和四元数相关的方法，参考Horn’ s quaternion ((Horn et al., 1988))
- 在本文框架中值得注意的是transformation estimation满足多相机模型，因此依据（2）中的公式以多相机MKFs得到的map points来近似出MCS p_i之间的transformation S

Subsequently the points are transformed to the camera frames and projected to the image plane. Now, the reprojection error can be computed and used to determine the number of inliers. If the transformation yields enough inliers, a guided matching is instantiated to search for more correspondences, also between cameras. Then, $\mathbf{S}$ is optimized by minimizing the reprojection errors of both transformed point sets (Equation 4) in both MKFs. The optimization is again carried out using g2o and outliers are down-weighted using a Huber kernel. If more than 20 inliers are retained after optimization, $\mathbf{S}$ is accepted and the loop correction is started.

- 在RANSAC迭代结束后，进一步进行最小化重投影误差+g2o的优化

Loop correction and fusion The first step to loop correction is, to correct all MKFs that are connected to the current MKF_i, as well as all map points that are part of the local map. After this step, the local maps spanned by MKF_i and MKF_c should align. The corrected pose $\hat{\mathbf{M}}_t$ of a MKF is computed by first estimating the relative orientation between pose $\mathbf{M}_t^i$ of the current MKF_i and the MKF pose $\mathbf{M}_t$ and subsequent correction using the similarity:

$$\hat{\mathbf{M}}_t = (\mathbf{M}_t \mathbf{M}_t^{i^{-1}})\mathbf{S} \tag{5}$$

Subsequently, the map points need to be corrected as well. First, each map point is rotated to the MKF frame using the uncorrected pose. Then, the point is transformed to the corrected map point position $\hat{\mathbf{p}}$ by applying the inverse of the corrected MKF pose, i.e. the map point is directly transformed back into world coordinates:

$$\hat{\mathbf{p}} = \hat{\mathbf{M}}_t^{-1}(\mathbf{M}_t \mathbf{p}) \tag{6}$$

The correction of the local map will result in many point duplications and redundant MKFs. Thus the same map point fusion procedure presented in Section 4.4 is carried out and the co-visibility graph is updates.

- 这一节详细介绍回环的修正与融合的设计，总是以当前的MKF和map points修正既往的MKFs
- 对于每一个MKF的pose，以第一次和当前新的测量按照公式（5）计算修正量$\hat{M}_t$，同时以（6）来修正map points 得到修正量p

Finally, the essential graph is optimized. First, all MKF poses $\mathbf{M}_t$ are converted to similarities $\mathbf{S}_t$ by initially setting the scale to 1.0. Then, the relative pose constrains between all MKFs in the map are computed:

$$\Delta\mathbf{M}_{ij} = \mathbf{M}_j\mathbf{M}_i^{-1} \tag{7}$$

between some MKF i and j. Again, all relative pose constrains are converted to similarities $\Delta\mathbf{S}_{ij}$. The optimization is carried out over pose-pose constrains and follows the work of (Strasdat et al., 2010). The residual that we try to minimize is defined as:

$$\mathbf{r}_{i,j} = \log_{Sim(3)}(\Delta\mathbf{S}_{ij}\mathbf{S}_i\mathbf{S}_j^{-1}) \tag{8}$$

where the log is the inverse relation of the exponential map (see (Strasdat, 2012)). The goal of the optimization is to adjust $\mathbf{S}_i$ and $\mathbf{S}_j$ such that the transformation sequence (back and forth between both MKFs) is as close to the identity as possible. In the beginning, all residual transformations Equation 8 will be the identity except for the part of the map, that was corrected above. Then the error is propagated back over all pose-pose constrains during optimization as the similarities $\mathbf{S}_i$ and $\mathbf{S}_j$ are gradually changed to optimally fit the loop closure constrain.

- 这样的矫正会导致多余点和重复点的出现，因此设计算法进行删除
- 两个MKF之间以公式（7）进行融合得到M_{ij}
- 利用relative pose constraints得到相似性S_t，以（8）为优化函数进行优化，该过程参考文献(Strasdat, 2012)

5. EXPERIMENTS AND RESULTS

To test the performance of the presented MCS SLAM, various tests are performed. More information about the dataset can be found on (Urban and Jutzi, 2016). First, we evaluate the impact of using multiple fisheye cameras instead of one, in terms of accuracy, runtime, successfully tracked frames and loop closing.

To evaluate the accuracy of SLAM systems, two metrics are commonly used that compare the estimated camera poses $\mathbf{M}_t$ to a ground truth pose $\mathbf{M}_t^{gt}$ at some time t or an interval Δt. The difference between these two poses at time t is given by the relative orientation between them:

$$\mathbf{M}_t^{rel} = \mathbf{M}_t^{gt\,-1}\mathbf{M}_t \tag{11}$$

The first metric is called ATE and estimates the root mean squared translation differences between both trajectories. In order to calculate the absolute error, the two trajectories need to be aligned in advance using a similarity transformation $\mathbf{S}$. For N pose pairs, the ATE can then be calculated as:

$$\text{ATE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N} \|\text{trans}(\mathbf{M}_t^{rel})\|^2} \tag{12}$$

$$= \sqrt{\frac{1}{N}\sum_{t=1}^{N} \|\text{trans}(\mathbf{M}_t^{gt\,-1}\mathbf{S}\mathbf{M}_t)\|^2} \tag{13}$$

where "trans" returns the translational component of the transformation matrix $\mathbf{M}$.

The second metric is called RPE and allows to evaluate the local accuracy and drift of the trajectory over some time interval Δ. Thus we can calculate $M = N - \Delta$ relative orientation errors along the trajectory. The RPE at time step t can be defined by:

$$\text{RPE}(\Delta) = \sqrt{\frac{1}{M}\sum_{t=1}^{M} \|\text{trans}(\mathbf{M}_t^{rel})\|^2} \tag{14}$$

but this time the relative transformation is defined as:

$$\mathbf{M}_t^{rel} = (\mathbf{M}_t^{gt\,-1}\mathbf{M}_{t+\Delta}^{gt})^{-1}(\mathbf{M}_t^{-1}\mathbf{M}_{t+\Delta}) \tag{15}$$

To calculate the relative error of subsequent poses we set $\Delta = 1$. In the case of ATE only the translation is evaluated. For the relative error, we can also evaluate the rotational accuracy. This is done by replacing the "trans" with a function that returns the Rodriguez vector of the rotation matrix in $\mathbf{M}$.

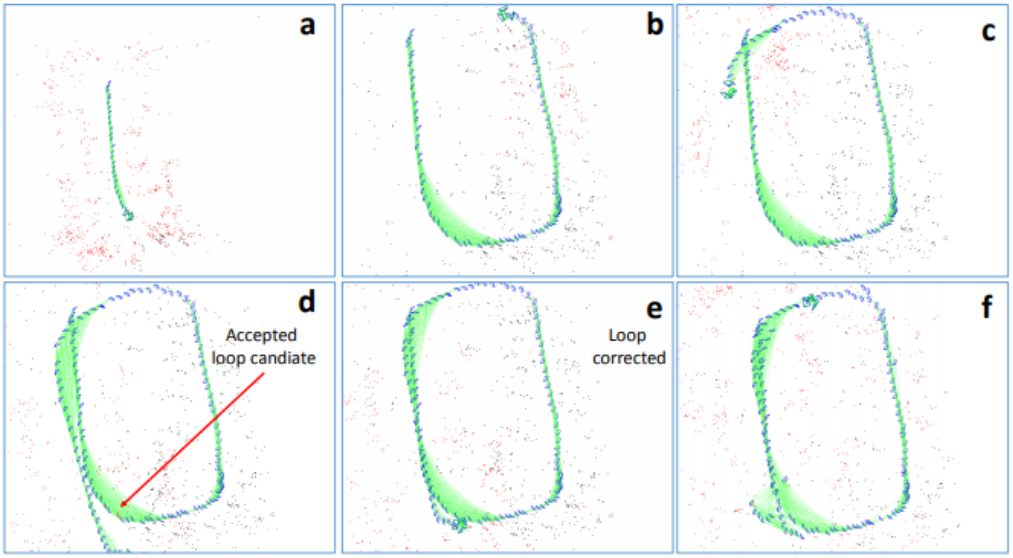
Each trajectory is evaluated five times, i.e. the SLAM algorithms are used five times to estimate the camera trajectory. All accuracies and run-times are calculated as the median value over the five runs.

- 该部分为MCS SLAM所做的实验
- （11）给出ground truth和时间t上位姿的相对误差计算
- 基于上述相对误差，以ATE和RPE为两种误差评估方式进行评估

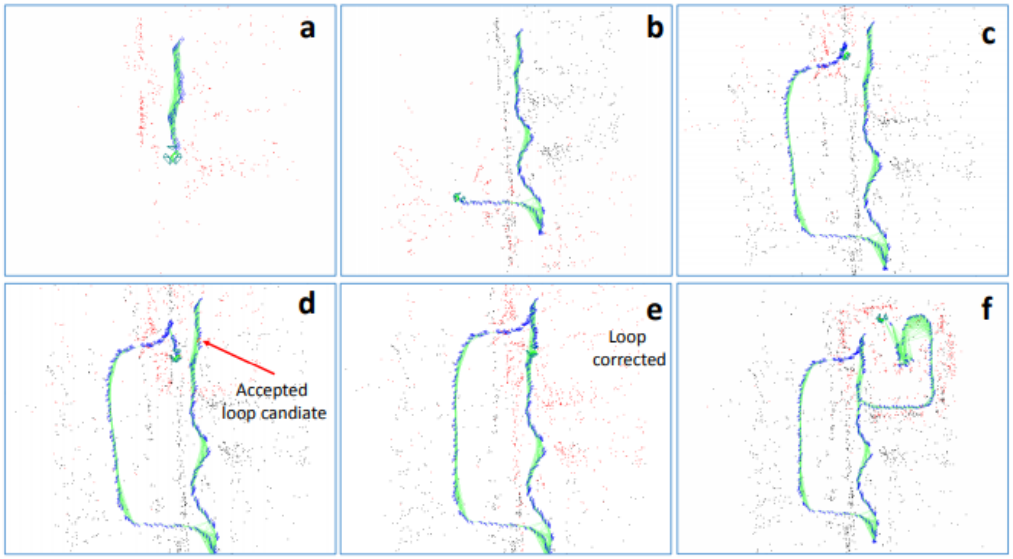
5.1 Single- vs. Multi-Camera SLAM

First, we align the KFs or MKFs respectively, by estimating a similarity transformation between ground truth and SLAM trajectory. Then, the ATE (Equation 13) is evaluated for all trajectories. The results are depicted in Table 9a. Obviously, MultiCol-SLAM significantly outperforms its single camera pendant in terms of Keyframe accuracy. One explanation of the large performance gap is the simple initialization of the single camera SLAM. The authors of ORB-SLAM proposed an initialization based on homography and fundamental matrix estimation. Both matrices can not be readily computed for the camera model employed in this work. Thus we simply initialize the single camera SLAM by estimating the essential matrix and selecting a solution based on a threshold on the magnitude of the translation vector, which is obviously less robust than the method proposed in (Mur-Artal and Tardos, 2014).

To get a measure of the local accuracy, we also estimate the RPE (Equation 14) for all trajectories and all poses by setting $\Delta = 1$. The trajectories do not need to be aligned in this case. The accuracies for translation and rotation are depicted in Table 9b. Still, using multi-cameras yields a better performance, especially for the translation. The rotational components show a similar trend, but the differences are less prominent. The rotational accuracy for the two lasertracker trajectories is a lot better because the trajectory has only little rotation of the camera system about the up-axis. On average the rotational accuracy of the MCS is about 0.5-1.5° and the translational component, depending on the walking speed and scene between 1.0-2.5 cm.



(a) Trajectory 1



(b) Trajectory 2

	Single fisheye camera	Multi-fisheye camera system
	[cm]	[cm]
Laser 1	31.0	1.4
Laser 2 fast	28.1	5.3
Indoor 1 stat. env.	32.4	2.1
Indoor 2 dyn. env.	13.3	1.8
Outdoor 1 dyn. env	(X)	3.6

(a) **ATE**

Single fisheye camera	Multi-fisheye camera system
[cm]/[deg]	[cm]/[deg]
1.95/0.32	1.2/0.33
2.6/0.31	2.7/0.56
2.8/1.72	1.1/1.54
2.8/2.04	1.1/1.78
(X)	2.3/1.28

(b) **RPE**

Figure 9. (a) Median KF and MKF **ATE**s for single and multi-camera SLAM respectively. The translational accuracy was calculated after 7DoF alignment between ground truth and estimated frames. (b) **RPE** for single and multi-fisheye camera SLAM. Here we set $\Delta = 1$, i.e. the frame two frame tracking accuracy is estimated. (X) means that tracking failed at some point and a significant part of the trajectory was not tracked.

Thread	Operation	Single fisheye camera	Multi-fisheye camera system
Tracking	Frame creation	12/12/2	21/23/5
	Pose estimation	4/4/1	3/4/3
	Track local map	5/5/2	4/5/4
	Total	**21**	**28**
Mapping	Map point creation	27/26/10	96/96/17
	Map point fusion	11/11/6	11/12/9
	Local BA	185/198/125	170/174/65
	Total	**223**	**277**

Table 1. Depicted is the time (mean, median and standard deviation) for each step in different threads. In case of Single camera SLAM, we extracted 1000 features. In case of multi-camera SLAM 400 features per camera are extracted and the extraction is performed in parallel. The pose estimation is slower in the single camera case, as no analytical Jacobian was provided.

- 将多相机的框架和单目做了对比，包括轨迹和精度

6. SUMMARY AND CONCLUSION

In this paper MultiCol-SLAM, a real-time multi-fisheye camera SLAM system was proposed. First, we recapitulated the current state-of-the-art in the field and argued why keyframe-based approaches outperform filter-based SLAM systems. Then we subsumed the MultiCol model and detailed our contributions. Subsequently, we elaborately detailed our framework that builds upon ORB-SLAM and is divided into several threads running in parallel. Finally, all proposed modules were examined using accurate ground-truth data and it showed, that using multi-camera systems helps to improve the accuracy and robustness of SLAM in challenging environments.

In addition, we make the proposed SLAM system available to the public (`https://github.com/urbste/MultiCol-SLAM`) and hope that it helps to further encourage research in multi-camera egomotion estimation and related topics.

- 实验表明的结论：使用多摄像机系统有助于提高SLAM在挑战性环境中的准确性和鲁棒性