CamOdoCal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry

Conference Paper *in* Proceedings of the ... IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE/RSJ International Conference on Intelligent Robots and Systems · November 2013

DOI: 10.1109/IROS.2013.6696592

3 authors, including:

Lionel Heng
DSO National Laboratories
28 PUBLICATIONS **2,037** CITATIONS
SEE PROFILE

Marc Pollefeys
ETH Zurich
626 PUBLICATIONS **31,136** CITATIONS
SEE PROFILE

CamOdoCal: Automatic Intrinsic and Extrinsic Calibration of a Rig with Multiple Generic Cameras and Odometry

Lionel Heng, Bo Li, and Marc Pollefeys
Computer Vision and Geometry Lab, ETH Zürich, Switzerland

***Abstract*— Multiple cameras are increasingly prevalent on robotic and human-driven vehicles. These cameras come in a variety of wide-angle, fish-eye, and catadioptric models. Furthermore, wheel odometry is generally available on the vehicles on which the cameras are mounted. For robustness, vision applications tend to use wheel odometry as a strong prior for camera pose estimation, and in these cases, an accurate extrinsic calibration is required in addition to an accurate intrinsic calibration. To date, there is no known work on automatic intrinsic calibration of generic cameras, and more importantly, automatic extrinsic calibration of a rig with multiple generic cameras and odometry.**

We propose an easy-to-use automated pipeline that handles both intrinsic and extrinsic calibration; we do not assume that there are overlapping fields of view. At the begining, we run an intrinsic calibration for each generic camera. The intrinsic calibration is automatic and requires a chessboard. Subsequently, we run an extrinsic calibration which finds all camera-odometry transforms. The extrinsic calibration is unsupervised, uses natural features, and only requires the vehicle to be driven around for a short time. The intrinsic parameters are optimized in a final bundle adjustment step in the extrinsic calibration. In addition, the pipeline produces a globally-consistent sparse map of landmarks which can be used for visual localization. The pipeline is publicly available as a standalone C++ package.

I. INTRODUCTION

There has been an explosive growth in the use of cameras on robotic and human-driven vehicles. From the robotic perspective, a camera offers a rich source of visual information which greatly enhances robot perception in contrast to lidar line scanners; recent advances in computing hardware facilitate real-time image processing. From the automotive perspective, multiple cameras are useful for driver assistance applications which help improve road safety. Image processing applications utilizing multiple cameras on a vehicle require both an accurate intrinsic calibration for each camera and an accurate extrinsic calibration. An accurate intrinsic camera calibration consists of an optimal set of parameters for a camera projection model that relates 2D image points to 3D scene points; these optimal parameters correspond to minimal reprojection error. An accurate extrinsic calibration corresponds to accurate camera poses with respect to a reference frame on the vehicle, usually the odometry frame. Accurate calibration allows feature points from one camera to be reprojected into another camera with low reprojection errors, and furthermore, odometry data can be used in conjunction with the camera extrinsics to efficiently compute a good initial estimate of the camera poses which can then be refined via local bundle adjustment with minimal correction.

We use the unified projection model described by Mei et al. in [1] which works well in practice for regular, wide-angle, fish-eye, and catadioptric cameras. This model differs from the unified Taylor model proposed by Scaramuzza et al. [2] in the aspect that Mei's model is of a parametric form and explicitly models a generic camera while Scaramuzza's model is represented by an arbitrary Taylor polynomial. As a result, a closed-form Jacobian matrix can be formulated using Mei's model but not for Scaramuzza's model. Contrary to claims that Mei's model applied to fish-eye cameras has limited accuracy, we find from experimental data that Mei's model works very well for fish-eye cameras in practice.

Our pipeline has two stages: intrinsic and extrinsic calibration. In the intrinsic calibration stage, we require a large chessboard on which all squares have an equal known dimension. For each camera, with the chessboard held in a wide variety of poses, we use an automatic chessboard corner detector to find all interior corners on the chessboard in every image until we accumulate a minimum number of corners. We then use the set of estimated corner coordinates to find the parameters to Mei's model by generating an initial estimate of the intrinsic parameters and refining the parameters via non-linear optimization; an analytical Jacobian is used to significantly speed up the optimization.

In the extrinsic calibration stage, we separately run monocular VO with sliding window bundle adjustment for each camera. It is possible for the VO to break occasionally in poorly-textured areas. Nevertheless, we use all sets of VO estimates to find an initial estimate of the camera-odometry transform for each camera; each set of VO estimates has a different scale. We triangulate the inlier feature point correspondences generated by monocular VO using the initial camera-odometry transforms and odometry data. The resulting sparse map is then optimized via bundle adjustment; the odometry poses are kept fixed while all 3D scene points and the camera-odometry transforms are optimized. At this point, the camera-odometry transforms become more accurate; however, they are not sufficiently accurate for reprojection of feature points from one camera to another camera with sub-pixel reprojection errors. To solve this issue and still adhere to the assumption of no overlapping fields of views, we find feature point correspondences across different cameras. Starting from the first odometry pose, we maintain a local frame history for each camera, and we find feature point correspondences between each camera's current frame and every frame in every other camera's frame history. For each frame pair, we rectify the two images on a common image

plane which corresponds to the average rotation of the two corresponding cameras. This rectification greatly increases the number of point feature correspondences; we obtain a valid subset of feature point correspondences after geometric verification. For each feature point correspondence, we project the existing 3D scene points in the map and seen in the corresponding cameras onto the rectified images, and associate each feature point with the 3D scene point whose projected image point is nearest to that feature point and within a distance of 2 pixels.

To ensure the global consistency of the sparse map of landmarks, we use a vocabulary tree to detect loop closures using all frames from all cameras. We run another bundle adjustment similar to that described earlier, but which also optimizes all intrinsic camera parameters and odometry poses, to obtain a set of intrinsic parameters for each camera, a set of camera-odometry transforms, and a globally-consistent sparse map of landmarks.

Our approach is novel in the aspect that we are the first to develop a full automatic pipeline for both intrinsic calibration for a generic camera and extrinsic calibration for a rig with multiple generic cameras and odometry without the need for a global localization system such as GPS/INS and the Vicon motion capture system. We make the pipeline robust to breaks in monocular visual odometry which occur in areas with low texture.

A. Related Work

There has been much research on the hand-eye calibration problem [3], [4]. In general, published generic solutions to the hand-eye calibration problem do not work for vehicles with planar motions, as the height of the camera with respect to the odometry is unobservable. Hence, camera-odometry calibration requires specialized solutions [5], [6]. [5] finds the extrinsics, camera intrinsics, and odometry parameters simultaneously for a differential drive robot; however, a set of known landmarks is required. Similarly, [6] finds the extrinsics and odometry parameters for a differential drive robot without the need for known landmarks, but only obtain the three degrees of freedom of the camera-odometry transform. In contrast, our approach does not require known landmarks, and obtains the full six degrees of freedom of the camera-odometry transform for each camera. We do not attempt to calibrate the odometry parameters, as motion models greatly vary across vehicles.

Our approach is most similar to the works of [7] and [8]. Guo et al. [7] estimates the camera-odometry transform for one camera and an odometer by using a set of camera and odometry motions to find a least-squares solution. However, monocular VO has pose and scale drift which is exacerbated in environments where a majority of natural features are considerably far from the camera, and as a result, the translation component of the camera-odometry transform can be inaccurate. Furthermore, they do not use visual landmarks to further refine the camera-odometry transform. Our approach uses natural features in the vehicle's environment to improve the accuracy of the camera-odometry transform. Carrera et al. [8] proposes an extrinsic calibration of a multi-camera rig by using a modified version of MonoSLAM to build a globally consistent sparse map of landmarks for each camera, finding feature correspondences between each pair of maps via thresholded matching between SURF descriptors, and using the 3D similarity transform together with RANSAC to find inlier feature correspondences. At the end, a full bundle adjustment is run to optimize the camera poses, 3D scene points, and robot poses. Here, no attempt has been made to calibrate the camera poses with respect to the odometry frame, and the 3D similarity transform step can fail in outdoor environments where the majority of natural features are located far away from the cameras, and their estimated 3D locations can have substantial noise as a result, leading to few inliers. Furthermore, the calibration procedure requires the vehicle to make a full 360° turn such that near the end of the data collection, each camera observes the same features seen at the beginning; this requirement is to ensure that the 3D similarity transform step works by enforcing the global consistency of the maps. This requirement can be prohibitive for large vehicles in cluttered settings. In contrast, our extrinsic calibration works well without this requirement; we only require the vehicle to make a wide range of steering changes for optimal calibration accuracy. In addition, our extrinsic calibration optimizes the intrinsics; we find that if the intrinsics are not optimized as part of the bundle adjustment, the calibration accuracy is suboptimal.

II. Platform

Fig. 1: The two test car platforms, each equipped with a set of four fish-eye cameras that provides an all-surround view of the environment.

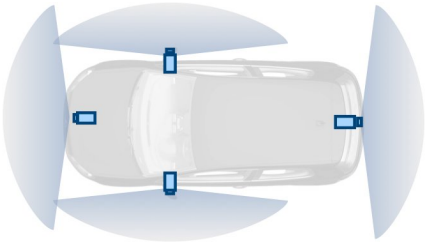


Fig. 2: The sensor coverage of the test car platforms. The four fish-eye cameras are marked in blue.

The platform is a VW Golf VI car modified for vision-guided autonomous driving under the auspices of the V-Charge project. Two of our platforms are shown in figure 1. Four fish-eye cameras are mounted on the car as shown

by blue cameras in figure 2. Each fish-eye camera has a nominal FOV of 185° and outputs 1280 × 800 images at 12.5 fps. For synchronous image capture, one camera acts as a master trigger source for all other cameras. A single PC timestamps the images together with the odometry data. The odometry data is not generated at the same time as the images; for each image, we find two odometry poses in a temporal odometry pose buffer, one whose timestamp is closest to and before the image timestamp, and one whose timestamp is closest to and after the image timestamp. We perform linear interpolation to get the odometry pose that corresponds to the image timestamp.

For purposes of clarity, in subsequent figures, we color all camera poses and 3D scene points associated to the front, left, rear, and right cameras as red, green, blue, and yellow respectively.

III. Intrinsic Calibration

We make some improvements to the automatic chessboard corner detection method of [9] in terms of speed and corner accuracy. In particular, after a set of corners is detected, we use a spline-based method to check the global consistency of the positions of the detected corners. We fit a spline through each row of corners; we choose the first, middle, and last corners as control points. The corner set is marked as invalid if the shortest distance between any corner and the corresponding spline exceeds a certain threshold. This check significantly reduces the false corner detection rate. The OpenCV chessboard corner detection implementation only works for images captured by normal and wide-angle cameras. An example of detected corners for a chessboard image is shown in figure 3.

Fig. 3: The detected corners of a chessboard.

We need to solve for the camera's intrinsic parameters $[\xi, k_1, k_2, k_3, k_4, \alpha, \gamma_1, \gamma_2, u_0, v_0]$ in addition to the camera poses. ξ models the mirror transformation, $[k_1, k_2, k_3, k_4]$ models the radial and tangential distortions, and $[\alpha, \gamma_1, \gamma_2, u_0, v_0]$ are the parameters for the standard pinhole model used for the generalized camera projection where α is the skew, γ_1, γ_2 are the focal lengths and $[u_0, v_0]$ is the principal point. Here, we assume all pixels to be square. For the initial estimate, we set $\xi = 1$, $k_1 = k_2 = k_3 = k_4 = \alpha = 0$, and (u_0, v_0) to be the image center coordinates. To find the value for the generalized focal length $\gamma = \gamma_1 = \gamma_2$, we iterate over each row of corners for each chessboard; in each iteration, we obtain an initial estimate of γ using the method in [1], estimate the camera pose from 2D-3D correspondences by rectifying the corners' image coordinates, and compute the reprojection error. We find the minimum reprojection error that corresponds to the best initial estimate of γ. We refine the initial estimate of the intrinsic parameters by using non-linear optimization to minimize the sum of reprojection errors over all images, and a robust loss function to minimize the outlier effect. For faster optimization, we derive and use an analytical Jacobian to reduce the optimization time by a factor of ∼100. For each camera, we typically obtain from 100 images an average reprojection error of between 0.2 and 0.4 pixels for the estimated intrinsic parameters.

IV. Extrinsic Calibration

A. Monocular VO

We run monocular VO for each camera in order to obtain a set of camera motions together which is required for the subsequent step of computing an initial estimate of the extrinsics. Our monocular VO steps are similar to those in [10] but for real-time performance, we use the five-point algorithm [11] instead of the eight-point algorithm, and instead of using three-view triangulation, we use linear triangulation to triangulate the feature points in the last two views, and check that the reprojection error of the resulting 3D scene point in the first view does not exceed a threshold which in our case is 3 pixels. We extract a synchronized set of keyframes from all cameras when the corresponding odometry pose is at least 0.25 m away from the odometry pose at which the last set of keyframes was taken. Subsequently, we use the OpenCV implementation of GPU SURF to extract feature points and their descriptors, use the distance ratio metric to match feature points, and find inlier feature point correspondences via geometric verification. For each inlier feature point correspondence which does not correspond to a previously initialized 3D scene point, we triangulate this correspondences; otherwise, we associate the already initialized 3D scene point to the correspondence. We find the current camera pose via the iterative form of PnP RANSAC with the initial estimate set to the rotation component of the previous camera pose multiplied by the rotation between previous and current camera poses estimated by the five-point algorithm during the geometric verification, and the translation of the previous camera pose. At each iteration of monocular VO, we use a sliding window bundle adjustment; the window contains 10 frames, and we keep the camera poses in the first 3 frames fixed. We show the inlier feature point tracks and the estimated camera poses for the four fish-eye cameras in figures 4 and 5 respectively.

B. Initial Estimate of Camera - Odometry Transform

We expand the method of [7] to robustly estimate each camera-odometry transform in feature-poor environments; it is common for the VO estimation to break occasionally,

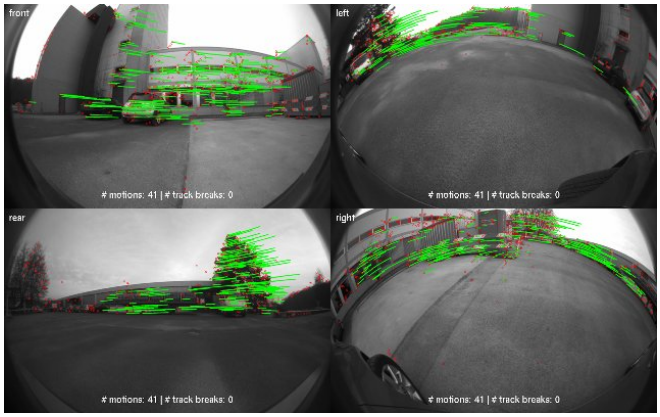


Fig. 4: Inlier feature point tracks.

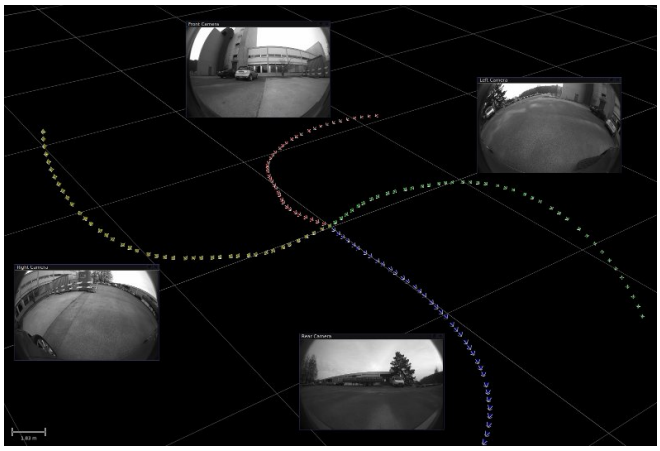


Fig. 5: Camera poses estimated by monocular VO for the four fish-eye cameras when the car is making a gradual right turn.

resulting in several sparse maps with different scales from at least one camera. In the first step, our approach solves for the pitch and roll components of the camera-odometry rotation by minimizing a least-squares cost function as in [7]. In the second step, we solve for the yaw component of the camera-odometry rotation, the camera-odometry translation and the scales for all sparse maps by minimizing another least-squares cost function. As the vehicle motion is planar, the z-component of the camera-odometry translation is unobservable.

For each camera, in the first step, we estimate q_{yx}, and in the second step, we estimate q_z, ${}^{O}\mathbf{t}_C = [t_x, t_y]^T$, and s_j for $j = 1, ..., m$ where ${}^{O}\mathbf{q}_C = q_z q_{yx}$ and ${}^{O}\mathbf{t}_C$ are the rotation quaternion and translation that transforms the camera frame to the odometry frame, q_z corresponds to the yaw quaternion, q_{yx} corresponds to the pitch-roll quaternion, and s_j is the scale for each of the m sparse maps.

We first start with the well-known hand-eye problem using the quaternion representation:

$$ {}^{O_{i+1}}\mathbf{q}_{O_i} \, {}^{O}\mathbf{q}_C = {}^{O}\mathbf{q}_C \, {}^{C_{i+1}}\mathbf{q}_{C_i} \tag{1} $$

$$ (\mathbf{R}({}^{O_{i+1}}\mathbf{q}_{O_i}) - \mathbf{I}) \, {}^{O}\mathbf{t}_C = s_j \mathbf{R}({}^{O}\mathbf{q}_C) \, {}^{C_{i+1}}\mathbf{t}_{C_i} - {}^{O_{i+1}}\mathbf{t}_{O_i} \tag{2} $$

where ${}^{O_{i+1}}\mathbf{q}_{O_i}$ is the unit quaternion that rotates odometry frame i to odometry frame $i+1$, ${}^{C_{i+1}}\mathbf{q}_{C_i}$ is the unit quaternion that rotates camera frame i to camera frame $i+1$, ${}^{O}\mathbf{q}_C$ is the unit quaternion that rotates the camera frame to the odometry frame.

1) Finding the pitch-roll quaternion: To estimate q_{yx}, we substitute ${}^{O}\mathbf{q}_C = q_z q_{yx}$ into equation 1 and use the fact that rotations around the z-axis commute to get:

$$ {}^{O_{i+1}}\mathbf{q}_{O_i} q_{yx} - q_{yx} \, {}^{C_{i+1}}\mathbf{q}_{C_i} = 0 \tag{3} $$

which can be written as a matrix vector equation:

$$ (\mathcal{L}({}^{O_{i+1}}\mathbf{q}_{O_i}) - \mathcal{R}({}^{C_{i+1}}\mathbf{q}_{C_i}))\mathbf{q}_{yx} = 0 \tag{4} $$

where the matrix which we call $\mathbf{S}$ is a 4×4 matrix and

$$ \mathcal{L}(\mathbf{q}) = \begin{bmatrix} w_q & -z_q & y_q & x_q \\ z_q & w_q & -x_q & y_q \\ -y_q & x_q & w_q & z_q \\ -x_q & -y_q & -z_q & w_q \end{bmatrix} \quad \mathcal{R}(\mathbf{q}) = \begin{bmatrix} w_q & z_q & -y_q & x_q \\ -z_q & w_q & x_q & y_q \\ y_q & -x_q & w_q & z_q \\ -x_q & -y_q & -z_q & w_q \end{bmatrix} $$

We have two constraints on the unknown $\mathbf{q}_{yx}$:

$$ x_{q_{yx}} y_{q_{yx}} = -z_{q_{yx}} w_{q_{yx}} \quad \text{and} \quad \mathbf{q}_{yx}^T \mathbf{q}_{yx} = 1 \tag{5} $$

Given $n \geq 2$ motions, we build the $4n \times 4$ matrix:

$$ \mathbf{T} = \begin{bmatrix} \mathbf{S}_1^T & ... & \mathbf{S}_n^T \end{bmatrix}^T \tag{6} $$

which has rank 2 in the absence of noise. We find the singular value decomposition $\mathbf{T} = \mathbf{U}\mathbf{S}\mathbf{V}^T$; the last two right-singular vectors $\mathbf{v}_3$ and $\mathbf{v}_4$ which are the last two columns of V span the null space of $\mathbf{T}$:

$$ \mathbf{q}_{yx} = \lambda_1 \mathbf{v}_3 + \lambda_2 \mathbf{v}_4 \tag{7} $$

We use the two constraints from equation 5 to solve for λ_1 and λ_2, and therefore, obtain $\mathbf{q}_{yx}$.

2) Finding the yaw quaternion, scales, and translation: Given that the z-component of the camera-odometry translation is unobservable due to planar motion, and removing the third row from equation 2, we have:

$$ \begin{bmatrix} \cos\phi - 1 & -\sin\phi \\ \sin\phi & \cos\phi - 1 \end{bmatrix} \begin{bmatrix} t_x \\ t_y \end{bmatrix} - s_j \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + {}^{O_{i+1}}\mathbf{t}'_{O_i} = 0 \tag{8} $$

where α is the yaw, $[p_1 \ p_2]^T$ are the first two elements of the vector $\mathbf{R}(\mathbf{q}_{yx}) \, {}^{C_{i+1}}\mathbf{t}_{C_i}$ and ${}^{O_{i+1}}\mathbf{t}'_{O_i}$ denotes the first two elements of the vector ${}^{O_{i+1}}\mathbf{t}_{O_i}$.

We reformulate equation 8 as a matrix vector equation:

$$ \begin{bmatrix} \mathbf{J} & \mathbf{K} \end{bmatrix} \begin{bmatrix} t_x \\ t_y \\ -s_j \cos\alpha \\ -s_j \sin\alpha \end{bmatrix} = - {}^{O_{i+1}}\mathbf{t}'_{O_i} \tag{9} $$

where $\mathbf{J} = \begin{bmatrix} \cos\phi - 1 & -\sin\phi \\ \sin\phi & \cos\phi - 1 \end{bmatrix}$ and $\mathbf{K} = \begin{bmatrix} p_1 & -p_2 \\ p_2 & p_1 \end{bmatrix}$.

For the m VO segments with $n_1 \geq 2$, ..., $n_m \geq 2$ motions respectively where $n = \sum_{i=1}^{m} n_i$, we build the $2n \times (2 + 2m)$

matrix:

$$\mathbf{G} = \begin{bmatrix} \mathbf{J}_1^1 & \mathbf{K}_1^1 & ... & 0 & 0 & ... & 0 & 0 \\ ... & ... & ... & 0 & 0 & ... & 0 & 0 \\ \mathbf{J}_{n_1}^1 & \mathbf{K}_{n_1}^1 & ... & 0 & 0 & ... & 0 & 0 \\ 0 & 0 & ... & \mathbf{J}_1^j & \mathbf{K}_1^j & ... & 0 & 0 \\ 0 & 0 & ... & ... & ... & ... & 0 & 0 \\ 0 & 0 & ... & \mathbf{J}_{n_j}^j & \mathbf{K}_{n_j}^j & ... & 0 & 0 \\ 0 & 0 & ... & 0 & 0 & ... & \mathbf{J}_1^m & \mathbf{K}_1^m \\ 0 & 0 & ... & 0 & 0 & ... & ... & ... \\ 0 & 0 & ... & 0 & 0 & ... & \mathbf{J}_{n_m}^m & \mathbf{K}_{n_m}^m \end{bmatrix} \tag{10}$$

where the matrices $\mathbf{J}_i^j$ and $\mathbf{K}_i^j$ correspond to the ith motion in VO segment j, and

$$\mathbf{G} \begin{bmatrix} t_x \\ t_y \\ -s_0 \cos\alpha_0 \\ -s_0 \sin\alpha_0 \\ ... \\ -s_m \cos\alpha_m \\ -s_m \sin\alpha_m \end{bmatrix} = - \begin{bmatrix} {}^{O_1}\mathbf{t}'_{O_0} \\ ... \\ {}^{O_{n+1}}\mathbf{t}'_{O_n} \end{bmatrix} \tag{11}$$

for which we use the least squares method to find the solution to ${}^{O}\mathbf{t}_C = [t_x \ t_y]^T$, s_j and α_j. We have estimated the scale s_j for each VO segment, and the translation ${}^{O}\mathbf{t}_C$; however, we have m hypotheses of α. We choose the best hypothesis that minimizes the cost function:

$$C = \sum_{i=0}^{n_j} ((\mathbf{R}({}^{O_{i+1}}\mathbf{q}_{O_i}) - \mathbf{I})\,{}^{O}\mathbf{t}_C - s_j \mathbf{R}(\alpha)\mathbf{R}(q_{yx})\,{}^{C_{i+1}}\mathbf{t}_{C_i} + {}^{O_{i+1}}\mathbf{t}_{O_i}) \tag{12}$$

We then refine the estimate of q_{yx}, α, ${}^{O}\mathbf{t}_C$, and s_j for $j = 1, ..., m$ by using non-linear optimization to minimize the above-mentioned cost function C.

C. 3D Point Triangulation

For each camera, we iterate through every frame, and at each frame, we find all features which are visible in the current frame and last two frames, and do not correspond to a previously initialized 3D scene point. We triangulate each feature correspondence in the last and current frames using linear triangulation [12]; if the reprojection error of the resulting 3D scene point in the second last frame does not exceed a threshold of 3 pixels, we associate the 3D scene point to the corresponding feature in the second last frame. Subsequently, each feature track is associated to the same 3D scene point which the first three features already correspond to. We use the Ceres solver [13] to run a bundle adjustment which optimizes the extrinsics and 3D scene points by minimizing the image reprojection error across all frames for all cameras.

D. Finding Local Inter-Camera Feature Point Correspondences

We iterate over each odometry pose in order of increasing timestamp, and maintain a local frame history for each camera; the length of the local frame history for one camera is defined by the distance the vehicle moves before a significant number of features observed in the current frame gets seen in a frame in any other camera.

At each odometry pose, for each possible pair of cameras, we match features in the first camera's current frame against those in the second camera's frame history, and find the frame in the history that yields the highest number of inlier feature point correspondences. Before the feature matching, the image pair is rectified on a common image plane which corresponds to the average rotation between the first camera's pose and the second camera's pose; the camera poses are computed using the current extrinsic estimate and the odometer poses. An example of a rectified image pair between the right and front cameras is shown in figure 6. This rectification step is to ensure a high number of inlier feature point correspondences; in contrast, we get a very low number of inlier feature point correspondences between images that are distorted and unrectified.

We find the subset of inlier feature correspondences that correspond to 3D scene points in the map by projecting 3D scene points in the map corrresponding to the first camera into the first rectified frame, and doing the same for the second camera. For each inlier feature correspondence, we find the pair of nearest projected 3D scene points; if the image distance between the feature points and corresponding projected 3D scene points exceeds 2 pixels, we reject the feature correspondence. We show an example of inlier feature correspondences with associated 3D scene points in figure 7.

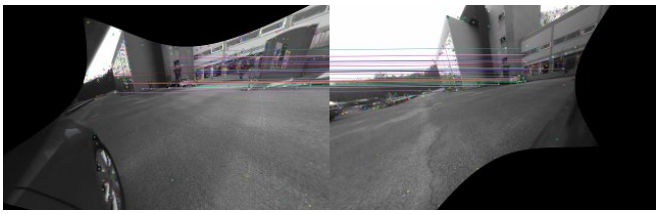


Fig. 6: Inlier feature point correspondences between rectified images in the right and right cameras.

Fig. 7: The subset of the same feature point correspondences that correspond to 3D scene points in the maps.

This step is critical to finding all optimal camera-odometry transforms that are consistent with each other on a global scale.

E. Loop Closures

In this step, we use the DBoW2 implementation of the vocabulary tree [14]. For each frame for each camera, we convert the features of the corresponding image into a bag-of-words vector and add the vector to the vocabulary tree. For each frame for each camera, we find n most similar images, and we filter out matched images which belong to the same

monocular VO segment and whose keyframe indices are within a certain range of the query image's keyframe index. This is to avoid unnecessary linking of frames whose features may already belong to common feature tracks. In most cases, we observe the linking of features between the last frame in one monocular VO segment for a particular camera, and the first frame in the next monocular VO segment for the same camera; this linking contributes towards the global consistency of the sparse map of landmarks.

F. Full Bundle Adjustment

We perform full bundle adjustment that optimizes all intrinsics, extrinsics, odometry poses, and 3D scene points. Since we only estimate the relative heights of the cameras, we obtain the absolute heights of the cameras above the ground by taking a hand measurement of the height of one camera above the ground, and using this height measurement to obtain the absolute heights of all other cameras. The accuracy of this hand measurement is not important to image-processing applications.

V. EXPERIMENTS AND RESULTS

We compare our extrinsic calibration results against those generated by a AR-marker-based extrinsic calibration currently used in the V-Charge project. In this AR-marker-based calibration, three different types of AR markers are placed around the vehicle: camera markers that are visible in at least one camera, floor markers that are placed on the ground around the car, and wheel markers that are placed on the rear wheels to help identify the nominal odometry frame. The origin of this nominal odometry frame is defined to be the projection of the center of the rear axle to the ground plane, while the x, y, and z axes are parallel to the longitudinal axis, lateral axis and vertical axis of the vehicle respectively. The cameras on the vehicle together with a calibrated handheld camera capture images of the markers. Subsequently, the markers are detected in all images, and a bundle adjustment is run to optimize the poses of the markers and cameras. At the end of the bundle adjustment, the poses of the cameras are extracted with respect to the nominal odometry frame; the locations of the floor markers define the ground plane, and the wheel markers define the rear axis of the vehicle.

We show the sparse maps of landmarks for each camera and combined sparse map at each stage of the extrinsic calibration process in figure 8. We observe that the final sparse maps are much more well-defined and well-aligned to one another compared to the sparse maps computed by triangulation at the beginning. The average reprojection error associated with the final maps and over 450000 points is 0.5 pixels.

We also tabulate in table I the estimated angles between the front camera and the other three cameras for both our estimated extrinsics and the AR-marker-based extrinsics. for the left, rear, and right cameras. Similarly, we tabulate in table II the unit estimated translations between the front camera and the other three cameras to ignore the effect of scale. We show a visual comparison of the final estimated extrinsics with the extrinsics computed by the AR-marker-based calibration in figure 9. A green sphere marks the origin of the odometry frame. We observe that both sets of extrinsics are almost identical in terms of the cameras' relative transforms; the scale of the estimated extrinsics is 0.44% smaller on average which is statistically insignificant. There is a y-offset of 10 cm between the estimated and AR-marker-based extrinsics; the explanation is that our calibration uses odometry data unlike the AR-marker-based calibration.

We are unable to compare our approach with [8] as their 3D similarity transform step fails in our experimental settings. A vast majority of scene points are located far away from the cameras, and thus, the estimates of their 3D coordinates have significant noise.

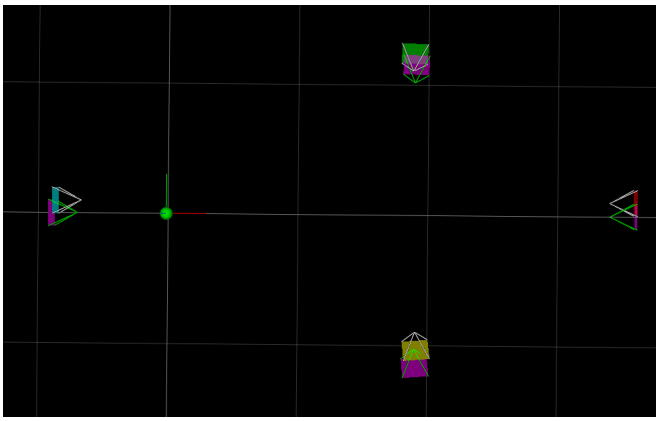
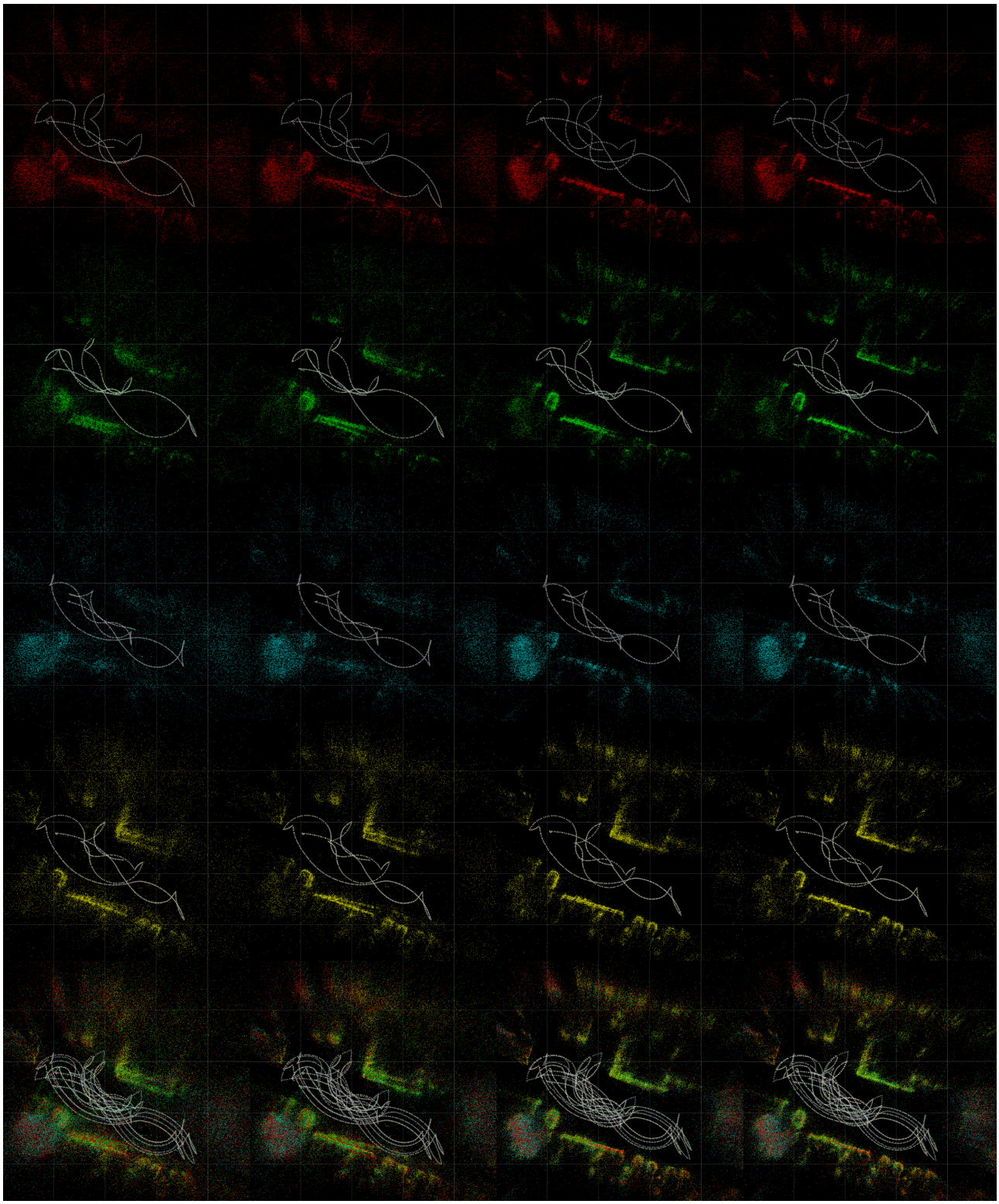


Fig. 9: Comparison of extrinsics generated by our pipeline and the AR-marker-based calibration method. The cameras corresponding to the AR-marker-based extrinsics are marked in purple with green lines. The grid resolution is 1 m.

We perform an experiment to show that the estimated calibration parameters are more accurate than that of the AR-marker-based parameters; we use a multi-view plane-sweep-based stereo algorithm [15] with semi-global block matching to compute dense depth maps from rectified images for each camera using only the odometry poses, camera intrinsics, and extrinsics without any camera pose optimization. The plane sweep is extremely sensitive to the accuracy of the intrinsic and extrinsic calibration parameters; we compare the depth maps computed by the plane sweep using our estimated intrinsics and extrinsics with the depth maps computed with the intrinsics and extrinsics generated by the AR-marker-based calibration. We show in figure 10 that the use of our estimated parameters results in a more accurate dense map especially on the ground compared to the use of the AR-marker-based parameters. This result demonstrates the higher accuracy of our estimated parameters, and can be attributed to the fact that our pipeline takes into account odometry data, and makes use of a high number of features over a wide range of distances to ensure optimal accuracy.

On an 2.80 GHz Intel Core i7 PC, our pipeline takes approximately 15 seconds to compute the intrinsic parameters for each camera using 100 images each, and approximately 90 minutes to compute the extrinsic parameters using a



(a) After 3D scene point triangulation.

(b) After BA with 3D scene point triangulation.

(c) After BA with local inter-camera correspondences.

(d) After BA with loop closures.

Fig. 8: Sparse maps of landmarks at various stages of the pipeline. The first four rows correspond to the front, left, rear, and right cameras respectively, while the last row shows the combined map from all cameras. The grid resolution is 10 m.

TABLE I: Estimated angles between front camera (reference camera) and the other three cameras.

	Left camera	Rear camera	Right camera
Our calibration	86.7507°	179.559°	88.1723°
AR-marker-based calibration	86.8544°	178.262°	89.3818°

TABLE II: Estimated unit translations between front camera (reference camera) and the other three cameras.

	Left camera	Rear camera	Right camera
Our calibration	$[-0.5397 - 0.1987 - 0.8180]$	$[-0.0120 - 0.0902 - 0.9959]$	$[0.5199 - 0.2184 - 0.8258]$
AR-marker-based calibration	$[-0.5912 - 0.1780 - 0.7862]$	$[-0.0277 - 0.1048 - 0.9941]$	$[0.5143 - 0.2364 - 0.8244]$

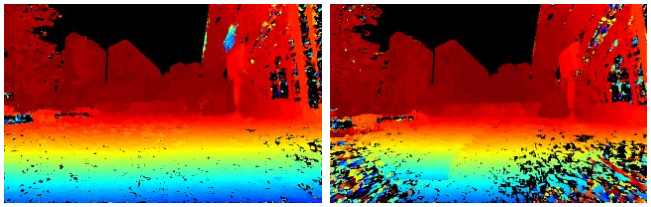


Fig. 10: 3-view plane sweep stereo with front camera. Forward camera motion is challenging for plane sweep stereo. The left depth map is computed using our estimated parameters, while the right depth map is computed using the AR-marker-based parameters.

total of 2000 frames; most of the time is spent finding local inter-map feature point correspondences and on bundle adjustment.

A video showing the stages of the calibration pipeline together with the calibration package can be accessed at `http://people.inf.ethz.ch/hengli/camodocal`.

VI. CONCLUSIONS

Our calibration pipeline is able to compute a highly accurate intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry in an arbitrary environment without prior knowledge of the rig setup and without the need for environment modifications. Due to the automated nature of the pipeline, minimal human intervention is required, and the pipeline is easy to use. Furthermore, the pipeline produces a globally-consistent map of landmarks. The pipeline does not require a special calibration setup; no substantial time-related or material-related costs are incurred. As a result, the pipeline has huge potential benefits for the robotics community and automotive industry.

From experiments, we demonstrate the sub-pixel accuracy of our intrinsic calibration with regards to the unified projection model, and thus, show the unified projection model to work well in large-scale environments. An accurate extrinsic calibration with multiple cameras and odometry allows for real-time odometry-aided vision applications which are more robust and accurate compared to vision-only applications in the sense that we avoid computationally-intensive camera pose estimations from 2D-3D correspondences; occasionally, a camera pose estimated from a low number of 2D-3D correspondences can be significantly inaccurate and lead to a catastrophic failure. At the same time, we produce a high-quality globally-consistent sparse map of landmarks using features from all cameras.

We plan to work on an online version of the calibration. The motivation is that several cameras on our car platforms are mounted on movable parts, for example, a car door, and that vision applications will fail when the cameras are moved.

VII. ACKNOWLEDGMENTS

We thank Christian Häne for generating the depth maps from plane sweep stereo. The first author was funded by the DSO National Laboratories Postgraduate Scholarship. In addition, this work was supported in part by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant #269916 (V-Charge).

REFERENCES

[1] C. Mei and P. Rives, Single View Point Omnidirectional Camera Calibration from Planar Grids, In *Proc. International Conference on Robotics and Automation*, 2007.

[2] D. Scaramuzza, A. Martinelli, and R. Siegwart, A Toolbox for Easy Calibrating Omnidirectional Cameras, In *Proc. International Conference on Intelligent Robots and Systems*, 2006.

[3] K. Daniilidis, Hand-Eye Calibration Using Dual Quaternions, In *International Journal of Robotics Research, 18(3):286-298*, 1999.

[4] J. Brookshire and S. Teller, Extrinsic Calibration from Per-Sensor Egomotion, In *Proc. Robotics: Science and Systems*, 2012.

[5] G. Antonelli, F. Caccavale, F. Grossi, A. Marino, Simultaneous Calibration of Odometry and Camera for a Differential Drive Mobile Robot, In *Proc. International Conference on Robotics and Automation*, 2010.

[6] A. Censi, A. Franchi, L. Marchionni, and G. Oriolo, Simultaneous calibration of odometry and sensor parameters for mobile robots, In *IEEE Trans. on Robotics*, 2013.

[7] C. Guo, F. Mirzaei, and S. Roumeliotis, An Analytical Least-Squares Solution to the Odometer-Camera Extrinsic Calibration Problem, In *Proc. International Conference on Robotics and Automation*, 2012.

[8] G. Carrera, A. Angeli, and A. Davison, SLAM-Based Automatic Extrinsic Calibration of a Multi-Camera Rig, In *Proc. International Conference on Robotics and Automation*, 2011.

[9] M. Rufli, D. Scaramuzza, and R. Siegwart, Automatic Detection of Checkerboards on Blurred and Distorted Images, In *Proc. International Conference on Intelligent Robots and Systems*, 2008.

[10] B. Kitt, R. Rehder, A. Chambers, M. Schoenbein, H. Lategahn, and S. Singh, Monocular Visual Odometry using a Planar Road Model to Solve Scale Ambiguity, In *Proc. European Conference on Mobile Robots*, 2011.

[11] D. Nister, An Efficient Solution to the Five-Point Relative Pose Problem, In *IEEE Trans. on Pattern Analysis and Machine Intelligence, 26(6):756-770*, 2004.

[12] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2004.

[13] S. Agarwal and K. Mierle, Ceres Solver: Tutorial & Reference, Google Inc.

[14] D. Galvez-Lopez, and J. Tardos, Bags of Binary Words for Fast Place Recognition in Image Sequences, In *IEEE Trans. on Robotics, 28(5):1188-1197*, 2012.

[15] R. Yang, and M. Pollefeys, Multi-Resolution Real-Time Stereo on Commodity Graphics Hardware, In *Proc. International Conference on Computer Vision and Pattern Recognition*, 2003.