Auto Calibration for Self Driving Cars

Aarushi Agarwal (2016216)

PAPER I

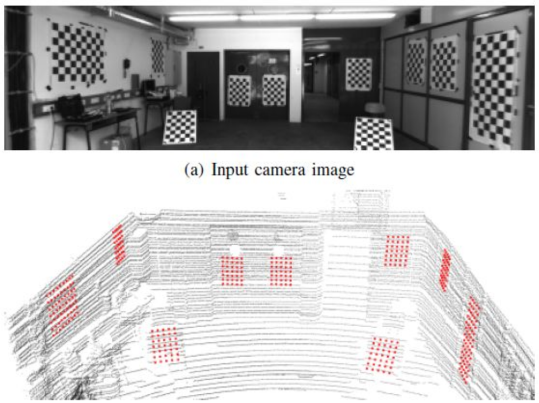
Automatic Camera and Range Sensor Calibration using a single Shot **(ICRA (2012) - Geiger)**

Automatic Camera and Range Sensor Calibration using a single Shot

1. Toolbox with **Web Interface** for fully automatic
    a. Camera to camera calibration
    b. Camera to range calibration
2. Robust to varying imaging conditions.
3. Input :-
    a. Uses single image and range scan per sensor.
    b. Distance between inner checkerboard corners.
4. Corner detection outperforms state of the art.
5. Processing Time $<$ 5 mins
6. **Assumption**
    a. Common field of view.
    b. Planer checkerboards presented

How different from prev approaches?

1. Multiple images of a single calibration pattern required.
    a. This method finds multiple calibration patterns in a single image
2. Number of checkerboard rows and columns as input.



(a) Input camera image

(b) Input range data and calibration result

<u>Camera to Camera Calibration</u>

Four Stages -

1. Corner Detection
    a. Compute corner likelihood using different prototypes (kernels)
2. Sub Pixel Corner and Orientation refinement
3. Structure Recovery (for every shot captured by individual cameras)
4. Matching checkerboards (correspondence) between different images.
    a. Consider all possible combination of checkerboards between two images.
    b. Compute 2D similarity transformation

Camera to Range Sensors

1. Determines 6 parameters (r1, r2, r3, t1, t2, t3) in a single shot (image).
2. Segmentation
    a. Identify contiguous regions (planes) in range data
3. Generate transformation between every pair of planes in camera and lidar frame - **Registration**
4. Fine registration - gradient descent
5. Solution selection

Drawback

Overlapping field of view assumption.

Codes references

Paper 1

1. https://github.com/ftdlyc/libcbdetect
2. https://github.com/qibao77/cornerDetect
3. https://github.com/onlyliucat/Multi-chessboard-Corner-extraction-detection-
4. https://github.com/eherozhao/StereoCameraCalibration

NOTE :- Only detecting corners and structure recovery code present

PAPER II

CamOdoCal: Automatic Intrinsic and Extrinsic Calibration of a Rig with Multiple Generic Cameras and Odometry (**IROS (2013)**)

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

1. Automated pipeline for both intrinsic and extrinsic calibration
2. No assumption like - overlapping views

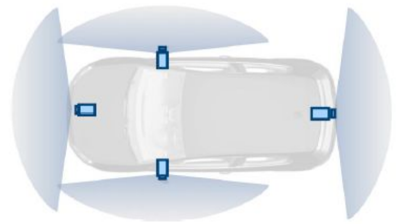


Fig. 2: The sensor coverage of the test car platforms. The four fish-eye cameras are marked in blue.

Intrinsic Calibration

1. Automatic, requires chessboard (multiple poses for each camera)
2. Corner detection
3. Spline (an imaginary line passing through a row of corners)
4. All the corners lying beyond a certain threshold from the spline marked invalid.
5. Initial estimate of intrinsic parameters with non-linear optimization.

Extrinsic Calibration (Five Stages)

<u>Stage I (Monocular VO)</u> - To get the feature point in image and estimating 6-DOF trajectory (camera poses)

1. Runs Monocular VO algorithm for every camera
2. Uses 5 point algorithm instead of 8 point algorithm
3. Frames are taken after every 0.25 m change in odometer pose.
4. OpenCV SURF to extract feature points (corners/edges) and descriptors (vector representation).
5. At each iteration of VO - sliding window of 10 frames, considering first 3 frames of camera with fixed pose.

OUTPUT - Camera poses and feature tracks for each camera

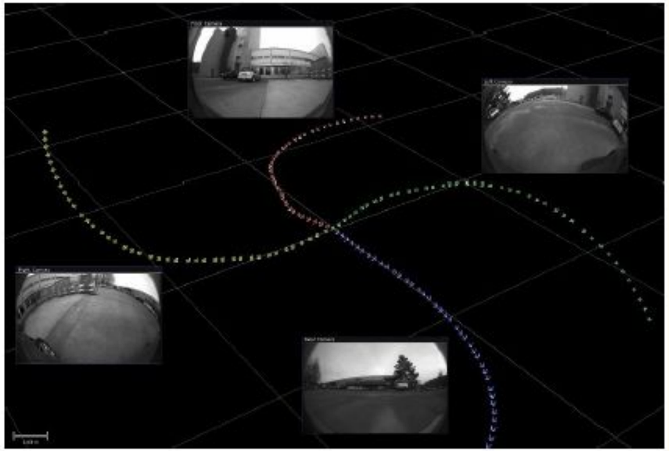
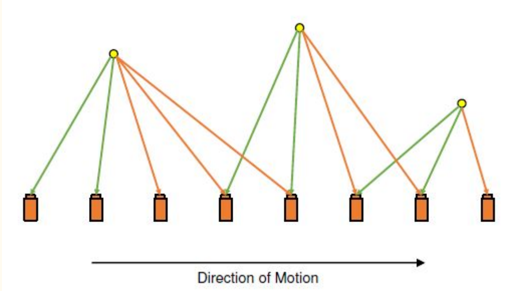


Fig. 5: Camera poses estimated by monocular VO for the four fish-eye cameras when the car is making a gradual right turn.

Extrinsic Calibration (Five Stages)



Stage II (Initial estimate of Camera - Odometry Transform)

1. VO estimate breaks commonly.
2. Several sparse maps from each camera.
3. Use all sets of VO to find an initial estimate of camera-odometry transform (R and t)

Stage III (3D Point Triangulation)

4. Find all the features which are visible in three consecutive frames and do not correspond to a previously initialized 3D point.
5. Uses linear triangulation instead of three view triangulation [back projecting 3D point from image points].
6. Reproject 3D onto image plane and check if it doesn't exceed a threshold of 3 pixels.
7. Output 3D Scene map(Note - the Camera Odometry Transform is used)

Extrinsic Calibration (Five Stages)

Stage IV (Finding local inter camera feature point correspondences)

1. Since no overlapping view assumption, we store frame history for each camera.
2. **Length of history** - Distance the vehicle moves before X number of features in current frame are seen in any frame of other camera.
3. At each odometry pose -
   a. For every permutation of camera pairs, match features in first camera with second camera's frame history.
   b. Find the frame in the history which has higher number of feature correspondences.
4. Feature matching procedure -
   a. Image pair rectified on a common image plain. (By getting average rotation between the two camera poses).
   b. Rectifying helps in higher feature correspondence.

Extrinsic Calibration (Five Stages)

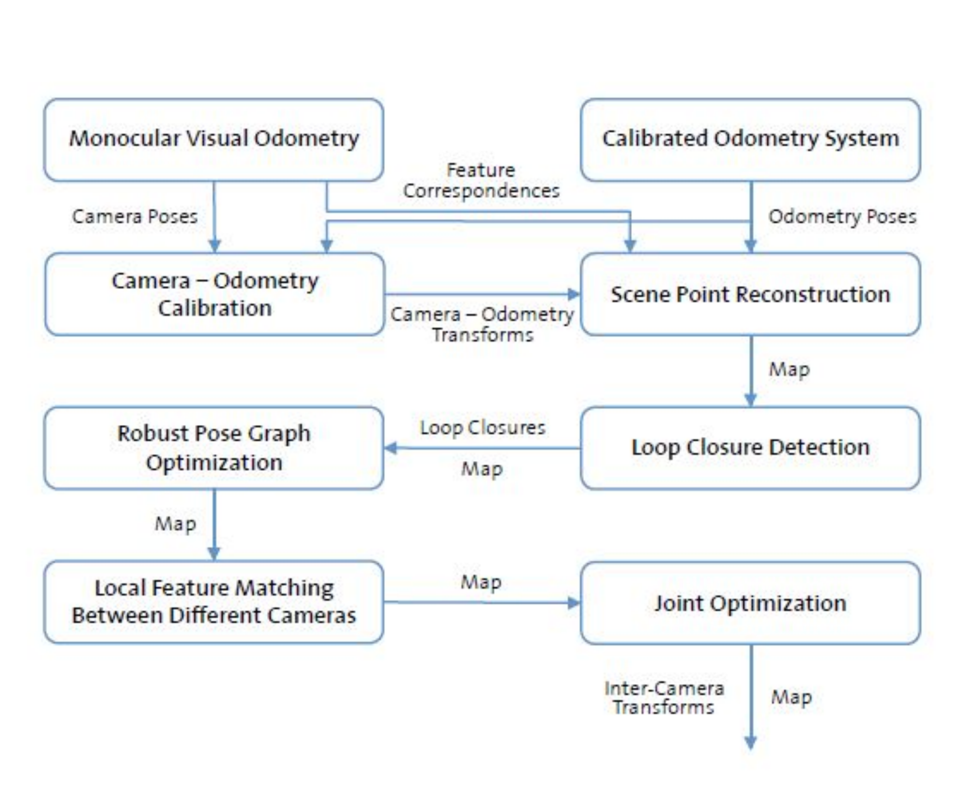
Stage IV (Continued...)

1. Find the nearest pair of 3D points in map of both the cameras.
2. Project them onto the image of respective cameras in rectified frame.
3. If the projections and the feature pixels are within 2 pixels , establish the feature correspondence.
4. Finally, we get a subset of feature correspondences.

Stage V (Loop Closures)

1. Create a bag of words (features) vector for each frame of each camera.
2. For each frame find the **n most similar images** , filter out which belong to same monocular segment.
3. This helps in linking of features between discontiguous monocular VO segments.
4. Leads to global consistency of the mapping.

Whole Pipeline



Drawback and Benefits

<u>Benefits</u>

1. Fully automated without environment modifications.
2. Produces a globally consistent map of landmarks.
3. Sub pixel accuracy over other state of the art models

<u>Drawback</u>

1. Highly computationally expensive. (90 mins for extrinsic calibration)
2. We need a calibrated odometry system.

Codes references

Paper 2

1. https://github.com/hengli/camodocal
2. https://github.com/gaowenliang/-camera_model

PAPER III

Extrinsic calibration of a set of range cameras in 5 seconds without pattern (**IROS (2014)**)

Extrinsic calibration of a set of range cameras in 5 seconds without pattern

1. An algorithm to calibrate range sensors and RGB-D cameras.

2. Planes extracted from depth images using **region-growing** approach.

3. Plane correspondences set
   a. Rough guess of **relative poses of camera provided by user**

4. Correspondences refines using heuristics -
   a. **Angle between normal of both planer region** is smaller than a threshold.
   b. **Distance of both the planes from camera center** $<$ Threshold
   c. The sizes of planes are enough.
   d. Refined through RANSAC algorithm.

5. After the correspondences are set, rotation and translation are solved using MLE.

Codes references

Paper 3

1. https://github.com/dzunigan/extrinsic_calib

PAPER IV

Automatic extrinsic calibration of depth sensors with ambiguous environments and restricted motion **IROS(2017))**

Automatic extrinsic calibration of depth sensors with ambiguous environments and restricted motion

1. Unsupervised algorithm using ego-motion information known as “Delta Transforms”

2. Delta-transform, written as Dti , is defined as the transform that takes points from sensor i's frame of reference at timestep t + delta(t) to timestep t.

$$D_1^t A p_2^{t+\Delta t} = A D_2^t p_2^{t+\Delta t} = p_1^t.$$

$$R_1 R_2 = R R_2,\ R T_2 + T = R_1 T + T_1$$

3. Further R and T vectors are computed by representing the above vectors in quaternion form.

PAPER V

Extrinsic calibration of multiple RGB-D cameras from line observations **RAL(2018)**

Extrinsic calibration of multiple RGB-D cameras from line observations

1. Unsupervised algorithm to calibrate 3D LiDARs and RGB-D Cameras. Used for extracting lines and matching them.

2. 2D line segments are extracted from the images by applying a Canny filter in combination with RANSAC - corresponding normals are calculated. These are transformed to 3D.

3. 3D planes are extracted by applying RANSAC for plane fitting. 3D lines are extracted from the range sensor data as **plane intersections.**

4. **Correspondences set between the lines and Transformations calculated.**

PAPER VI

LiDAR-Camera Calibration using 3D-3D Point correspondences

1. Propose a very accurate and repeatable method to estimate extrinsic calibration parameters in the form of 6 degrees-of-freedom between a camera and a LiDAR.
2. The intrinsic parameters of the camera should be known before starting the LiDAR-camera calibration process.
3. The method is proposed keeping in mind lower-density LIDAR like Velodyne 16.
4. **Using 3D-3D Correspondences:**
    - Fusion is used to visualize the accuracy of extrinsic calibration parameters.
    - Aruco tags method is proposed.
5. **Experimental Setup**
    - Aruco markers are to be printed and put on a hard board inclines at some angle with dimensions known.
    - Lidar can be kept at a distance of 2-2.5m from the cardboard.

Fusion using lidar_camera_calibration

- To verify the method in a more intuitive manner, lidar_camera_calibration was used to fuse point clouds obtained from two stereo cameras.
- First, we compare the calibration parameters obtained from lidar_camera_calibration against meticulously measured values using tape by a human.
- There is a very minute translation error (~1-2cm) and almost no rotation error using stereo cameras.

Fig. 6. Experimental setup with rectangular cardboard and ArUco markers using 3D-3D correspondences.

What’s done?

- Intrinsic Calibration of Point Gray Camera is done using checkerboard technique provided as a ROS Package.
- Lidar_camera_calibration.launch file is build and run successfully with all the required windows showing.
- Two 5X5 A3 size Aruco markers are printed.

What’s remaining?

- Setup of markers on the cardboard.
- Measurement of distances by tape. (Has to be figured out how)
- Fusion of Point Cloud obtained by 2 cameras.

Code References

1. https://github.com/ankitdhall/lidar_camera_calibration
2. http://wiki.ros.org/lidar_camera_calibration
3. Video for fusion result for overlapping field of view
4. Video for fusion result of non overlapping field of views with cameras kept at ~80 degrees apart.

Thank You!

Notes

Stereo - Multi eyed system

Mono - Single eyed system

Fundamental matrix - Un calibrated camera

Essential matrix - Calibrated camera

Both homogenous, rank - 2 , 3*3

Globally consistent map - Map in line with the ground truth

Loop Closure - Loop closing is the act of correctly asserting that a vehicle has returned to a previously visited location.