Category-based SLAM

1st Mikhail Kurenkov
Skoltech
Moscow, Russia
Mikhail.Kurenkov@skoltech.ru

2nd Ilya Belikov
Skoltech
Moscow, Russia
Ilia.Belikov@skoltech.ru

3rd Nikolay Zherdev
Skoltech
Moscow, Russia
Nikolay.Zherdev@skoltech.ru

***Abstract*—This project tackles the problem of object position estimation and camera localization using object SLAM with GTSAM backend. Objects in real world are considered as semantic features or landmarks and their keypoints are used to retrieve 3D pose of object. In original paper [9] keypoints are extracted using neural network. However we used simplified approach by utilizing synthetic keypoints on a laptop model. With found on a single image 2D locations of laptop's keypoints, we use the pose and shape adjustment scheme proposed in [8] to lift this 2D keypoints to 3D wireframe, thereby using GTSAM backend for estimation of the shape and pose of the object with respect to camera.**

***Index Terms*—SLAM, object SLAM, GTSAM, keypoints**

I. Introduction

Simultaneous Localization and Mapping (SLAM) finds various real-world applications such as autonomous navigation, visual inspection, mapping, and surveillance. Monocular cameras have evolved as popular choices for SLAM, especially on platforms such as hand-held devices and Micro Aerial Vehicles (MAVs). Most state-of-the-art monocular SLAM systems[7] operate on geometric primitives such as points, lines, and planar patches. Others operate directly on images, without the need for expensive feature extraction steps [3].

However, both these sets of approaches lack the ability to provide a rich semantic description of the scene. Recognizing and keeping track of objects in a scene will enable a robot to build meaningful maps and scene descriptions. Object-SLAM is a relatively new paradigm [11], [10], [6] towards achieving this goal.

Summarized in one sentence, object-SLAM attempts to augment SLAM with object information so that robot localization, object location estimation (in some cases, object pose estimation too), and mapping are achieved in a unified framework. There are two dominant paradigms in object-SLAM research, depending on the way objects are characterized in the SLAM framework. In the first paradigm object level (instance-specific) models are assumed to be available beforehand. However, the very nature of monocular SLAM with scale ambiguity coupled with the loss in information due to projection onto the image plane renders this paradigm unfeasible for monocular object-SLAM systems. The second paradigm, assumes a generic model, regardless of the object category. For instance, models all objects as ellipsoids, and [4] model all objects as cuboids. Both these approaches suffer a few disadvantages. Relying on object-level models will result in the need to have precise object models for all instances of an object category. On the other hand, generic models do not give much information about an object beyond the object category label. In many applications, such as manipulation for instance, it is advantageous to know the object pose.

In this paper, we are following ORIGINAL article for monocular object-SLAM, that combines the best-of-both-worlds. To enjoy the expressive power of instance-specific models yet retain the simplicity of generic models, we construct category-specific models, i.e., the object category is modeled as a whole. We employ the widely used linear subspace model to characterize an object category and define object observations as factors in a SLAM factor graph [2],

The object-SLAM back-end estimates robot trajectory and map, as well as poses and shapes of all objects in the scene.

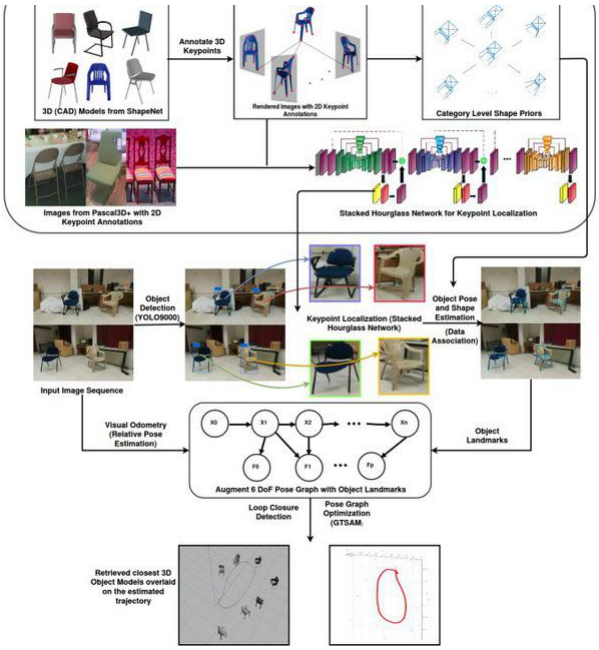

Fig. 1. Overall structure of the category-based SLAM

II. Data preparation

A. Key points annotation

Manually added colored beacons are traced on image with cv2.inRange method, and for each found contour the X-Y coordinates are calculated with cv2.moments method and saved as dictionary to numpy binary file.

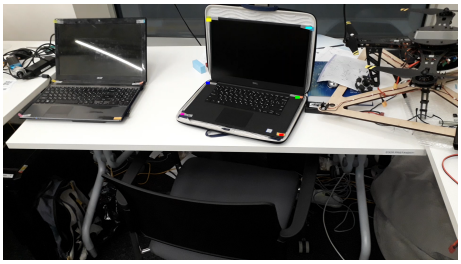

Fig. 2. A laptop augmented with colored beacons

III. Construction category specific model

The core idea of our method is that we use category - based representation of objects, that is each object in the category (in our case it is laptop category) represents as a wireframe. The nodes of this wireframe are keypoints of the 3d object, for example corners or centers of symmetry, they are connected by lines, and thus build a skeleton of this object. Different instances of this category have different lengths of edges and angles between lines which connect keypoints, but we can calculate the average instance of this category and characterized each object as deviation from this mean object. Mathematically these parameters are represented as PCA coordinates in space of this category or in our case in space of laptops. But for simplicity in our case we assume that each laptop characterized by three parameters: width, height and angle between keyboard and monitor.

To calculate these parameters we use idea from [8]. They use adjustment scheme which lift 2d points from image to 3d points of object's wireframe. For this aim we must have annotated keypoints for example from neural network for correct data association. When keypoints is annotated it is possible to find a position relatively camera and shape of objects which belong to one category by solving non linear optimization problem.

A. Optimization problem

The core idea of optimization problem is that if we know shape and position of objects we can transform 3d keypoints to 2d keypoints. This transformation can be done by formula $\pi(KTs_k(\Lambda))$, where π is projection function from 2d homogeneous coordinates to 2d image coordinate in pixels, K is camera matrix, T is transform matrix of from camera frame to object frame and $s_k(\Lambda)$ is function which return 3d object points from shape parameters, Λ - shape parameters and k is index of a key point. To calculate cost function of our optimization problem (1) we use euclidean norm on the image between points which we receive from 3d keypoints and extracted 2d keypoints from original image. Also we use regularization function $\rho(\Lambda)$ which permit from big deviation of shapes from the mean instance of the category.

$$J = min(\sum_k ||\pi(KTs_k(\Lambda)) - S_k||^2 + \sum \rho(\Lambda)) \quad (1)$$

B. Result of optimization problem

The optimization have been done on python language by using package scipy.optimize. Firstly, optimization was tested on the synthetic data, and then on the real laptops with artificially annotated keypoints. Result of the second experiment you can see on 3. One of a big problem of this method is initialization of optimization problem, because this cost function have many local minimal and optimizer found incorrect minimal value. One of solution of this problem is convex relaxation.

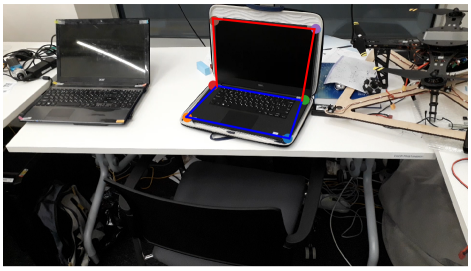

Fig. 3. Laptop with markers

IV. 3D pose SLAM

For finding position of the camera in the world and laptops it is used simultaneous localization and mapping method (SLAM) based on graph optimization [2] . The back end which realized this method is GTSAM [1]. Also iSAM updating technology[5] is used for incremental updating and solving optimization problem.

A. Camera calibration

For knowledge of transformation from 3d space to camera space we had to find the camera matrix which represents focal length and center point of the camera. Camera was calibrated using a chessboard according to opencv calibration pipeline. Found matrix also was used for laptop pose estimation.

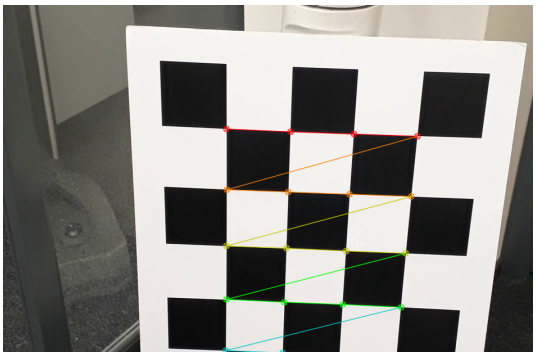

Fig. 4. Chessboard calibration

B. Visual odometry

For prediction we used visual odometry based on FAST feature detector and RANSAC algorithm with Nistr's 5-point motion estimator. Result of visual odometry tested on a sequence from KITTI dataset is presented on the image 5. The factors in factor graph are 3d pose factor edges, that is the transformation between frames is characterized by rotation (Euler angles) and translation.

Fig. 5. Laptop with markers

C. Landmarks measurements

For update step we used laptops as 3d landmarks. Positions of these landmarks were retrieved via solving optimization problem with keypoints estimation. The factor in factor graph is 3d pose that is we consider that laptop have translation and rotation component of transformation between camera frame and object frame.

V. Result and discussion

In this project we accomplish following work:

- Prepare dataset with annotated keypoints
- Make visual odometry
- Solve optimization problem for pose estimation
- Prepare SLAM framework for joint camera pose estimation and landmark localization

On the future we want to enhance our SLAM algorithm and make it work.

References

[1] Frank Dellaert. Factor graphs and gtsam: A hands-on introduction. Technical Report GT-RIM-CP&R-2012-002, GT RIM, Sept 2012.

[2] Frank Dellaert and Michael Kaess. Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing. *The International Journal of Robotics Research*, 25(12):1181–1203, dec 2006.

[3] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. pages 834–849. Springer, Cham, 2014.

[4] Dorian Gálvez-López, Marta Salas, Juan D. Tardós, and J.M.M. Montiel. Real-time monocular object SLAM. *Robotics and Autonomous Systems*, 75:435–449, jan 2016.

[5] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, feb 2012.

[6] Beipeng Mu, Shih-Yuan Yuan Liu, Liam Paull, John Leonard, and Jonathan P. How. SLAM with objects using a nonparametric pose graph. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2016-Novem, pages 4602–4609. IEEE, oct 2016.

[7] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardos. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163, oct 2015.

[8] J. Krishna Murthy, Sarthak Sharma, and K. Madhava Krishna. Shape priors for real-time monocular object localization in dynamic environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1768–1774. IEEE, sep 2017.

[9] Parv Parkhiya, Rishabh Khawad, J. Krishna Murthy, Brojeshwar Bhowmick, and K. Madhava Krishna. Constructing Category-Specific Models for Monocular Object-SLAM. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9, feb 2018.

[10] Renato F. Salas-Moreno, Richard A. Newcombe, Hauke Strasdat, Paul H.J. Kelly, and Andrew J. Davison. SLAM++: Simultaneous Localisation and Mapping at the Level of Objects. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1352–1359. IEEE, jun 2013.

[11] Niko Sunderhauf, Trung T. Pham, Yasir Latif, Michael Milford, and Ian Reid. Meaningful maps with object-oriented semantic mapping. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2017-Septe, pages 5079–5085. IEEE, sep 2017.