A System Framework for Localization and Mapping Using High Resolution Cameras of Mobile Devices

Lifeng Liu Jian Li
Futurewei Technologies, Inc.
111 Speen Street, Framingham, MA, 01701, USA
{lifeng.liu, jian.li}@futurewei.com

Abstract

We propose a hierarchical framework for processing high-resolution images on mobile devices for visual SLAM. It is based on the insights from analysis of new progress in primary features' detection, object detection and pose estimation. A rectification/unwarping operation is applied in regions of interest (ROIs) to improve the object/parts classification/detection performance; the object-part spatial relationships are created and contribute in map building, object detection, and localization; and a geometric constraints based pose refinement is followed to further improve the localization accuracy. Our design facilitates the more accurate pose estimating and localization using mobile devices for SLAM, and Augmented Reality/Mixed Reality applications.

1. Introduction

The resolution of cameras on mobile devices, such as smart phones and AR glasses/wearables, has increased rapidly in recent years. For example, Apple iPhone X has dual 12-megapixel cameras, Huawei P30 has a 40-megapixel camera, and Xiaomi announced that it would introduce a smartphone featuring a 100-megapixel sensor in late 2019.

However, the stereo and monocular systems for object detection, depth computing and their corresponding benchmarks are still mostly focused on low-resolution images [23]. For example, in KITTI dataset [4], the corresponding camera resolution is 1.4 megapixel, and many stereo images in its database have only 0.4 megapixel resolution, which is far more less than the typical smart phone camera resolution.

On one hand, the high-resolution images provided from the new devices will make it possible to achieve high accuracy in object detection and localization, facilitate the computation of AR applications and improve user's immersive experiences. If a user can get accurate location information by just moving its phones to capture pictures and match the 3D map (in the cloud or downloaded locally or created using the video sequence from the phone cameras), it would offer great experience for the users. On the other hand, the higher computation requirements for processing such big images inevitably create new challenges.

In addition, the baselines of the multiple cameras on the smart phones are smaller, and their lens can also have limitations. Nonetheless, a user can move the phone to create multiple virtual cameras with bigger baselines and larger coverage if the camera poses can be calculated accurately in real time. Moreover, researches of computing depth from monocular images have made a lot of progress.

Therefore, a new processing framework is desired to handle the challenges and make it practical to fully integrate the new achievements in mobile devices' computation power, camera resolution and computer vision models. Note that the recent DNN based vision models were mostly developed and suited for low-resolution images.

To address the challenges, we propose a novel computation framework to process high-resolution images in a hierarchical way to support high-accuracy object detection, map building, and localization. The framework, as shown in Figure 1, includes the following components and stages:

1. Use down-sampling to get low resolution images for primary features' detection: lines (vertical lines, parallel lines, horizontal lines, and semantic lines), planes (walls, furniture surfaces, road/ground surface). Depth information can be fused for the detection if available. See details in Sec. 3.1.
2. Initially estimate the camera pose based on the obtained primary features. This might be just camera's orientation/viewpoint relative to the world (not exactly location information, i.e. transformation is not fully fixed yet. For example, the camera pose is recovered up to a scale or similarity transformation). See details in Sec. 3.2.
3. For planar regions (or can be treated as planar regions), rectify the corresponding regions (from high resolution images) to get front view, rectified (and cropped) images and conduct more accurate object/parts detection and locating. As a result, the perspective transforms and affine transforms can be

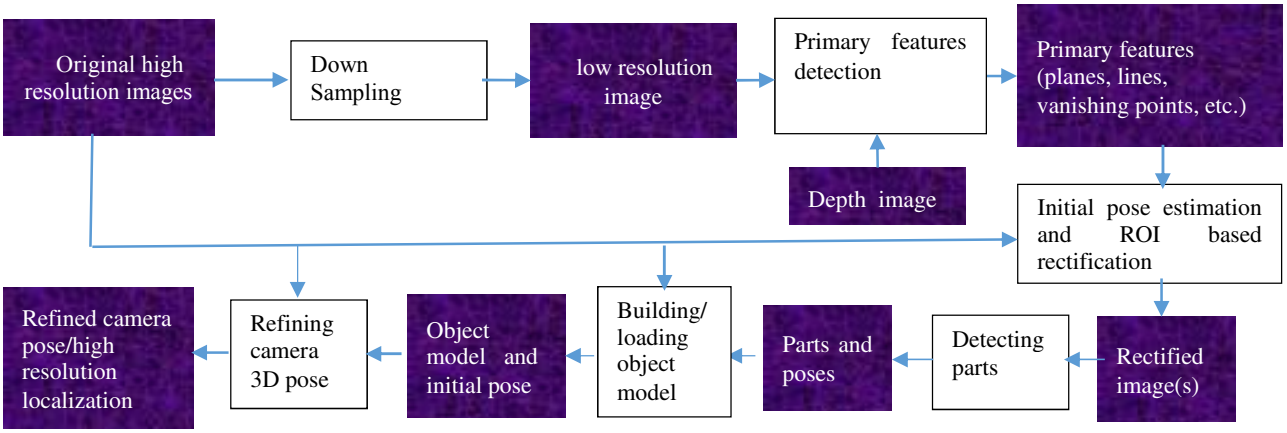


Figure 1: Architectural overview of map building and localization using high resolution images

eliminated from the rectified/unwarped images, and higher accuracy can be achieved in the followed vision tasks. See details in Sec.3.3.

4. Build object model from the detected parts and create the object-part spatial relationships and use them in map building and object detection/localization. See details in Sec.3.4.
5. From the object model, re-project to the original high-resolution image (or the edge/feature maps) for high resolution alignment optimization to minimize geometric errors. This can give more accurate object pose relative to the camera and facilitate the applications with high accuracy localization requirement (such as AR/MR, navigation applications, etc.). See details in Sec. 3.5.

2. Related Work and Motivation

2.1. Sensor fusion using guided geometries

Infra-red based depth sensors were already added to mobile devices, such as supporting Apple faceID application. Different sensor types have different limitations in obtaining the depth information in the environment, either in resolution, density, or working ranges.

Combining multiple sensors to improve the environment perception and localization is an interesting research field. Guided stereo matching [14] makes use of sparse, yet precise depth information collected from an external source, such as a LiDAR to assist state-of-the-art deep learning frameworks for stereo matching.

DenseFusion[21] estimates 6D object pose using RGB-D images. DeepLiDAR[15] combines LiDAR data and color images and makes use of surface normal to guide depth prediction for outdoor scene.

These researches show that simple and sparse geometric information (such as surface normal) are very helpful for depth computing and pose estimation. In our proposed system design, the geometric information (from color images or other sensors) is utilized for converting/unwarping the image regions for better performance of object detection and localization.

2.2. Data representation optimization

In Pseudo-LiDAR[20], the images from the stereo cameras are used to generate a 3D point cloud which is then rotated in 3D to produce a top-down perspective of a vehicle's surroundings. This allows for improved accuracy that puts their approach on par with LiDAR solutions. On the popular KITTI benchmark, achieves impressive improvements over the existing state of-the-art in image-based performance — raising the detection accuracy of objects within 30m range from the previous state-of-the-art of 22% to an unprecedented 74%.

Ma et al. [13] provide a stacked U-Net for document image unwarping, and it greatly improves the effectiveness of the state-of-the-art text detection systems. Due to the output from single U-Net may not be satisfactory, a second U-Net is used for further refining the unwarped image. Their evaluation showed that unwarping improved the multi-scale structural similarity (MS-SSIM) from previous 0.13 to 0.41.

Tai et al. [18] propose Equivariant Transformers (differentiable image to image mappings) via specially derived canonical coordinate systems, and use the estimated pose parameters to apply an inverse transformation to the image. The equivariant transformers improve the error rate of the baseline on projective MNIST by 2.79%, a relative improvement of 43%.

The insight is that the data representation has a big impact on the performance of the computer vision tasks (up to 3X in accuracy based on the above examples). This might be because the majority of the training examples for the deep learning models of object recognition are front view

images, and the data augmentation cannot fully fill the gap of lacking complete coverage of different view angles. Affine transformations except translation are not easy to augment with since that requires accessing the full three-dimensional scene models. Augmenting data with combinations of different transformations would result in an enormous dataset. The fact that objects are composed of parts, makes it even harder to catch all possible configurations of object parts.

It is also clear that humans prefer to see objects with front views (for example, people like cameras' correction functions to adjust a skewed presentation picture to a straight view angle). Therefore, we believe that the object recognition and pose estimation can gain significant improvement if the rectified front view image can be used as input to the object/part recognition and pose estimation models.

2.3. Limitations of end to end based pose estimation

In our workflow, object/part localization/pose refinement is separated from the classification/recognition stage: First execute classification/recognition on the rectified images, then refining the pose using the original high-resolution images.

One motivation for additional pose refinement is that there are accuracy limitations using end to end DNN for pose estimation. Sattler et al. [16] investigate the camera pose estimation using end to end DNN, and claim that end to end approaches based on convolutional neural networks do not achieve the same level of pose accuracy as 3D structure-based methods for camera pose estimation. They predict that absolute pose regression (APR) techniques are not guaranteed to generalize form training data in practical scenarios (APR is more closely related to image retrieval approaches than to methods that accurately estimate camera poses via 3D geometry).

Therefore, more accurate pose estimation should be based on 3D geometry structures, i.e. regression to minimize the geometric errors. For example, Liu et al. [11] first use an anchor-based method for regression of the dimension and orientation of the object, then sample a large number of candidates in the 3D space and project the 3D bounding boxes to 2D image individually to get the best candidate by exploring the spatial overlap between the proposals and the object (using a FQNet). Wang et al. [21] use re-projection errors between multi-view images for computing the geometric consistency cost.

Stereo R-CNN in [9] combines 2D boxes with sparse key points, viewpoints, and object dimensions to calculate a coarse 3D object bounding box, then recovers the accurate 3D bounding box by a region-based photometric alignment to achieve sub-pixel matching accuracy. In their method, Object RoI is treated as a geometric constraint entirely, and the dense alignments bring significant improvements (precision was improved from around 40% to over 80%)

Accordingly, in our system architecture, the final pose refinement stage is based on minimizing the geometric errors via projecting object model features to the original images and utilizing the object-part spatial relationships.

Mathematic analysis from the stereo vision shows that the accuracy of locating geometric structures in images directly influences the accuracy of depth computation. As shown in Figure 2, a 3D physical point has projections on the images of the two cameras in a stereo vision system, the disparity is the x coordinate difference for the image points in left image and right image: $d = x_L - x_R$

Assume the camera focal length is f and the camera centers' distance is b (the baseline), the depth value of the

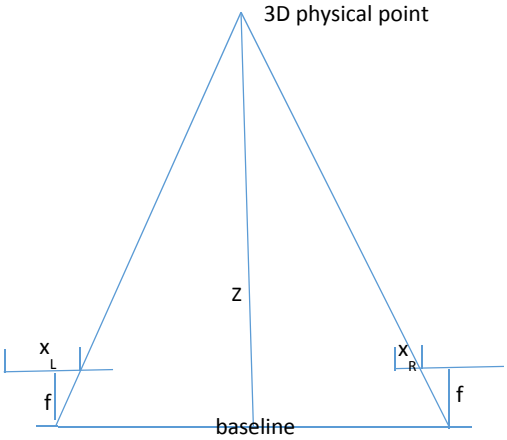


Figure 2: Illustration of stereo geometry

physical point is then computed as follows:

$$z = \frac{b*f}{d} \quad (1)$$

The depth error is related with disparity error [3] in the following way:

$$\delta z = \frac{b*f}{d} - \frac{b*f}{d+\delta d} = \frac{z^2*\delta d}{b*f+z*\delta d} \approx \frac{z^2*\delta d}{b*f} \quad (2)$$

The computation of *disparity* is based on the feature correspondence for the 3D geometric structure, if assuming the accuracy of disparity is about pixel level, then the image resolution determines the accuracy of the depth estimation.

For the same size of field of view, 16-megapixel camera has the pixel size of *1/4* of the pixel size of 1-megapixel camera in X/Y dimension, so the depth error can be reduced up to only *1/4* of the depth error of 1-megapixel camera.

As a result, pose refinement based on minimizing geometric errors can take full advantage of the high-resolution images from modern smart phone cameras, and will not be limited by the accuracy of the end to end neural network models.

2.4. Object-part spatial relationships

An object can be seen as a geometrically organized set of interrelated parts. For many man-made objects, different parts might be treated as on different planes. Kosiorek et al. [7] describe an unsupervised version of capsule networks to discover parts and objects directly from an image: Stacked Capsule Autoencoders. The highly structured decoder networks are used to train the encoder networks which can compose the parts into coherent objects. The trainable fixed object-part relations are composed with the detected object pose and compared with the detected part poses for estimating the part likelihood.

It is more robust for an object detection system to make explicit use of the geometric relationships among the object's parts since the relationships are viewpoint invariant.

To improve the map representation and localization accuracy for visual SLAM, we propose to split the objects into parts based on the planar regions of the object (as shown in Figure 3), and store the rectified images to associate with the detected objects' parts, and encode the geometrical relationships among the object's parts via the 3D planar bounding box representations of the part in the object coordinate system. Therefore, viewpoint invariant localization can be conducted robustly in this scheme for visual SLAM, and the object/parts detection based on appearance matching can be simplified based on the rectified images.

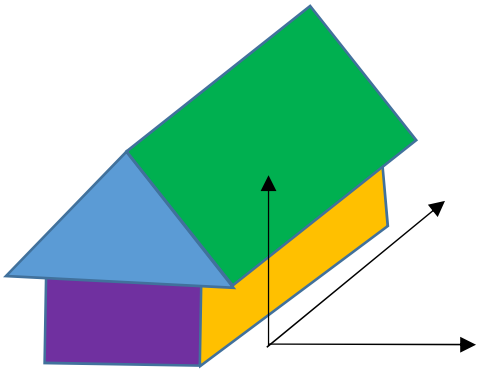


Figure 3: Parts of an object with plane representation (different colors are used to label different planar parts of the object).

2.5. Our contributions

Different from the prior arts of only handling low-resolution images, or not considering the challenges from view angle differences, our proposed system framework integrates the prime feature detection, region unwarping, detection using object-part spatial relationships, and geometric constraints based pose refinement in one unified architecture, and make the computation more efficient and more accurate for using high-resolution images for visual SLAM:

- Primary features' detection makes automatic unwarping feasible, and the ROI based unwarping and detection can significantly reducing the computation requirements;
- The front view appearance from unwarping operation and the object-part spatial relationship information used in detection will improve the recognition robustness and accuracy significantly;
- The geometric constraints based refinement will achieve high accuracy in localization.

3. System Design

3.1. Primary features' detection

The first stage in our system architecture is detecting primary features, such as vanishing points, planes, orthogonal or parallel lines. These primary features offer important information for camera pose estimation. Recent progress in deep learning brought architectures and methods for detecting planes and lines with more information and semantic meanings.

For example, Sun et al. [19] proposed HorizonNet to detect the boundaries of floor-wall, ceiling-wall, and wall-wall. These boundaries help to reconstruct the 3D room layout and infer the room shape.

Planar regions are popular in outdoor/indoor environments, especially abundant in man-made environments, such as building/architectures in city scenes. Planar regions offer key geometric cues for scene understanding, viewpoint/orientation estimation and robot navigation. Recently deep-learning-based methods were proposed for planar region detection.

Liu et al. [10] propose a PlaneRCNN to detect planes in monocular images, and planar regions in the images can be detected and the corresponding plane parameters, depth map can be reconstructed. The reconstructed 3D planes will be used for further (and finer) classification, recognition, and pose estimation/locating in our scheme.

Lee et al. [8] proposes a semantic line detector using convolutional neural network with multi-task learning (classification about whether a candidate is a sematic line or not, and regression about determining the offset for refining the line location). The resulting semantic lines are important for some computer vision tasks, such as estimating the levelness of an image.

By the way, the 3D planes' intersections are 3D lines, and these intersection lines generally have semantic meanings (such as intersection between walls and ground) and are useful for estimating the camera's pose in the environment.

Vanishing points are also useful for camera pose estimation, and CNN networks were proposed for vanishing point detection [6].

Although in some situations using the detected primary features, the camera pose is only recovered up to a similarity transform [5], but that is good enough for the image rectification in the following sections.

3.2. Camera pose estimation using primary features.

As discussed in Sec. 1, better camera pose estimation can benefit the stereo vision using smart phones.

3D plane pairs or 3D line pairs from two cameras can be used to estimate the relative relationship among the cameras, therefore facilitate the depth information computing using large baselines (via moving camera) to improve the accuracy.

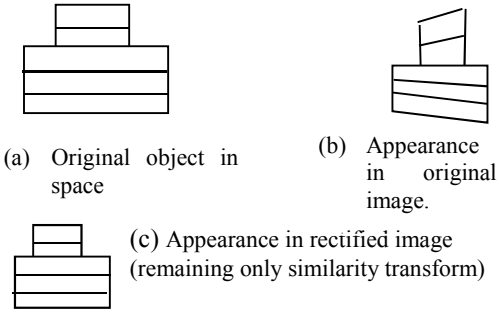
The book "multi-view geometry in computer vision" [5] lists ways of using points and line features for camera calibration: For example, vanishing lines may be computed given equally spaced coplanar lines.

3.3. Image rectification/unwarping

Image rectification in general refers the transforming one image from stereo images to match the other image, as a result, the rectified image has the row to row correspondence to the image from the other camera and supports the disparity computation.

Here, we talk a different rectification: unwarp the original image regions according to the camera pose information and the orientation of the object/part of interest and generate a new image with front view for the object/part.

This process can be treated as using a virtual camera which has front view of the object(s), as illustrated in Figure 4.



(a) Original object in space

(b) Appearance in original image.

(c) Appearance in rectified image (remaining only similarity transform)

Figure 4: Illustration of rectification to obtain front view.

As shown in Figure 5 for each pixel in the unwarped image, the pixel value is obtained by projection the pixel to the object plane, get the intersection point, and re-project the intersection point to the original image to fetch the corresponding pixel value. The formulas for the projection are explained in equations in Sec. 3.4.

The orientation of the virtual camera can be based on the plane normal estimation from the image in the original camera (obtained in Sec. 3.1 and Sec. 3.2).

The unwarped images have one to one correspondence for the planar regions (or nearly planar regions) in the low-resolution image but are generated using the original high-resolution image.

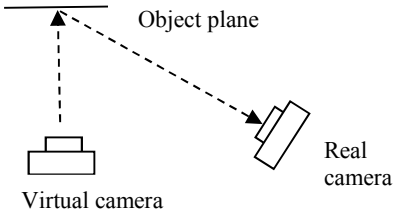


Figure 5 : Top view of virtual camera setup for unwarping the original image.

Object/part recognition based on the front view image would achieve high accuracy (as discussed in Sec. 2.2), and the pose can be refined further once there are the object models (which are available for man-made structures from the corresponding CAD model, or 3D model/map created based on SLAM).

As an extension, combining with camera-aware neural network properties [2], it might be possible to train a neural network to take the camera information and ROI region's plane information to do image unwarping and get the rectified image (as shown in Figure 6). One note is that for generalizing this kind of neural networks, data normalization might be needed (as used in [2] and [5]).

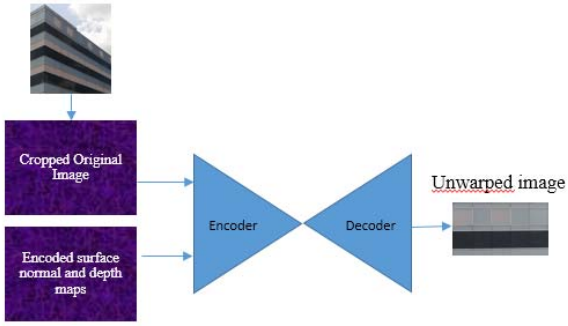


Figure 6: Illustration of rectification network.

3.4. Detection and locating based on object-part spatial relationships.

The architecture of building and using object model for detection and localization is shown in Figure 7. The detected parts and their associated poses from the rectified images are used to compose the object model, which keeps the object-part spatial relationships, and the generated object models also contribute to map building and pose refinement for high-accuracy localization.

From the detected parts, object encoder network [7] might be used to compose the parts to objects and generate the object-part relationships. Alternatively, the semantic

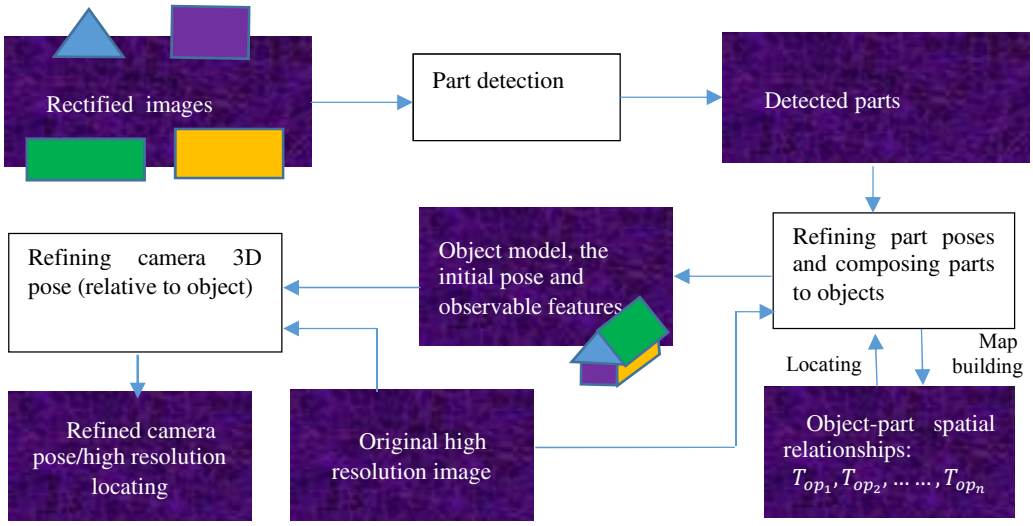


Figure 7: Object detection, map building, and localization using object-part spatial relationships.

segmentation results [1] [12] might be used to group parts to objects. The object model building is explained in the following.

For each planar part, its 2D bounding box (on the part plane) might be used for the part representation.

The intersections of the part planes are 3D lines and can be trimmed as 3D line segments based on the bounding boxes for the associated parts. Therefore, we can use the geometric center of the end points of the intersected line segments as the object center, i.e. the origin of the 3D object coordinate system is defined as follows:

$$objectCenter = \frac{\sum_{i=1}^{N}(P_{i0}+P_{i1})}{2N} \quad (3)$$

Where N is the number of line segments, and P_{i0} *and* P_{i1} are the end points of the *ith* line segment.

Alternatively, the object center might be the centroid of all the corner points of the bound boxes of its parts.

Z axis of the 3D object coordinate system can be based on the plane normal of its main part, and X axis and Y axis might be based on the intersected line segments of this plane with other planes, as shown in Figure 3.

Every part of the object will be represented by a 3D transformation T_{op} which maps the unit square on the XY plane (with plane equation of $z = 0$) to the part plane bounding box in the object coordinate system. The unit square on the XY plane has the following corner points: *(0,0,0), (1, 0,0), (0,1,0), (1, 1, 0)*. Hence, the origin and axis directions of the part coordinate system are well defined based on the bounding box. T_{op} contains x-scale, y-scale and rigid transforms, and an example is shown in Figure 8, where T_{op} includes a rotation of 90 degrees around X axis, and a 2X scale in X direction. The unit square is transformed to a rectangle in *X'Z'* plane in the *X'Y'Z'* coordinate system as shown with T_{op} :

$$T_{op} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

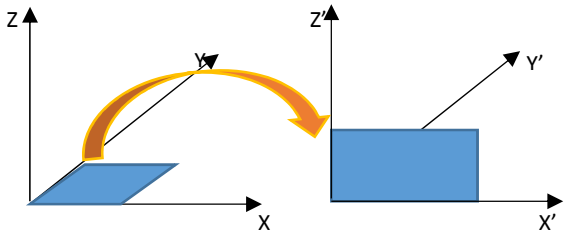


Figure 8: Object-part transform is used to map the unit square to the part bounding box.

The corresponding rectified image (front view of the part) is also associated with the part in the object model.

As shown in the following structure of the object model, the intersections between the parts of the object are also good features, and can be combined with the features of each part for pose refinement and accurate localization (see Sec. 3.5):

$$Object\ Model$$

$$\left\{ \begin{array}{cccc} & Intersection & features\ of & parts \\ part_1 & T_{op_1} & FeatureList_1 & RectifiedImage_1 \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\ part_n & T_{op_n} & FeatureList_n & RectifiedImage_n \end{array} \right\}$$

Borrowing the idea from capsule network [7], the object detection accuracy can be improved based on the part relationships (such as 3D distances between parts, angle

between part planes, etc.) without using a large number of training examples. The rectified images (front views of the parts) might be used as part templates [7], and may also make few-shot learning feasible.

The generated object models can be included in the environment maps of SLAM and used for high accuracy localization as explained in the next section.

Please note that the pose refinements (in Sec. 3.5) can also be applied for part pose refinements in building the object model.

3.5. Geometric constrained pose refinements

In the pose refinement stage, the objective is to minimize the projection errors from the object 3D model to the camera images (if there are multiple cameras, all of them can be used for refining the object 3D pose, as shown in Figure 9, two images are used for projecting).

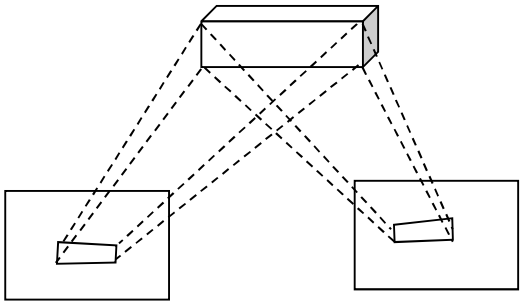


Figure 9: Projection of object model features to image space (only the features from the front part are illustrated as examples).

As long as we know (or have hypothesis of) the camera pose relative with the object, we can project the object model features to the image planes. The projecting and re-projecting equations are listed in the following:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = M * X \quad (4)$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x'/z' \\ y'/z' \end{bmatrix} + \begin{bmatrix} t1 \\ t2 \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} t1 \\ t2 \end{bmatrix} \right) \quad (6)$$

$$X = M^{-1} \begin{bmatrix} x'' * depth(p) \\ y'' * depth(p) \\ depth(p) \end{bmatrix} \quad (7)$$

where, X is the 3D model point, $\begin{bmatrix} x \\ y \end{bmatrix}$ is the corresponding image point, and $M, t1, t2$ compose as the camera matrix that represents the projective transformation. M, $t1, t2$ are computed based on the intrinsic parameters of the camera and the camera pose (relative to the object) information.

One kind of geometric error metric is to compare the feature matching between the projected feature locations and the image feature locations, e.g. the cost function for bundle adjustment:

$$cost = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{p} v_{ijk} distance(P_{ijk}, P'_{ijk})^2 \quad (8)$$

Where i is the image index (or camera index) for the images used for the pose refinement, j is the part index, and k is the feature index; v_{ijk} is binary variable to denote whether feature k of part j is visible in image i; P_{ijk} is the image feature location detected in image i for feature k *of* part j, and P'_{ijk} is the projected location of feature k of part j in image i based on the object model and pose information.

For each point on the part feature, the corresponding physical position in the camera coordinate system is computed as follows:

$$X = T_o * T_{op_j} * P \quad (9)$$

where, P is one point of the features in part j, T_{op_j} is the pose of part j in the object coordinate system, and T_o is the object pose in the camera coordinate system. Applying equations 4 and 5 on the transformed feature point (output of equation 9) generates $the\ values\ of\ P'_{ijk}$.

Another way is to compare the pixel values between the left/right images for the same 3D model feature points:

$$cost = \sum_{k=1}^{m} \sum_{i=1}^{n-1} \sum_{j=2}^{n} diff\ (Intensity_{ik},\ Intensity_{jk})^2 \quad (10)$$

Where $Intensity_{ik}$ is the pixel intensity on image i for object feature k, and $Intensity_{jk}$ is the pixel intensity on image j for object feature k. This method can be treated as warping one image accordingly and compare with the other image with the same object. Difference computation might be based on the normalized intensity.

The feature locations in image might also be extracted using deep neural networks, such as the key points detection method used in [11].

It is also possible to refine the object pose and adjust the part-object relationships (T_{op} in Sec. 3.4) simultaneously for map building.

Traditionally, these kinds of regression are done using nonlinear optimization methods, such as Levenberg-Marquardt minimization algorithms, and a coarse 3D pose can be used as the starting point for the optimization procedure. However, these methods may require feature engineering to get the corresponding features among the 3D model and every input image with the object. Deep neural networks were proposed for computing the geometric errors by training with labelled samples, such as FQNet in [11], which learns computing the geometric errors by learning from examples without manual feature engineering. Still it requires to compare many samples of the 3D pose candidates.

Therefore, a feasible way is to use an end to end neural network to compute the geometric errors (or comparing which candidate has better matching), and the minimization method will make use of this network for error/cost comparison.

4. Advantages

The down-sampling step can significantly reduce the computation requirements in primary feature detection and still obtain the coarse orientation and depth information for automatic rectification purpose. The high-resolution pose refinement is limited to the regions of objects of interest.

Based on the insights from analysis of new progress of primary features' detection, object detection and pose estimation, we have proposed to apply rectification/unwarping operation in regions of interest (ROIs) to improve the object/part classification/detection performance, and a geometric constraints based pose refinement to further improve the locating accuracy. As such, our preliminary analytical model yields up to 3X precision improvement via ROI rectification and further improvement with geometric constraints (including the object-part relations) based pose refinement.

5. Conclusion

A hierarchical design is proposed to integrate primary features' detection, 3D-orientation based ROI rectification for part detection, building and using object models with object-part spatial relationships, and geometric-constraints based pose refinement. This novel workflow will make good use of the high-resolution images available on modern mobile devices, to build maps using objects with better semantic meanings and spatial structures and achieve more accurate pose estimation and localization. Based on analysis of existing works, ROI rectification may improve the precision of object detection up to 3X, and object-part relationships and 3D structures are used for geometric-constraints based pose refinement, which may further improve the accuracy to another fold.

Our staged system design also has the benefits of reducing the computing requirement for high-resolution images, generating object model with semantic meanings, easy troubleshooting and clear explanation of the roles in each stage, and will improve immersive user experience of AR/MR applications where pose estimation/localization accuracy is crucial.

The proposed hierarchical architecture will potentially help in 3D map building and localization based on mobile devices and enhance the user experiences and unlock more applications.

References

[1] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A.L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *ICLR, 2014*

[2] Jose Facil, Benjaawamin Ummenhofer, Huizhong Zhou. CAM-Convs: Camera-Aware Multi-Scale Convolutions for Single View Depth. *CVPR 2019*.

[3] David Gallup, Jan-Michael Frahm, Philippos Mordohai, Marc Pollefeys. Variable Baseline/Resolution Stereo. *CVPR 2008*.

[4] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013

[5] Richard Hardly, and Andrew Zisserman. Multi-view geometry in computer vision, second edition. *Cambridge University press*, 2004

[6] Florian Kluger, Hanno Ackermann, Michael Ying Yang, and Bodo Rosenhahn. Deep Learning for Vanishing Point Detection Using an Inverse Gnomonic Projection. *GCPR 2017: Pattern Recognition* pp 17-28.

[7] Adam R. Kosiorek, Sara Sabour, Yee Whye Teh, Geoffrey E. Hinton. Stacked Capsule Autoencoders. *https://arxiv.org/abs/1906.06818.*

[8] Jun-Tae Lee, Han-Ul Kim, Chul Lee, Chang-Su Kim. Semantic Line Detection and Its Applications. *ICCV 2017*.

[9] Peiliang Li, Xiaozhi Chen, and Shaojie Shen. Stereo R-CNN based 3D Object Detection for Autonomous Driving. *CVPR 2019*.

[10] Chen Liu, Kihwan Kim1, Jinwei Gu, Yasutaka Furukawa, Jan Kautz. PlaneRCNN: 3D Plane Detection and Reconstruction from a Single Image. *CVPR 2019*.

[11] Lijie Liu, Jiwen Lu, Chunjing Xu, Qi Tian, Jie Zhou. Deep Fitting Degree Scoring Network for Monocular 3D Object Detection. *CVPR 2019*.

[12] J. Long, E. Shelhamer, and T. Darrell， Fully convolutional networks for semantic segmentation. *CVPR 2015*.

[13] Ke Ma, Zhixin Shu, Xue Bai, Jue Wang, Dimitris Samara. DocUNet: Document Image Unwarping via a Stacked U-Net. CVPR 2018.

[14] Davide Pallotti, Fabio Tosi, Stefano Mattoccia. Guided Stereo Matching. *CVPR 2019*.

[15] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, Marc Pollefeys. DeepLiDAR: Deep Surface Normal Guided Depth Prediction for Outdoor Scene from Sparse LiDAR Data and Single Color Image. *CVPR 2019*.

[16] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, Laura Leal-Taixe. Understanding the Limitations of CNN-based Absolute Camera Pose Regression. *CVPR 2019*.

[17] J. Long, E. Shelhamer, and T. Darrell， Fully convolutional networks for semantic segmentation. *CVPR 2015*.

[18] Kai Sheng Tai, Peter Bailis, Gregory Valiant. Equivariant Transformer Networks. *International Conference on Machine Learning, 2019.*

[19] Cheng Sun, Chi-Wei Hsiao, Min Sun, Hwann-Tzong Chen. HorizonNet: Learning Room Layout with 1D Representation and Pano Stretch Data Augmentation. *CVPR 2019*.

[20] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Weinberger. Pseudo-LiDAR from Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving, *CVPR 2019*.

[21] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Mart´ın-Mart´ın. DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion. *CVPR 2019*.

[22] Qingshan Xu and Wenbing Tao. Multi-Scale Geometric Consistency Guided Multi-View Stereo. *CVPR 2019*.

[23] Gengshan Yang, Joshua Manela, Michael Happold, Deva Ramanan. Hierarchical Deep Stereo Matching on High-resolution Images. *CVPR 2019*.