UNIVERSITÀ DEGLI STUDI DI CATANIA
Dipartimento di Matematica e Informatica
Dottorato di Ricerca in Matematica e Informatica XXXI Ciclo

Emiliano Spera

Egocentric Vision Based Localization of Shopping Carts

Tesi di Dottorato di Ricerca

Prof. Giovanni Maria Farinella

Anno Accademico 2017 - 2018

Abstract

Indoor camera localization from egocentric images is a challenge computer vision problem which has been strongly investigated in the last years. Localizing a camera in a 3D space can open many useful applications in different domains. In this work, we analyse this challenge to localize shopping cart in stores. Three main contributions are given with this thesis. As first, we propose a new dataset for shopping cart localization which includes both RGB and depth images together with the 3-DOF data corresponding to the cart position and orientation in the store. The dataset is also labelled with respect to 16 different classes associated to different areas of the considered retail. A second contribution is related to a benchmark study where different methods are compared for both, cart pose estimation and retail area classification. Last contribution is related to the computational analysis of the considered approaches.

Acknowledgements

I would like to tanks my supervisor Prof. Giovanni Maria Farinella as well as Prof. Sebastiano Battiato and Dr. Antonino Furnari for their guide and support during my PHD studies.

Contents

Chapter 1

Introduction

1.1 Motivation

The ability to estimate the position and orientation of a mobile object from egocentric images is crucial for many industrial applications[14, 11, 13]. In robotics, for instance, the opportunity to use a camera for the auto-localization of the robots is a cheap solution and not invasive for the context. In outdoor contexts the more traditional technology used for localization is the GPS, differently the classic solutions to address indoor localization include the employment of RF-ID tags [1] or Beacons [2] and the use of fixed cameras monitoring the different areas of the indoor context [4]. While these technologies can be used to obtain effective localization systems, they both have downsides. For instance, GPS and Beacons are not very accurate [2] and struggle with occlusions which can attenuate their signal [3], whereas pipelines based on fixed cameras need the installation of camera networks and the use of complex algorithms capable of re-identifying people across the different scenes.

To overcome these issues, localization using egocentric images has been investigated both in the context of indoor and outdoor environments [11, 13, 14] according to different levels of localization precision, in function of the environment characteristics and of application in which is involved, e.g. 6 Degrees Of Freedom (6-DOF) pose estimation [11, 13] for 3D location estimation, 3-DOF pose estimation [9] for 2D location estimation and room-based location recognition [22, 40, 41].

As it has been investigated by Santarcangelo et al. [40], in the context of retail stores, the position of shopping carts equipped with a camera can be obtained exploiting computer vision pipelines for scene classification. Such information can be used to analyse the customer behaviours, trying to infer, for instance, where they spend more time, which areas of the store are preferred (e.g., fruit, gastronomy, etc.) and how the placement of products can affect sales. Image-based localization abilities are also necessary to allow a robot to navigate and monitor the store or to assist the costumers [21].

1.2 Aims and approaches

This thesis work is focused on the problem of localizing shopping carts in retail stores from egocentric images acquired by cameras mounted on shopping carts. Differently from other indoor environments, retail is a very hard and specific environment for camera localization presenting unique properties and challenges:

- It is often large scale environment
- The 3D structures are typically repetitive (e.g. many shelves with same dimensions)
- similar products, form a visual point of view, can be in different parts of the store
- many visually dissimilar products are spatial near producing a strong visual difference between images acquired in similar position.

Figure 1.1 shows some examples of the typical variability of egocentric images acquired in a retail store.

In the last years the growing interest related to localization by means of egocentric images bring the scientific community to produce different dataset to address this task in indoor and outdoor environment [11, 14, 13]. Despite this growing interest a large dataset to address the task of shopping cart localization in a retail store was still missing. Hence, during my PHD activity, we proposed a new large scale dataset of RGB and depth images acquired in a retail store by using cameras mounted on shopping carts. By means of careful semi-automatic 3D reconstruction and registration procedures, each image has been labelled with a six Degrees Of Freedom (6-DOF) pose summarizing the 3D position of the shopping cart, as well as its orientation in the 3D space.

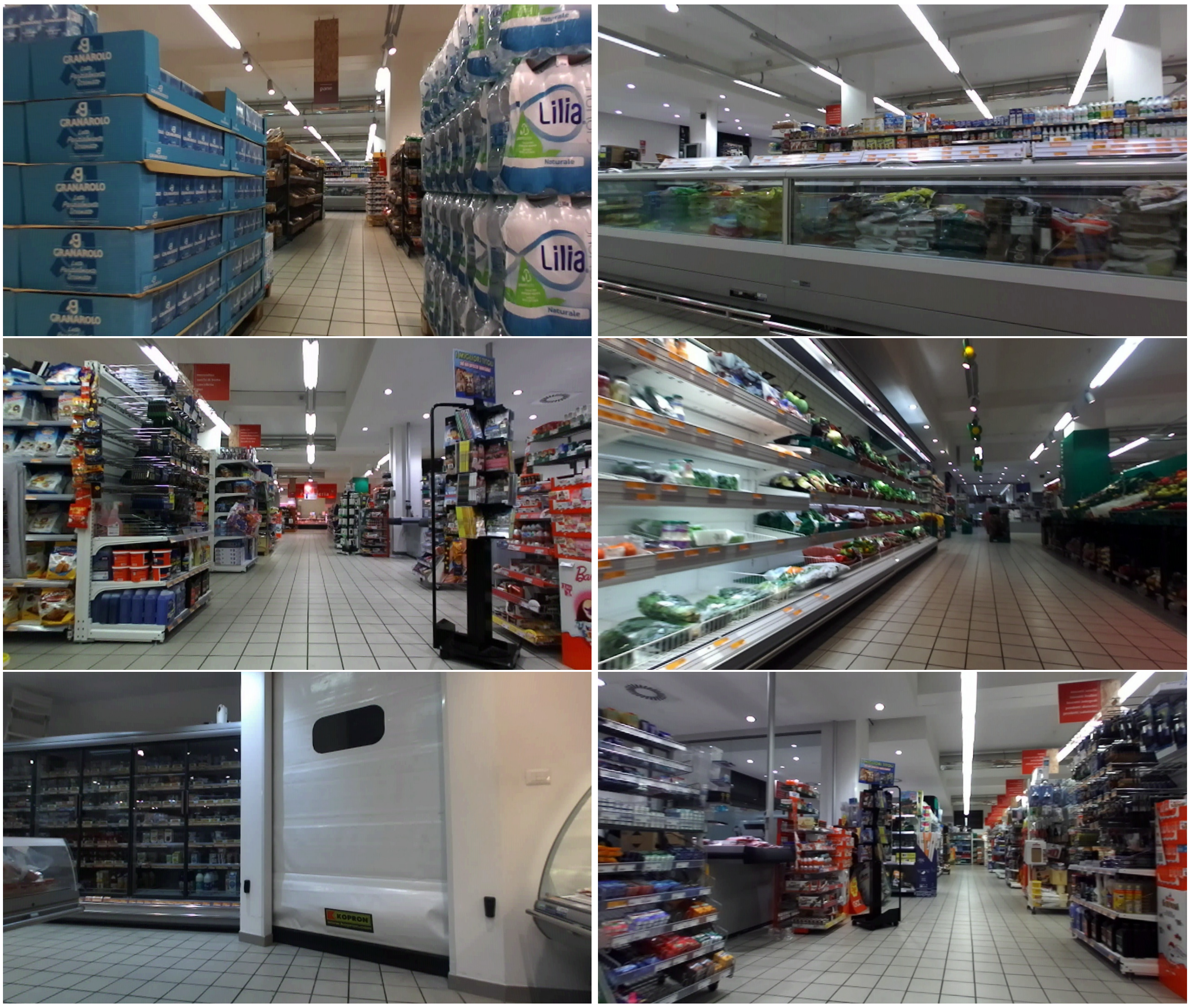


Figure 1.1: Visual variability of acquired egocentric images.

Our data analysis points out that most of the variance of the collected shopping cart positions is explained by their first two principal components. This leads us to frame the egocentric shopping cart localization problem as a three Degrees Of Freedom (3-DOF) pose estimation task. Therefore, we created a 3-DOF version of the dataset by projecting the 6-DOF poses onto a 2D plane parallel to the floor of the store. In this version of the dataset, each frame is associated with the 2D coordinates and angle describing the position and orientation of the shopping cart. Furthermore, to allow a deeper analysis of the problem, for each image of the dataset we furnished a depth image and a belonging class. The dataset was divided in 16 different classes each of them groups all the images of a convex area of the store. We decided to introduce depth image informations to analyse their usefulness to pose prediction and because several devices available on the market are now able to provide it in real time [1].
In order to deep investigate cart localization problem we benchmark two principal classes of approaches based on classification and regression.
The camera 3-DOF regression problem was investigate through two different families of methods:

- Traditional image retrieval based approaches
- Camera 3-DOF regressor-based approaches

Moreover an analysis of how much depth images can be useful to improve regression and classification performances was proposed. To examine which techniques shall be preferred depending on the computational constraints imposed by the employed hardware and by real-time requirements we proposed also a computational comparison of the different approaches.

1.3 Contributions

The main contributions of this thesis are the follow:

[1] `http:www.stereolabs.com`

- We propose a dataset to study the problem of egocentric shopping cart localization as classification and regression problem. The dataset is intended to foster research on the problem and it is publicly available at our web page[2];
- We benchmark classification, retrieval-based and regression-based localization techniques in the proposed application domain
- We propose an analysis of time performance and memory usage of best approaches
- We investigate different loss functions and architectures for CNN-based approaches
- We study the usefulness of depth information for classification and regression task in the considered context

The principal contribution of this thesis have been published in international journal and conferences:
International journal:

- E. Spera, A.Furnari, S. Battiato and G.M.Farinella. Egocart: shopping cart localization from egocentric videos.Submitted to Computer Vision and Image Understanding

International conferences:

- E.Spera,A.Furnari,S.Battiato,G.M.Farinella. Egocentric Shopping Cart Localization. In International Conference on Pattern Recognition (ICPR), 2018
- E.Spera,A.Furnari,S.Battiato,G.M.Farinella.Performance Comparison of Methods Based on Image Retrieval and Direct Regression for Egocentric Shopping Cart Localization. In 4th International Forum on Research and Technologies for Society and Industry (RTSI), 2018

[2] `http://iplab.dmi.unict.it/EgocentricShoppingCartLocalization/`

Appendix A reports a list of other works not directly related to this thesis published during my Ph.D.

The remainder of this work is organized as follows: In Section 2, we review the state of the art approaches for camera localization. In Section 3, we review the principal classic methods that we used during our study. In Section 4, we present the proposed shopping chart localization dataset. Section 5 discusses the approaches investigated in this study, whereas Section 6 discusses the results. Section 7 concludes the paper and reports insights for future research.

Chapter 2

Related Works

2.1 Localization in a retail store

Previous works have investigated the problem of localizing customers in a retail store. For instance, Contigiani et al. [5] designed a tracking system to localize customers using Ultra-Wide Band antennas installed in the store and tags placed on the shopping carts. Pierdicca et al. [6] addressed indoor localization using wireless embedded systems. Other researchers has focused on the integration of vision and radio signals to improve localization accuracy. Among those, Sturari et al. [2] proposed to fuse active radio beacon signals and RGBD data to localize and track customers in a retail store. Other researchers focused on computer vision based solutions. Liciotti et al. [7] used RGB-D cameras to monitor customers in a retail environment. Del Pizzo et al. [8] designed a system to count people from RGBD cameras mounted on the ceiling.

Differently from the aforementioned works, we consider a scenario in which shopping carts are localized relying only on images acquired from an on-board egocentric camera.

By the point of view of our research the localization of the shopping cart can be see as the camera localization task.

2.2 Image based camera localization methods

Camera localization methods are divisible in two principal families: algorithms that face the task as a classification problem and others that treat it as a regression problem. The regressive approaches are divided in two principal subfamilies: methods

based on image-retrieval and methods based on regressors.
In this section we propose an overview of works related to these different approaches.

2.2.1 Classification based methods

Classification-based approaches [22, 40, 41, 56, 54] face localization problem in a space divided in different areas and, by dividing the dataset in classes related to the different areas, tackle localization as classification problem.
These approaches aren't able to produce a fine-grade position estimation (e.g., accurate 2D or 3D coordinates) but could be the best choice in context in which a fine-grade estimation is not useful or is too hard to have.
Some of these methods are based on a BoW representation [56, 54]; differently in [41] transfer learning techniques and an entropy-based rejection algorithm have been used to employ representations based on Convolutional Neural Networks (CNN). On the other hand in [22] a CNN is trained end to end to face image geolocation problem as a classification-problem. They subdivide the surface of the earth into thousands of multi-scale geographic cells and show how their classification network outperforms classical approaches based on image-retrieval. Different classification methods [75, 76, 77, 78] use dataset of landmark building obtained through the clustering of web-photo collection. These methods normally lever on the landmark building framed to perform image retrieval approaches. Differently in [79] Support Vector Machine was trained on BoW of the different clusters associated with the landmark buildings. In Grocery context Santarcangelo et al. [40] propose a hierarchical classifier of egocentric image from a shopping cart that jointly classify action of the cart (stop and moving) and market department (e.g. Fruit,Gastronomy).

2.2.2 Regression based approaches

Unlike the classification approaches, regressive approaches try to predict accurately the 6-DOF camera pose starting by acquired image. Some of these methods are based on image retrieval techniques [46, 47]; they work by associating to a query image a set composed by the more similar images of geo-tagged training set in a particular features space and defined a specific metric. Different heuristics (e.g., k-NN approach) are finally used to estimate query image pose starting by the poses of images included in the set associated to it. Over the years, to improve these methodologies, some study focused on confusing [50] and repetitive [51] structures, or to scale to larger scenes [49], [52]. To handle large datasets, image retrieval methods that take advantage of descriptor quantization, inverted file scoring, and fast spatial matching, were proposed [45] [48] [46].

The image representation has a central role in image retrieval approaches. Some approaches encode the images using hand-crafted local features [23, 24], other use features extracted from CNNs intermediate layer. Some works use representation extracted from CNN model trained on different dataset on other task [26], other methods use representation extracted from CNN trained using the target dataset on classification or regression [28].

In [53], in order to face with the disturbing presence of repetitive structures, an automatically weight of features on the similarity score between images is proposed to reduce the impact of those related to repetitive structures and to take more the features with an unique local appearance into account.

Also Triplet and Siamese networks have been used to learn the features to address the 3D object pose estimation [32, 33], a task strongly correlated to that we are investigating in this work. In some of these works a contrastive loss [29] was used to train the network to build a features space in which similar images result clustered and dissimilar images result faraway between them [30, 31]. Some works investigate camera pose estimation in shopping small. In [81] the authors propose a methods based on Markov Random Field that, using monocular images and the shopping mall's floor-plan, jointly perform text detection, shop facades segmentation and camera pose estimation. In [80] was proposed a method based on two consecutive steps. In the first step the query image is matched by involving matching of store signs with the training set images to identify the "closest". In the second step the pose of the query image, respect to the "closest" camera reference system, is computed. Many of regression-based methods are based on a 3D model of the scene [14] [15] [37]. Associating the 3D points with one or more local descriptor, these methods build a matching between local features extracted by query image and a set of 3D points. Starting by these 2D-3D matching a query image pose is estimated using different heuristics [38] [39] [43]. To solve the time consuming problem, procured by descriptor matching task, different strategies were proposed: either searching for

the match on a subset of the 3D points [44], or based on a 3D model compression scheme [55, 79].
In the last years many works investigate CNN-based approaches that try to regress camera pose directly from images. In [11] the first end-to-end CNN based model for pose regression (POSENET) was proposed. This model based on GoogleNet architecture [42] has been obtained replacing classification layers with two fully connected layers to tackle the regression task. In [12] two different loss functions were proposed for the same architecture: one is based on trying to learn an optimal balance between position error and orientation error, the other one is based on geometric re-projection error. In [13] Long-Short-Term-Memory (LSTM) was combined with Posenet architecture for camera pose regression. The LSTM units allow to identify a more useful feature's correlations for the task of pose estimation. In [57] the authors use encoder-decoder CNN to camera pose prediction. In [58] a multi-task CNN, to deal the trade-off between orientation and position, and a data augmentation method for camera pose estimation was proposed. Even if these methods result less performing, in term of accuracy, compared to the methods based on 3D models, they are characterized by compactness and very short processing times. These characteristics make this family of methods very likable, in particular for work in embedded settings.

2.3 Dataset

In the last years different datasets were proposed in indoor and outdoor environments for camera localization task. One of the best known, for indoor context, is 7-Scenes dataset. This dataset was released in 2013 by Microsoft and formed by 7 different scenes and for each scene several sequences were provided each one consisting of 500-1000 frames. The dataset was collected using a handheld kinect RGB-D camera at 640 × 480 resolution. To obtain ground truth cameras poses, an implementation of the KinectFusion system and a dense 3D model of the scenes were used. The dataset was built extracting frames from different sequences for each scene. Each frame is formed by RGB images, depth images and positions and orientations of the cameras. Like most indoor datasets, in 7 scene dataset as well,the scenes are spanning the extension of a single room, only in the last years large scale indoor dataset was proposed. In [13], for instance, the authors propose TU Munich Large-Scale Indoor dataset and it is one of the first covering a whole building floor with a total area of 5,575 m^2. In order to generate ground truth pose information for each image, the authors captured the data using a mobile system equipped with six cameras and three laser range finders. In [82] a dataset acquired in the ground level of a shopping mall with an extention of 5,000 m^2 was proposed. The training set images of this dataset was captured using DSLR cameras while test set is composed by 2,000 cell phone photos taken by different users. To estimate ground truth camera pose 3D-2D matching algorithm was used levering on a 3D model obtained with a high precision LiDAR scanner.

Related to outdoor context, relevant datasets are Rome16k and Dubrovnik6k [79], and The Cambridge Landmarks dataset in [11]. Dubrovnik6k and Rome16k datasets were build from photos retrieved by Flick, the first is formed by 6,844 images while the second by 16,179 images. Both these datasets contain also 3D model of the scenes. The Cambridge Landmarks dataset is formed by 5 different scenes and contains 12K images with full 6-DoF camera poses. All these three outdoor datasets were generated using Structure From Motion algorithm.

In grocery context only VMBA15 dataset [40] composed by 7839 samples is available, the images are labelled according to action (i.e. stop, moving) location (indoor, outdoor) and scenes context (e.g. gastronomy, fruit) but isn't labelled in terms of 6 D.O.F. of the cameras.

Chapter 3

Background

3.1 Structure from motion

Starting from a set of images acquired in the same scene, Structure From Motion (SFM) problem consist by recovering the 3D scene and the camera 6 DOF for each image of the set.

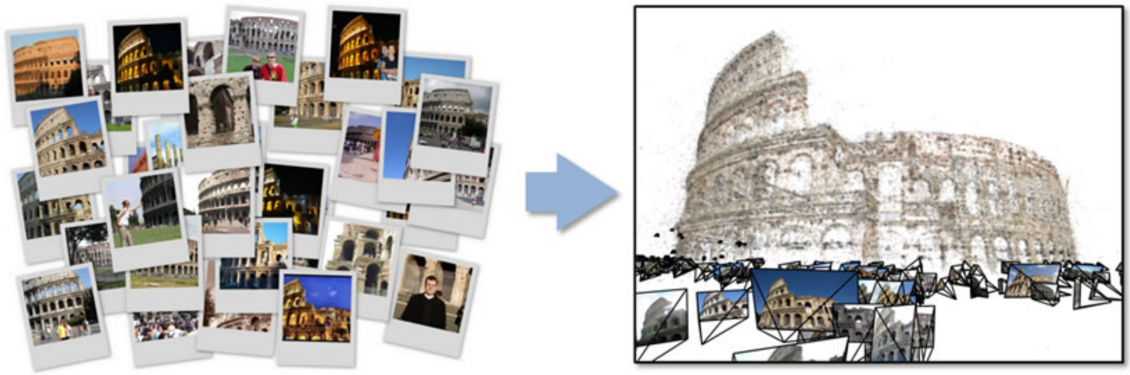


Figure 3.1: Structure From Motion aim[1]

The SFM algorithms are based on three main stage:

- By extracting image features and matching the features extracted by different frames between them
- Estimation of camera motion
- By buildind the 3D scene using the estimated motion and features

[1] image by `http://www.cad.zju.edu.cn/home/gfzhang/training/SFM/SfM.html`

3.1.1 Features and matching

Different features were proposed for SFM task. One of the most used features is the scale invariant feature transform (SIFT) [64], which has been extensively used in many of the SFM methods based on corresponding point. The SIFT features, based on local gradient histograms, result to be well performing for SMF methods because of their invariant to scaling and rotation, and their robustness as it regards illumination changes. To obtain a more compact representation in [65] PCA-SIFT features were proposed, obtained by applying principal component analysis (PCA) to the image gradients. Others features largely used in SFM algorithms are Speeded-Up Robust Features (SURF) as proposed in [66]. These features are invariant respect to scale and rotation as well and they require less computational cost for extraction compared with SIFT features. The features matching is generally performed by considering similar descriptors to be more likely matches. In many cases match correctly the features extracted by different images is a very hard task. For instance the presence in the 3D space of different objects that look similar can produce incorrect match of unrelated features and consequently major errors in camera pose estimation and 3D reconstruction. To face with ambiguity problem during features matching, different disambiguation approaches were proposed. In [67] the incorrect features matches are identified by means of relations induced by pairwise geometric transformations. Differently, in [68] disambiguation is performed by optimizing a measure of missing image projections of potential 3D structures.

3.1.2 Camera pose estimation

The first works, which investigate the theoretical opportunity to estimate camera pose using matching points, are of the early twentieth century. In [69] was proved for the first time that given two images, both framing at least the same five distinct 3D points, is possible to recover the positions of the points in the 3D space and at the same time the relative positions and orientations between the cameras up to a scale value. After many years by this first work, in [70] was showed that it's possible to estimate essential matrix of two cameras starting from eight different points correspondents just solving a linear equation. They also showed that, by mean of the decomposition of the essential matrix, is possible to obtain the relative cameras orientations and positions. The basic idea for the position and orientation camera estimation is levering on the epipolar constraints Eq. 3.1 imposed by the points matching by using the pinhole camera model (Figure 3.2).

$$\mathbf{p}_i^T \left([R_i^T(\mathbf{t}_j - \mathbf{t}_i)]_\times R_i^T R_j \right) \mathbf{p}_j =: \mathbf{p}_i^T E_{ij} \mathbf{p}_j = 0 \tag{3.1}$$

where $\mathbf{p}_i$ and $\mathbf{p}_j$ are the representation of the 3D point $\mathbf{P}$ respectively on the image planes i and j. $\mathbf{t}_i$ and $\mathbf{t}_j$ are the locations and R_i and R_j the orientation matrices respectively of the i'th and j'th camera and $E_{ij} \in \mathbb{R}^{3\times3}$ is the essential matrix

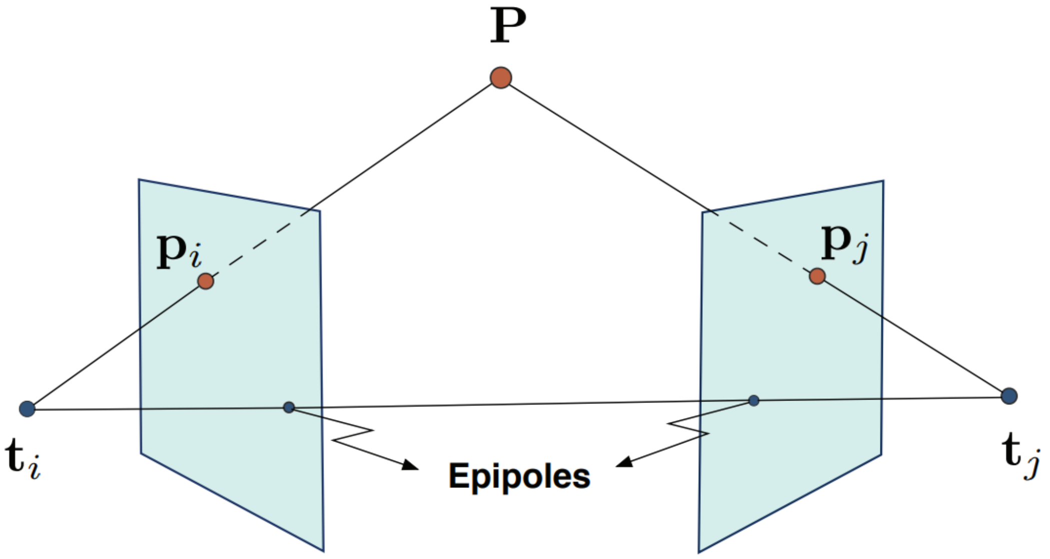


Figure 3.2: Pinhole camera model[2]

It's easy to observe that fixing a scale for the entries of E_{ij} (e.g. $\|E_{ij}\| = 1$) the 9 different elements of the essential matrix can be determined just imposing eight points matches and consequently eight epipolar constraints.

3.1.3 3D structure estimation

The methods for 3D points estimation are classically based on triangulation (Figure 3.3). Given the projection matrices of different cameras is theoretically possible

[2] image by [73]

to compute the exact 3D points position in the scene from their positions in images acquired by two or more views. Because of the noise, the back-projected rays, starting by the different centres of projection of cameras, are not generally intersected each other.

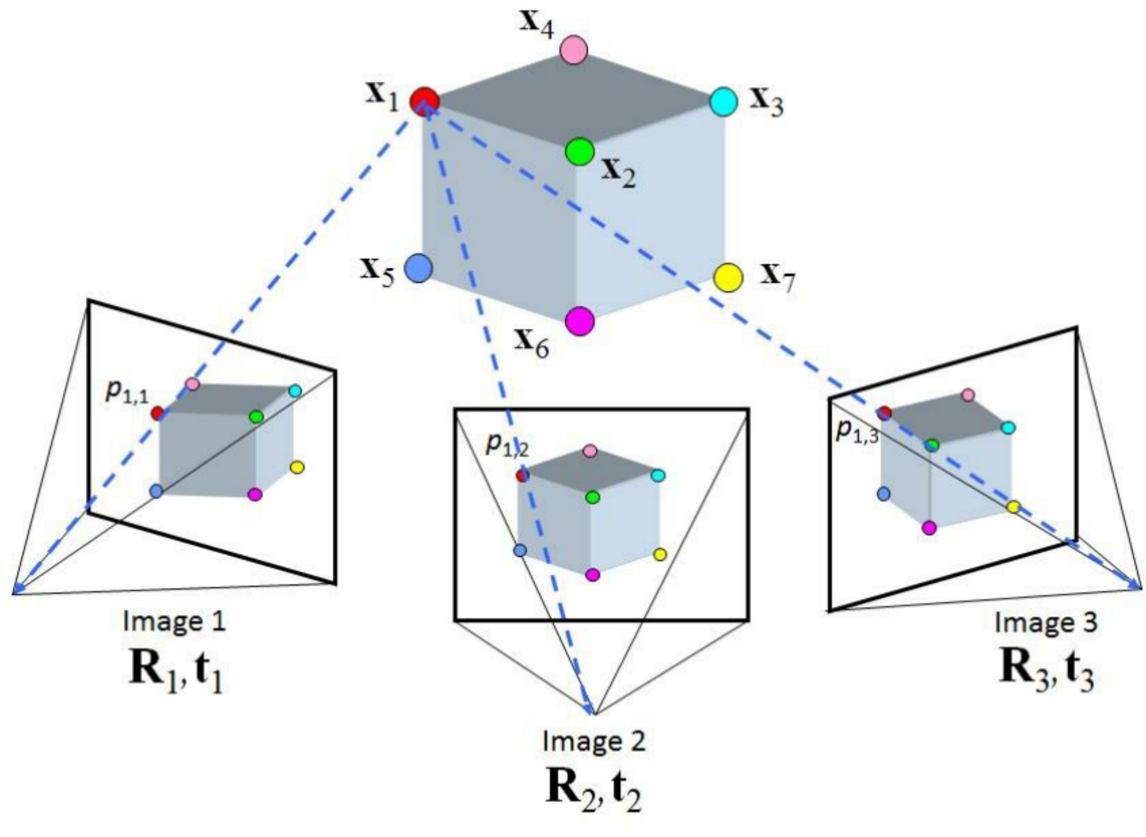


Figure 3.3: Graphical representation of triangulation procedure [3]

To find a good approximation of the 3D points locations, several methods try to minimize an appropriate error metric. Given a 3D point, the standard reconstruction algorithm identifies the 3D coordinate of the point as those that minimize the sum of squared errors between the measured pixels positions associated to the 3D point in two or more images, and the theoretical pixels positions associated to the 3D

[3] image by [74]

point, on the same images, computed by mean of projections Eq.3.2

$$P = \arg\min_{P} \sum_{i=1} \|p_i - \hat{p}_i(P)\|^2 \tag{3.2}$$

Where P is the predicted 3D point, p_i is the measured pixel position associated to 3D point in the i'th image and $\hat{p}_i(P)$ is the predicted pixel position for the same view (Figure 3.4). If the pixels positions noise is Gaussian-distributed this optimization give the maximum likelihood solution for P.

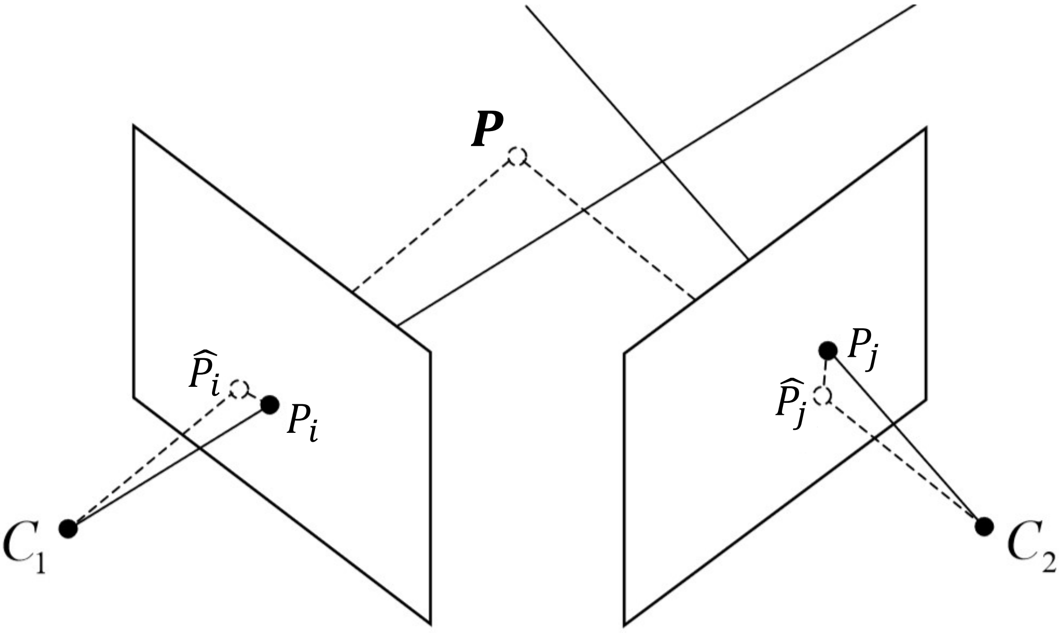


Figure 3.4: Graphical representation of minimization of the squared errors sum between measured and predicted pixel positions during triangulation

To face with SFM problem for an arbitrary number of view, two different approaches types were proposed: the sequential and the factorization algorithms. The sequential approaches are those working adding a different view one at time to the scene. These algorithms typically produce a scene initialization by computing camera orientation and 3D points cloud for the first two views. For any other image added to the scene a partial reconstruction is performed, by computing the positions of 3D points through triangulation. Different approaches were used to register new views to the scene, some of them levering on the two-view epipolar geometry to estimate position and orientation of the new camera starting by those of its predecessor. Other methods use the 3D-2D correspondents between the already reconstructed 3D points and the features extracted from the new image to determine its pose. In fact, it is possible to prove that through only 6 3D-2D matches the camera pose can be determinate. Other sequential SFM algorithms work by merging partial reconstructions related to different subset of views by using 3D points correspondents.

Differently from sequential approaches, factorization methods work computing 3D points cloud and cameras poses by using all the images simultaneously. This family of methods, introduced in [71], is generally based on direct SVD factorization of a measurement matrix composed by the measurements of the 3D points by the different cameras. These algorithms, compared to sequential methods, achieve a more evenly distributed reconstruction error across all measurements, but they fail for some structure and motion configuration.

Obtained a initial estimation of 3D points and of cameras poses a refinement process of these estimations are usually conducted using bundle adjustment techniques. Bundle adjustment works with an iterative non linear optimization to minimize a cost function related to a weighted sum of squared re-projection errors.

Bundle adjustment procedures try to determine an optimal set of parameters δ not directly measurable (cameras projection matrices, 3D points coordinates) for a set of noisy observations (e.g. pixel position associated to 3D points). Given a set of measurements M_i and the set of δ-dependent associated estimations, the features prediction errors $M_i(\delta)$ are defined as:

$$\mathbf{\Delta M}_i(\delta) =: \mathbf{M}_i - \mathbf{M}_i(\delta) \tag{3.3}$$

Bundle adjustment produces a minimization of a cost function depending of the likelihood of the features prediction errors. Assuming a Gaussian-distribution of the noise associated with the measurements, a typical appropriate cost function is:

$$f(\delta) = \frac{1}{2}\sum_i \mathbf{\Delta M_i}(\delta)^T \mathbf{W}_i \mathbf{M}_i(\delta) \tag{3.4}$$

where $\mathbf{W}_i$ is the matrix that approximate the inverse covariance matrix of the noise associated with the measurement $\mathbf{M}_i$. To optimize the cost function during bundle adjustment procedure several optimization methods were used and three main categories were strongly investigated during the years:

- the second-order Newton-style methods
- first order methods
- the sequential methods incorporating a series of observations one-by-one

A deep analysis of these methods was proposed in [72].

3.1.4 SAMANTHA

In this section we will describe the SFM algorithm [17, 16] used to obtain poses labels for the images of our dataset. This algorithm is based on a reconstruction process levering on a binary tree built through a hierarchical cluster of the images set. Each image corresponds to a leaf of the tree while the internal nodes are associated to a partial reconstruction of the model obtained by merging the partial models associated to the two sub-nodes. The first step of SAMANTHA algorithm is to perform the extraction of features based on difference of Gaussian with radial descriptor. The features matching is performed using nearest neighbour approach and different heuristics are sequentially implemented to maintain only the more significant matches. Given the features matching, an image affinity measure is used, in agglomerative clustering algorithm, to build the hierarchical cluster tree. The image affinity measure used takes in account the number of features matching between images and how much the features are spread in the images. With a bottom-up procedure the agglomerative clustering algorithm, starting from clusters formed by single images, merge iteratively the clusters with the smallest cardinality (sum of the views belonging to the two clusters) among the n closest pairs of clusters. The simple linkage rule is used to measure the distance between the different clusters. By exploiting the cardinality of the clusters during agglomerative clustering procedure, the algorithm is able to produce a more balancing hierarchical cluster tree (Figure 3.5) and consequently a reduction of time complexity [16].

[4] image by [16]

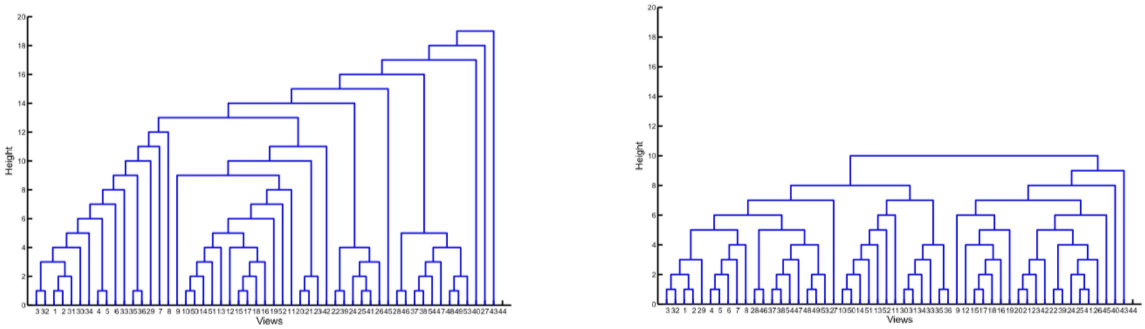


Figure 3.5: Example of hierarchical cluster tree produced merging the closest clusters using simple linkage role (left) and the more balanced tree obtained merging the clusters with the smallest cardinality among the n closest pairs [4]

Computed this hierarchical organization of the images the scene reconstruction is implemented. During this process three different operations are involved: the two views reconstruction (to merge two different views), a resection-intersection step to add a single view to the model and the fusion of two partial models (Figure 3.6).

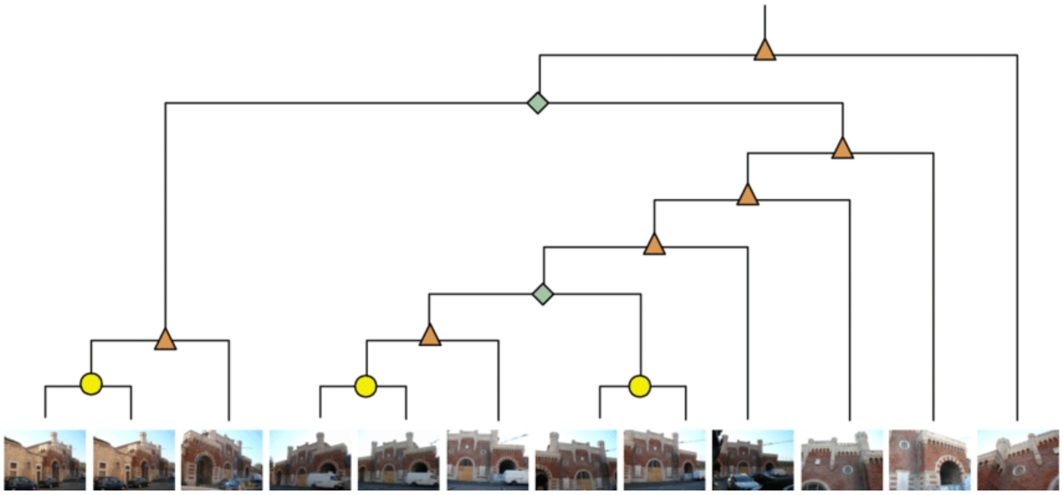


Figure 3.6: Example of hierarchical cluster tree in which on each internal node is associated the relative reconstruction operation. The circle corresponds to the creation of a stereo-model, the triangle corresponds to a resection-intersection, the diamond corresponds to a fusion of two partial independent models. [5]

[5] image by [25]

3.2 Support Vector Regression

The Support Vector Regression is a generalization of Support Vector Machine for regression task. Suppose to have a training set $(x_1, y_1), (x_2, y_2), ...(x_n, y_n)$, with $x_i \in X$ where X denotes the space of the input patterns and $y_i \in \mathbb{R}$, the Support Vector Regression method try to find a function $f(x)$ that have at most a distance ϵ from all the target y_i and is as flat as possible. This method therefore do not care about errors smaller than ϵ and optimize the parameters of $f(x)$ considering the prediction error bigger than ϵ. In this algorithm a central role is played by the choice of $f(x)$ function (e.g. linear, Polynomial). By using a linear function Eq.3.5

$$f(x) = \langle w, x \rangle + b \tag{3.5}$$

with $w \in X$ and $b \in \mathbb{R}$ is possible to write the regression problem as the minimization problem of the Soft Margin Loss functon Eq.3.6 [63]

$$minimize \quad \frac{1}{2}\|w\|^2 + C\sum_{1}^{n}(\delta_i + \delta_i^*) \tag{3.6}$$

subject to the follow constraints:

$$\begin{aligned} y_i - \langle w, x_i \rangle - b &\leqslant \epsilon + \delta_i \\ \langle w, x_i \rangle + b - y_i &\leqslant \epsilon + \delta_i^* \\ \delta_i, \delta_i^* &\geqslant 0 \end{aligned} \tag{3.7}$$

where δ_i and δ_i^* are variables that represent how much the target i is far from the area around the regression function, identified by the margin ϵ (Figure 3.7). The variables aforementioned are defined as follows:

$$\delta_\epsilon := \begin{cases} 0 & if \quad \delta \leqslant \epsilon \\ \|\delta - \epsilon\| & otherwise \end{cases} \tag{3.8}$$

where $\delta =\|y_i - f(x_i)\|$, $\delta_i = \delta_\epsilon$ if $y_i > f(x_i)$ and $\delta_i^* = \delta_\epsilon$ otherwise.
The constant $C > 0$ in 3.6, fixes a trade-off between the flatness of f and the amount of tolerated deviations larger than ϵ.

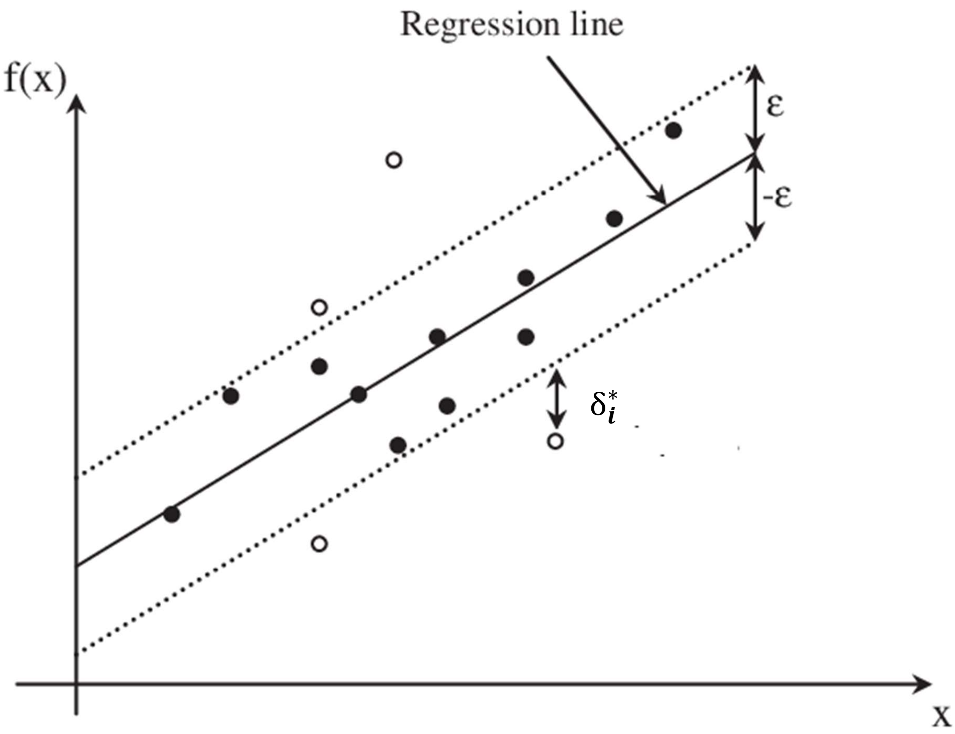


Figure 3.7: Soft margin loss for linear SVR

The minimization problem 3.6 can be solved using its dual formulation obtained through the Lagrangian function L:

$$
\begin{aligned}
L := \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\delta_i + \delta_i^*) - \sum_{i=1}^{n}(\eta_i\delta_i + \eta_i^*\delta_i^*) + \\
-\sum_{i=1}^{n}\alpha_i(\epsilon + \delta_i - y_i + \langle w, x_i\rangle + b) + \\
-\sum_{i=1}^{n}\alpha_i^*(\epsilon + \delta_i^* + y_i - \langle w, x_i\rangle - b)
\end{aligned}
\tag{3.9}
$$

where the Lagrange multipliers α_i, α_i^*, η_i and η_i^* have to satisfy the follow constraint:

$$\alpha_i, \alpha_i^*, \eta_i, \eta_i^* > 0 \tag{3.10}$$

Imposing equal to zero the partial derivatives of L, with respect to the primal variables (w, b,....), it's possible rewrite the equation 3.9 as the follow dual optimization problem:

$$maximize \begin{cases} -\frac{1}{2}\sum\limits_{i,j=1}^{n}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\langle x_i, x_j \rangle \\ -\epsilon\sum\limits_{i=1}^{n}(\alpha_i + \alpha_i^*) + \sum\limits_{i=1}^{n} y_i(\alpha_i + \alpha_i^*) \end{cases} \tag{3.11}$$

subjected to:

$$\sum_{i=1}^{n}(\alpha_i - \alpha_i^*) = 0 \quad and \quad \alpha_i, \alpha_i^* \in [0, C] \tag{3.12}$$

by levering the conditions imposed on partial derivatives the function $f(x)$ can be expressed as follow:

$$f(x) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\langle x_i, x \rangle + b \tag{3.13}$$

This formulation allows to evaluate $f(x)$ in terms of dot products between the data without compute explicitly w. Different optimization methods can be used to compute the b variable (e.g. using KKT conditions, interior point optimization method).

The typical approach to make SVR algorithm able to regress a non linear function consist to map the input onto a m-dimensional features space, by using some fixed (non linear) mapping, and then by applying the standard SVR algorithm to build a linear model in this feature space. Fixed a mapping function γ and defined the Kernel function K as dot product in the mapping space:

$$K(x, x_i) = \langle \gamma(x_i), \gamma(x) \rangle \tag{3.14}$$

the linear regressive function in the feature space can be expressed as folow:

$$f(x) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) K(x_i, x) + b \tag{3.15}$$

Some of the kernel functions most commonly used are the Polynomial function Eq.3.16 and the Radial basis function Eq.3.17

$$K(x, x_i) = (\langle \gamma(x), \gamma(x_i) \rangle + C)^d \tag{3.16}$$

$$K(x, x_i) = exp\left(- \frac{\|x - x_i\|^2}{2\sigma^2} \right) \tag{3.17}$$

Where d is the degree of the polynomial while σ is a free parameter.
As can be observed in Figure 3.8 the ability of SVR algorithm to perform a good regression strongly depend on the kernel function used.

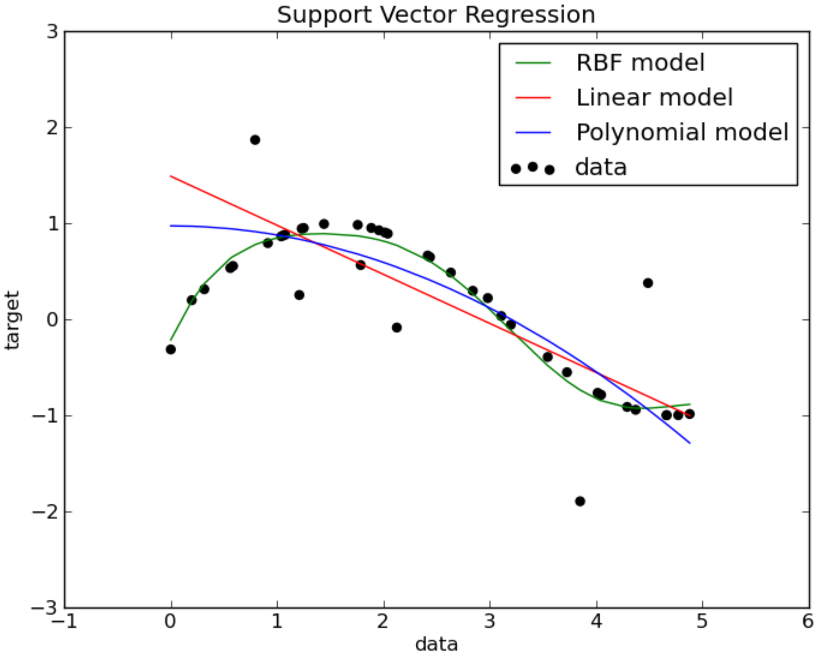


Figure 3.8: Sample of SVR regression curve obtained with different kernel on toy 1D data.[6]

[6] image by `http://scikit-learn.org/stable/auto_examples/svm/plot_svm_regression.html`

3.3 K-NN regression

K nearest neighbours is a simple and classical algorithm for the variable's regression. Given a query example and fixed a value of K, the basic idea of K-NN approach is to associate to the query example the average of the K nearest neighbours examples in the representation space. The average can be weighted with a multiplicative factor inversely proportional to the distance between query example and neighbours in the representation space. The choice of the distance used and of the K value have a central role for the algorithm performance. Classically, euclidean, cosine or Manhattan distances have been largely used for K-NN approach. The K value choice is frequently done through cross validation approach.

3.4 Improved Fisher Vector

Fisher Vector [19] is a global image descriptor obtained by pooling local image features. It works capturing the average of the differences, of first and second order, between the images descriptors and the centres of the Gaussian Mixed Model (GMM) that fits the distribution of the descriptors of the whole dataset. This representation was strongly used for image classification task. The procedure to build a Fisher Vector representation consist of different phases:

- extract a set of descriptors $\vec{x}_1$,...,$\vec{x}_N$ (e.g. sift) from each image
- learn a GMM fitting the distribution of the descriptors
- compute a soft assignments of each descriptor x_i to the K Gaussian components given by the posterior probability:

$$q_{ik} = \frac{exp[-1/2(\vec{x}_i - \vec{\mu}_k)^T \sum_k^{-1}(\vec{x}_i - \vec{\mu}_k)]}{\sum\limits_1^K exp[-1/2(\vec{x}_i - \vec{\mu}_k)^T \sum_k^{-1}(\vec{x}_i - \vec{\mu}_k)]} \tag{3.18}$$

- given the set of descriptors x_1,...,x_N of an image, for each k=1,...,K, compute the mean and variance deviation vectors

$$u_{jk} = \frac{1}{N\sqrt{\pi_k}} \sum_{i=1}^{N} q_{ik} \frac{x_{ji} - \mu_{j,i}}{\delta_{jk}} \tag{3.19}$$

$$v_{jk} = \frac{1}{N\sqrt{2\pi_k}} \sum_{i=1}^{N} q_{ik} \left[\left(\frac{x_{ji} - \mu_{j,i}}{\delta_{jk}} \right)^2 - 1 \right] \tag{3.20}$$

- build Fisher vector for the query image as concatenation of u_k and v_k for all GMM components:

$$FV = [\vec{u}_1, \vec{v}_1, ..., \vec{u}_k, \vec{v}_k] \tag{3.21}$$

The Improved Fisher Vector add other two components to classical Fisher Vector: the use Helling's kernel (or other non-linear additive kernel) and the normalization of the Fiher Vector through the l^2 norm. A modified version of Improved Fisher Vector is the spatially enhanced Improved Fisher Vector, it is obtained appending to the local descriptors $\vec{x}_i$ their normalised spatial coordinates (w_i, h_i) in the image before the quantization with the GMM as show below:

$$\vec{x}_i^{SE} = \left[\vec{x}_i^T, \frac{w_i}{W} - 0.5, \frac{h_i}{H} - 0.5 \right]^T \tag{3.22}$$

where $W \times H$ are the dimensions of the image.

3.5 Siamese and Triplet networks

In the last years Siamese and Triplet architecture was used in computer vision for different tasks as classification or 3D object pose estimation. A Siamese network consist of two networks, sharing the same weights, that are trained with couple of images labelled as similar or dissimilar. This type of network (Figure 3.9) can be trained with contrastive loss Eq.3.23 on embedding space with the aim to minimize

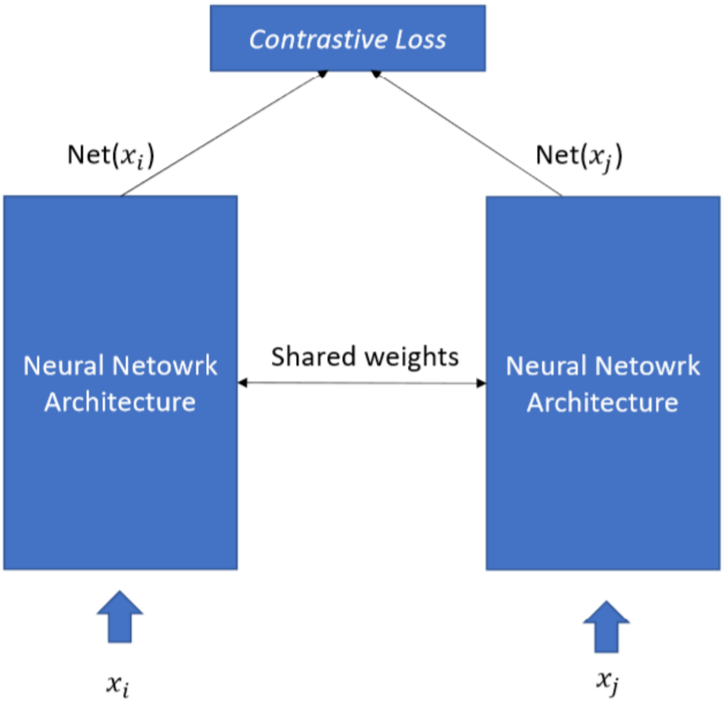


Figure 3.9: Typical siamese network architecture using contrastive loss

distance between similar samples and maximize distance between dissimilar images in the representation space.

$$\begin{aligned} Contrastive \quad Loss() = 1/2 * \delta(y_i, y_j) * (\|Net(x_i) - Net(x_j)\|_2) + \\ +1/2 * (1 - \delta(y_i, y_j)) * (\|Net(x_i) - Net(x_j)\|_2) \end{aligned} \tag{3.23}$$

Where $\delta(.)$ denotes the Dirac delta function, y_i and y_j are the labels associated to the frame x_i and x_j, $Net(x)$ is the embedding representation space produced by the network for the image x. Another typical loss function used to train Siamese network is the pairwise similarity loss:

$$\begin{aligned} Pairwise \quad Similarity \quad Loss() = \delta(y_i, y_j) * (1/k + Net(x_i, x_j)) + \\ +(1 - \delta(y_i, y_j)) * Net(x_i, x_j) \end{aligned} \tag{3.24}$$

where $Net(x_i, x_j)$ is the pairwise similarity score of the network (Figure 3.10).

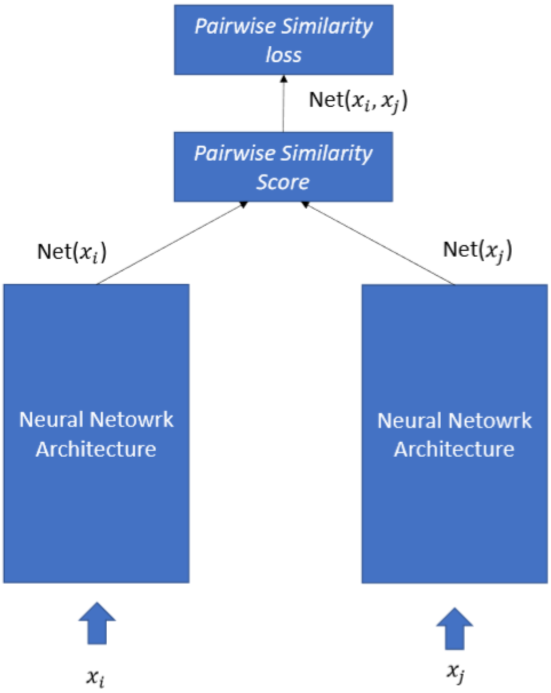

Pairwise Similarity loss
Net(x_i, x_j)
Pairwise Similarity Score
Net(x_i)
Net(x_j)
Neural Netowrk Architecture
Neural Netowrk Architecture
x_i
x_j

Figure 3.10: Typical siamese network architecture using pairwise similarity loss

The Siamese networks has been extended as triplet networks formed by 3 instances of the same feedforward network with shared parameters (Figure 3.11). This architecture during training take 3 input images, an anchor image denoted with x, a positive sample similar to the anchor sample denoted with x^+ and a negative sample dissimilar to the anchor sample denoted with x^-. When fed with the samples, the network outputs the distances between anchor sample representation and the representations of positive and negative samples in the embedding space.

This architecture is typically trained to separate similar samples by dissimilar in embedding space of a margin m (Figure 3.12) using the following Triplet loss Eq.3.25:

$$TripletLoss() = max(d(Net(x^+), Net(x)) - d(Net(x^-), Net(x)) + m, 0) \quad (3.25)$$

where d is a distance defined in embedding space.

Typically, Triple and Siamese networks include a large number of parameters and,

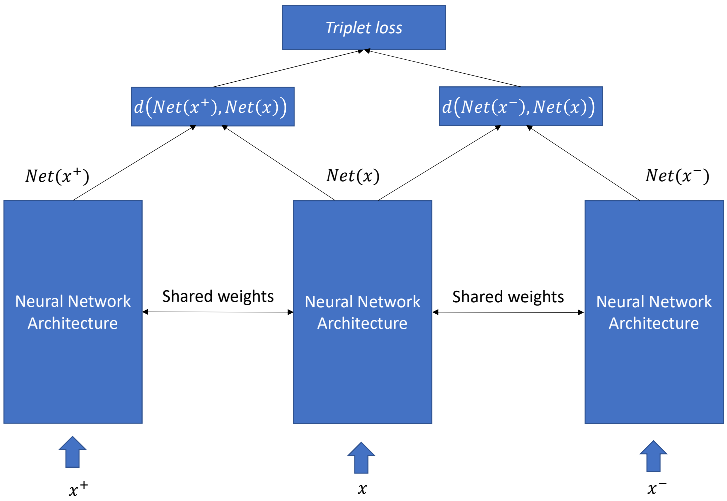


Figure 3.11: Typical siamese network architecture using pairwise similarity loss

by using typical Siamese and Triplet losses, most of the pairs or triplets samples produce a small or non-existent networks weights update during the training. Due to these two undesirable characteristics a huge number of pairs or triplets of samples must be processed to obtain a robust model. Moreover, sampling all possible pairs or triplets, as the size of the training dataset increases, can quickly become intractable and produce very slow convergence of the models. To face with this sampling problem, different heuristics was proposed in the last years. Some works prose a smart sampling strategy [59], by selecting pairs or triplet samples to avoid the useless samples for the training and to focus on the samples that show the most contradictory representations. Other works attach the problem by proposing global loss function to train the network [60] .

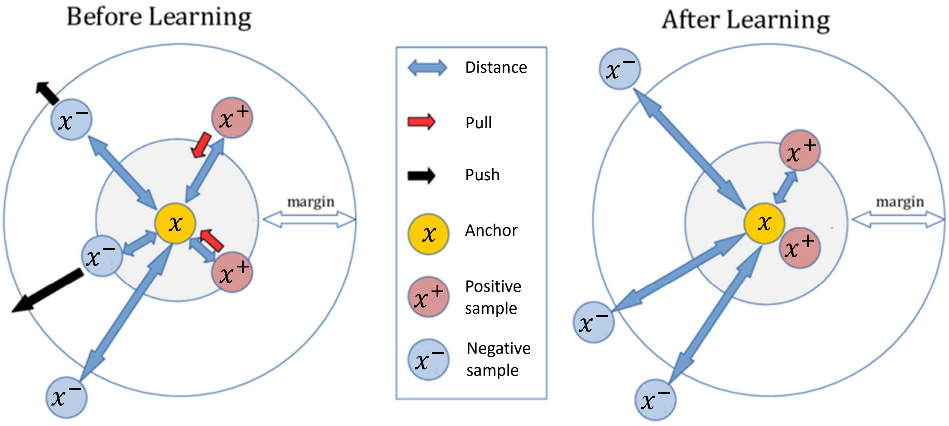


Figure 3.12: Graphical representation of how Triplet loss works in embedding space

Chapter 4

EgoCart dataset

In this section we introduce a large-scale indoor dataset of geo-localized images in grocery context proposed to address the shopping cart localization problem through egocentric images. Usually to build an accurate dataset for camera localization task, using only cameras, is necessary to acquire a huge quantity of images (normally through the acquisition of continuous videos) and, using SFM algorithms, infer at the same time a 3D model and image poses [11]. The images are in this way labelled with 6-DOF camera poses using 3D coordinates for the positions labelling and quaternions, Euler angles or other angular representation [58] for the camera orientations. In accordance with [13], we observe how this procedure become very difficult to apply in the context we are analysing, due to two principal reasons:

- The presence of repetitive structure elements (e.g, shelves, products, doors, check-out) that tend to create ambiguity.
- The big dimension of the environment that implies the need of a big number of images, and consequently a high computational cost, to build an accurate 3d model and an accurate poses estimation.

The datasets proposed for camera localization in indoor context are mostly related to small space with extension of a single room and only few dataset was proposed for camera localization in large scale indoor environments. For the complexity to apply standard procedure to build this type of dataset in large scale indoor environments, some time other sensors were used to simplify the dataset collection. In [13], for instance, the dataset was collected using a system composed by six high resolution cameras and three laser range finders.

To address the hard task of build a dataset for camera localization in our setting and to maintain a lower computational cost, we perform SFM algorithms on subset of images, extracted from the different videos, by building different 3D models related to parts of the store partially overlapping between them and with some images present in more than on subsets of the whole dataset. By taking advantage of the presence of the same images placed in the different 3D models we register them together in order to have an overall 3D model and all the frames in placed in the same reference system. The proposed dataset collects RGB images and the depth images associated (Figure 4.1) extracted from nine different videos acquired with the left cameras of two zed-cameras [1] mounted on a shopping cart. The depth images have been computed using the zed camera API. The cameras was positioned with focal axis parallel between them and to the store floor looking toward the travel direction of shopping cart (Figure 4.2).

The video frames were extracted with a frame rate of 3 fps and the SFM algorithm to estimate the camera position and orientation was performed using using SAMANTHA algorithm implemented on 3D ZEPHIR software [17, 16]. The dataset was collected in a store with extension of 782 m^2 during closing time. The dataset is formed by $19,531$ couples of RGB images and depth images divided in train and test set. These two set are obtained selecting images extracted from six videos for training set ($13,360$ frames) and images from the remaining three videos for test set ($6,171$ frames). Both training and test set contain images covering the entire store. Moreover the dataset was divided in 16 different classes each of them is related to a specific part of the store (e.g. corridors, fruit area) (Figure 4.4). The images was therefore labelled with their pose coordinate and with the id of belonging class. Figure 4.3 shows confounding pairs of images for pose regression task, couples of frames with high visual similarity and very dissimilar position and/or orientation and images acquired in the same position but with low visual similarity due to the different orientation of the cameras, that characterize the proposed dataset.

4.0.1 3-DOF labels

Due to the acquisition setting (cameras fixed to the shopping cart with focal axis with direction and verse concordant with shopping cart displacement vector) the

[1] http://www.stereolabs.com

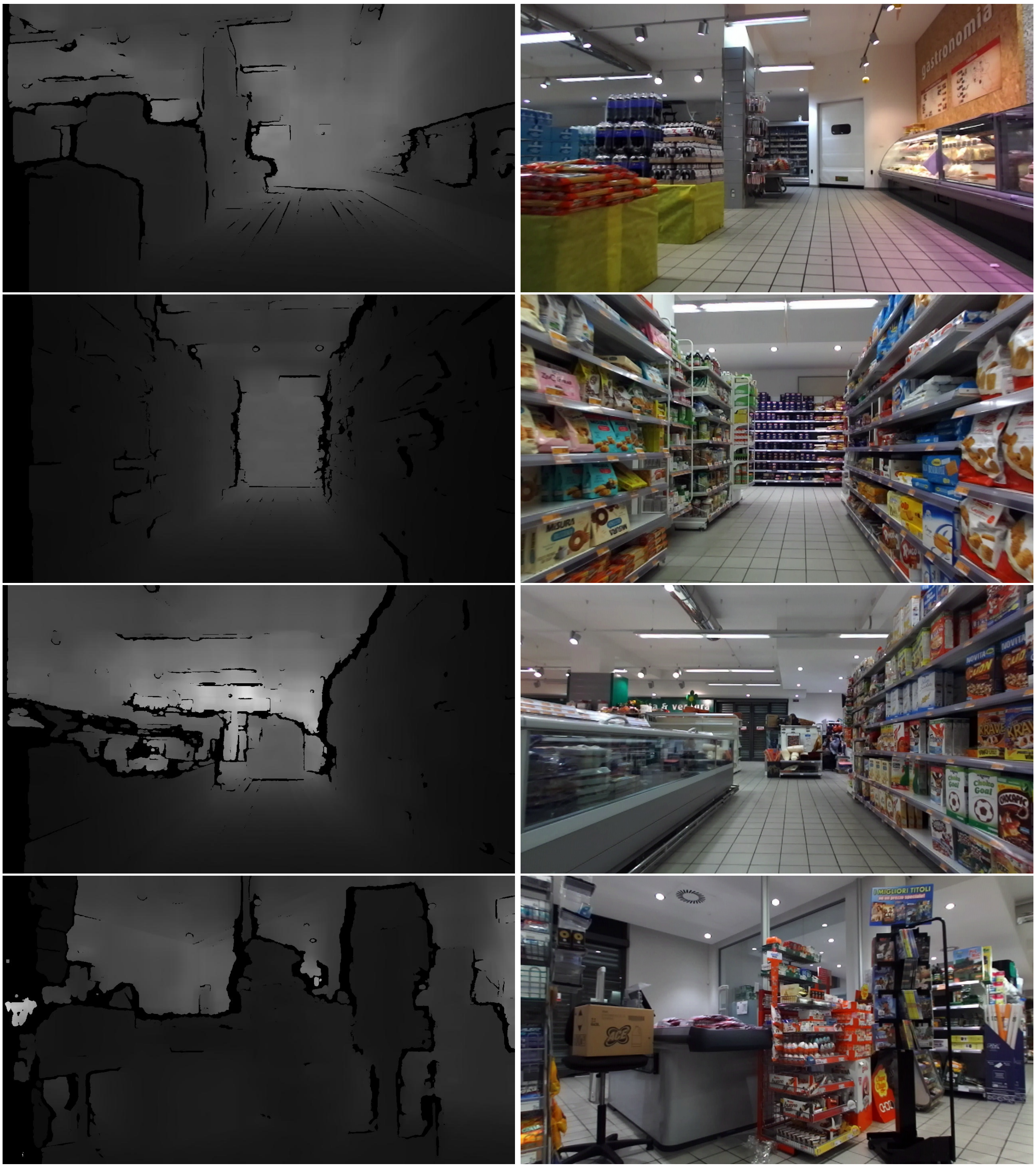


Figure 4.1: Samples of RGB images and depth images associated of our dataset

camera poses of the proposed dataset are limited to have 3 degrees of freedom. Two identifying the position and one identifying the orientation on a 2D plane parallel to the floor of the store. Applying the Principal Component Analysis (PCA) on 3D

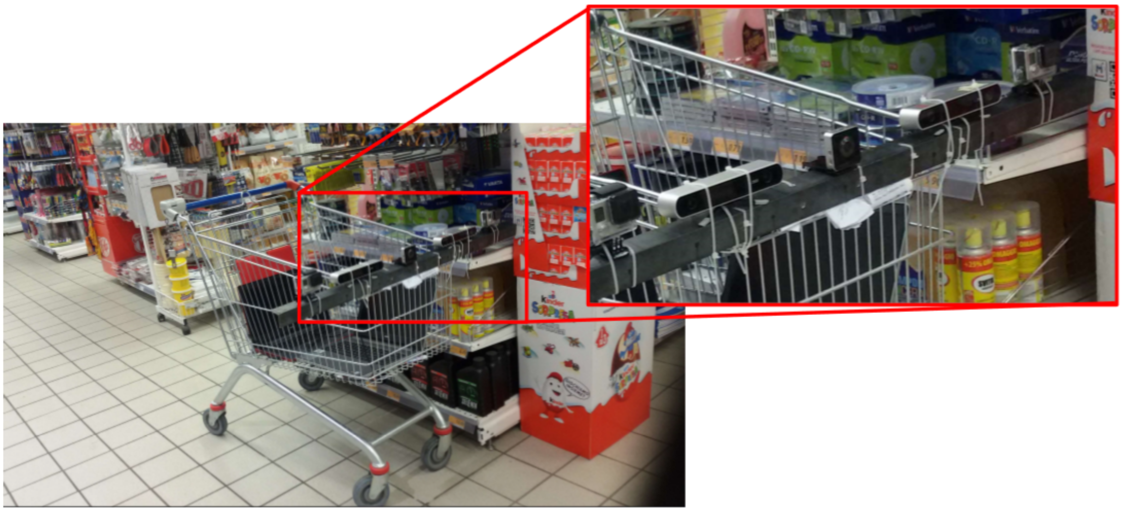


Figure 4.2: The hardware setup employed to collect the dataset using shopping carts

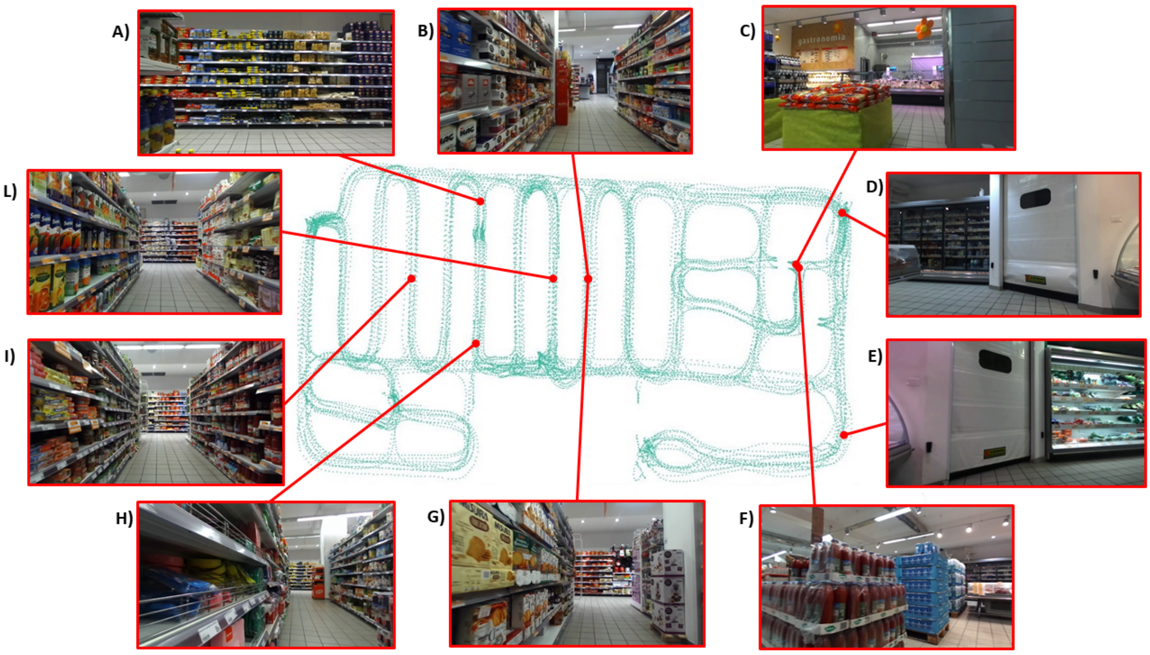


Figure 4.3: Confounding couples of frames for pose regression task: A) and H) images that frame the same shelf at different scale, B) and G) frames in the same corridor with opposite direction, C) and F) frames with same position but different orientation, D) and E) images, with different positions, frame similar structure, L) and I) images of two different corridors with high visual similarity

positions, obtained through SFM algorithm, of the images of our dataset is possible to observe that more than the 99.99% of the whole variance appertain to the first two principal components. These two components represent a reference system for the plane in witch the cameras moved during the acquisitions. By projecting all the 3D coordinates and the orientation vectors on these two component we obtain a 2D representation of the poses of the images of our dataset. In Figure 4.4 are showed the 2D coordinate of the images in the store. We take in consideration this 2D representation of our data considering it the most pertinent given the application domain characteristics. Specifically, we represent the shopping cart poses through two 2D vectors, one representing the position $\mathbf{p} = (x, y)$ and the other, with unitary length, representing orientation $\mathbf{o} = (u, v)$ of the cart. We represent the direction of the shopping cart with a 2D unitary vector rather than with a more compact scalar values, by expressing the angle in radians or degree, to preserve the increasing monotony of the relation between the distance between 2 different orientations and numerical distance between their representations. By using, for instance, scalar representation that express the angle in degree in the interval [-180,180], between a fixed vector and the direction vector of the shopping cart, we would have represent faraway between them two cameras with similar direction if their labelling are respectively near to the maximum and the minimum of the representation range (e.g. the directions corresponding to -179° and 179° differ between them by only 2° but the distance between their representation is of 364°) and more near two cameras with directions less similar (e.g the directions corresponding to -90° and 90° are 180° distant and their representation distance is also of 180°). Our choice of directions representation was therefore guided to avoid this counter-productive characterization.

4.0.2 Classification labels

In the stores, that are typically organized in department, also a rough localization of the cart could be very useful to analyse how the costumers move between the different departments. This type of analysis could have a central role to reorganize departments location in costumer-friendly manner. To analyse the image-based place recognition task in grocery context, we partitioned the store surface in 16 different convex areas and divided the dataset in 16 different classes each one gather all the images of a specific area. Fourteen of the classes are associated with the same amount of corridors, one is related to an open space and the last one is associated to a marginal area of the store composed by some shortest corridors. In Figure 4.4 a graphical representation of dataset subdivision is reported.

4.0.3 Error analysis

To have a qualitative reference point to evaluate the performances of the image-retrieval based methods that we benchmark for our dataset we compute the minimum error achievable with an image-retrieval approach for localization task on the proposed dataset. To compute the minimum error at each frame of the test set we associate the training set image nearest in the position-orientation 3D space. Due to the different measure units, meters for the 2D subspace associated with position and degrees for the 1D subspace associated with orientation, the identification of the nearest frame of the training set to a query image is possible only fixed an equivalence between a distance in the position space and a distance in the orientation space (e.g. 1m is equivalent to 10°). We fixed implicitly this equivalence using as metric a weighed sum of the two distances. Given two 3-DOF poses p_i and p_j, we define the following parametric distance measure:

$$d(p_i, p_j; \alpha) = \alpha \cdot d_p(p_i, p_j) + (1 - \alpha) \cdot d_o(p_i, p_j) \tag{4.1}$$

where $d_p(p_i, p_j)$ represents the Euclidean distance between the positions of the poses p_i and p_j, $d_o(p_i, p_j)$ represents the angular distance between the orientations of the poses p_i and p_j, and α is a parameter that define the weights associated with position and orientation distances. By choosing a specific value for α, we determinate a particular weights for the two distances summed in and consequently a specific equivalence between the distances in position space and the distances in orientation space and, fixed it, a well determined proximity measure between cameras. Fixed α and given a test image s_i with ground truth pose p_i, optimal nearest neighbour search is realize associating to s_i the training s_j with pose p_j such that $d(p_i, p_j; \alpha)$ is minimized. To measure the minimum errors achievable with image-retrieval approach we compute the error on position and orientation separately for α varying between 0 and 1.

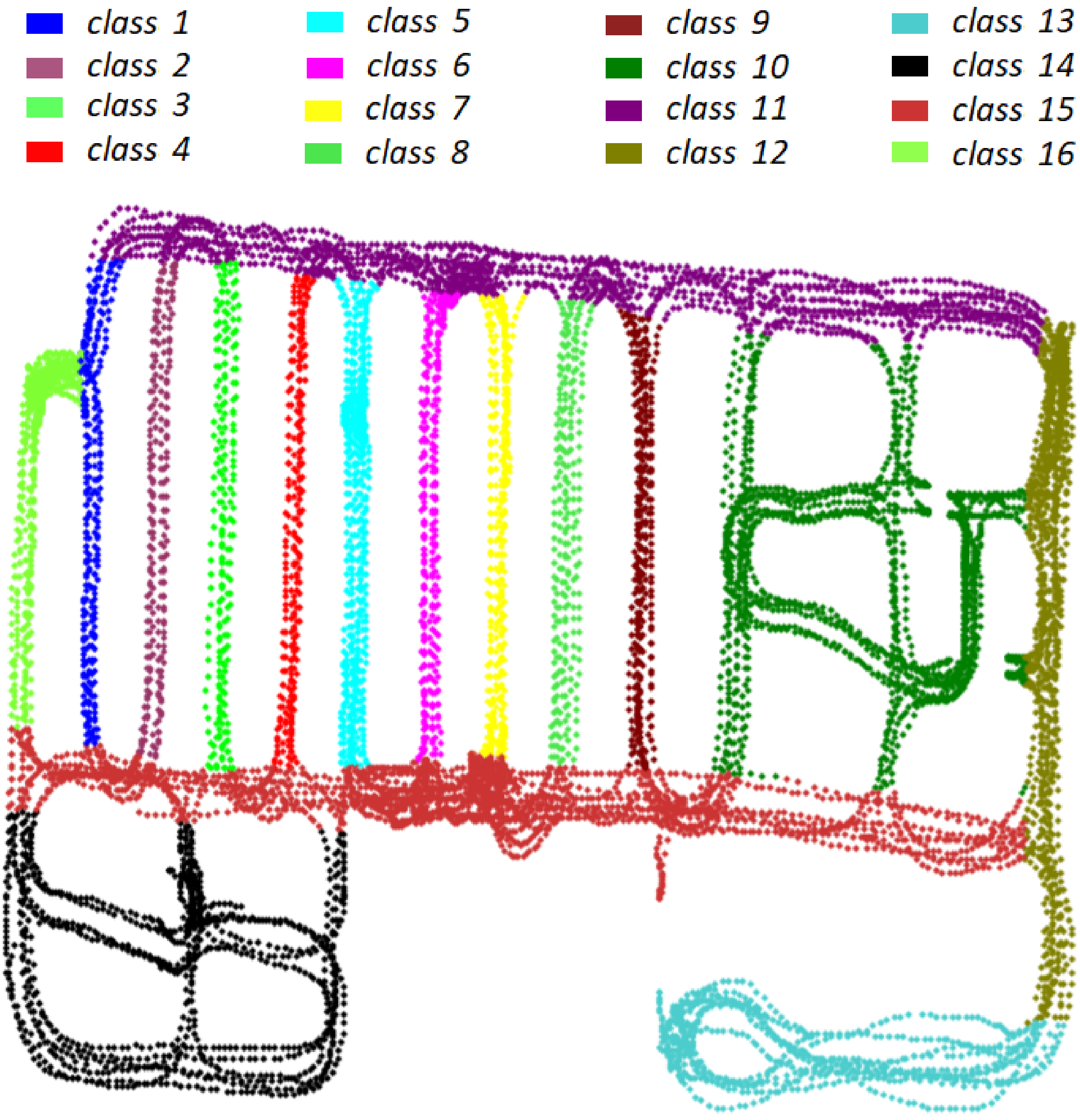


Figure 4.4: Training set divided in classes. The 2D locations of the cameras are plotted, images belong to the same class are plotted with the same colour

	Mean		Median	
α	P.E.(m)	O.E.(°)	P.E.(m)	O.E.(°)
0	9.89	0.54	9.00	0.31
0.1	0.32	1.73	0.27	1.34
0.2	0.25	2.48	0.21	1.87
0.3	0.21	3.20	0.18	2.35
0.4	0.18	4.03	0.16	2.84
0.5	0.16	4.99	0.14	3.44
0.6	0.14	6.31	0.12	4.22
0.7	0.12	7.98	0.11	5.28
0.8	0.11	10.47	0.10	6.63
0.9	0.09	17.31	0.08	10.08
1	0.05	90.45	0.04	90.47

Table 4.1: Mean and median position and orientation lower-bound errors obtained with optimal nearest neighbour search.

Table 4.1 reports the mean and median values of the Position Errors (P.E.) and of the Orientation Errors (O.E.) computed over the whole test set varying α on the parametric-distance defined above. For $\alpha = 0$ the weigh associated with position distance is 0 and the one associated with orientation distance is equal to 1 consequently the search of the nearest frame of the training set for each test image is determined exclusively only through the orientation. For this α value, we obtain a largest lower-bound position error of 9.89 m and a smallest orientation error of 0.54°. As the value of α increases the position distance becoming increasingly important in determining which of the training images is the closest to a query image and therefore the lower-bound position errors decrease and differently the lower-bound orientation errors increase their values until, for $\alpha = 1$, we obtain the larger mean orientation errors (up to 90.45°) and lower mean position errors (up to 0.05 m). The lower bound errors for image-retrieval based approaches proposed in Table 4.1 represent the best performances obtainable by these methods when a given equivalence between position distances and orientation distances is chosen. In an analysis in witch a desirable trade-off between position error and orientation error is not a priori fixed a method is considerable good if his mean and median errors are close to the values reported in someone of the rows of the Table 4.1.

Chapter 5

Methods

To face with egocentric image base shopping cart localization problem we analyse performances of two different type of approaches: classification based methods and methods for the 3 DOF camera pose estimation. The classification based approaches are less accurate trying to associate each test image to one of the sixteen parts of the market discussed in the previous section. The approaches that try to regress the 3 DOF of the camera are divided in two different sub-families: image retrieval based methods and regression based methods. This chapter presents the investigated methods and is organized as follows: in the first three sections are discussed methods based on RGB images only, the first one presents the image retrieval approach, the second one concerns the regression based methods while the third section is related to the methods for classification task, in the forth section are showed the methods that use depth images and in the last one the experimental setting are reported.

5.1 Image retrieval methods

The image-retrieval approaches are the more classical methods for the camera localization problem and for same applicative context they could be the more appropriate approaches despite their undesirable characteristic to require a memory quantity growing linearly with training set dimension. As image-retrieval based method we test k-nn approach on different features spaces varying k between 1 and 30. To perform nearest neighbour search we use euclidean distance and cosine distance in all the different spaces, moreover we also use Pearson correlation coefficient to define the vicinity on RGB linearised vectors space. We investigate different space typologies, the first space analysed was the space obtained by linearisation of RGB images, afterwards we focus on the Improved Fisher Vector and on the spatially enhanced Improved Fisher Vector shallow representations. Finally features extracted from CNN layers trained on classification or regression task on our dataset or on different datasets were investigated. To test transfer learning ability we used the features-vector formed by 4096 elements extracted from the fc7 layer of the VGG16 network and the 2048-dimensional features-vector extracted from the mixed-7c layer of inception-V3 both trained on the ImageNet dataset [18]. Both these two representation spaces were modified, to confront with localization task, fine-tuning the two models through triplet architecture [30]. The similarity concept between images, needed for triplet training, was defined by considering similar two images if their spacial distance is less than $30cm$ and the orientation distance is smaller than $45°$ and dissimilar if at least one of these two conditions are not verified. Furthermore, to investigate the intermediate representation produced by training end to end CNNs to regress directly by images the 3 D.O.F. of camera poses we use two different architectures. We extract internal representations obtained from a 2D version of POSENET [11](obtained reducing the output space of the network) trained on our dataset with the parameter $\alpha = 125$ and from a modified version of POSENET derived from Inception-V3 architecture (INCEPTION-V3 POSENET) trained with the NPP loss function showed in Eq. 5.2 and proposed in [12]. We will discuss deeply these architectures in the next section. Finally, we conducted experiments to evaluate the increasing of performances obtainable by imposing temporal constraint to K-nn approach. To impose the temporal constraint we took in consideration the sequentiality of the frames extracted from the different videos. The pose of the first frame of each video was regressed with the classical K-nn procedure while for the successive frames we implemented the nearest neighbour search on a subspace of the market space as described by the follow heuristic. Given the f_i frame of a video we conduct nearest neighbour search on the subset of training set composed by the frames placed on a neighbourhood of the position p_{i-1} associated to f_{i-1} frame of the video. We test this heuristic for different neighbourhood sizes observing the drift effect for too small sizes and the irrelevance of the heuristic for too big sizes. We find an approximation of optimal value for neighbourhood size by fixing a radius of $4m$.

5.2 Regression based methods

The methods for camera localization based on regression are characterized by very valuable properties, they don't need to maintain the wall training set in memory and consequently are generally more compact, moreover some of them allow also fast inference. To investigate the performance of CNN-based methods we adapt POSENET architecture [11] to our 3 DOF camera pose estimation problem by modifying the architecture to produce a 2D vector corresponding to cart position and a 2D unit vector for orientation. We train the architecture using the follow parametric loss function (PP loss) proposed in [11]:

$$PP \quad loss = d(P_i^{GT}, P_i^{PR}) + \alpha d(O_i^{GT}, O_i^{PR}) \tag{5.1}$$

Where d is the euclidean distance P_i^{GT} and O_i^{GT} are respectively the ground truth position and orientation vector of the frame i, P_i^{PR} and O_i^{PR} are the position and orientation vector predicted by the network while α is a parameter to weight orientation error in relation to position error. We test this architecture varying the α parameter between the following values $\{500, 250, 125, 62.5\}$ to search the best trade-off between position error and orientation error in the loss function. For $\alpha = 125$ we obtain the best performance for this network so in our analysis we will refer to this parametric value. Moreover we built an alternative version of POSENET architecture based on Inception-V3 architecture. The INCEPTION-V3 POSENET architecture was obtained replacing, in the Inception-V3 architecture [61], the final classification layer with two fully connected layers. Is possible to think this architecture as composed by two different parts with two different roles: the first part that take as input the images and bring it in a representation space and the second part, formed by the two fully connected layers, that has the role to regress the cameras poses from the representation space produced by the first part of the network. The INCEPTION-V3 POSENET architecture has been trained using the following No Parametric Loss function (NPP loss) Eq. 5.2 proposed in [12] that automatically try to compute the optimal trade-off between the position and the orientation losses:

$$NPP \quad loss = e^{-S_p} d(P_i^{GT}, P_i^{PR}) + S_p + e^{-S_o} d(O_i^{GT}, O_i^{PR}) + S_o \tag{5.2}$$

Where S_p and S_o are two weights added to the network to automatically learn an optimal trade off between the position and orientation error, d is euclidean distance, P_i^{GT} and O_i^{GT} are the ground truth position and orientation vector of the frame i and P_i^{PR} and O_i^{PR} are the position and orientation predicted by the network. By using this loss we don't need to define any hyper-parameter α. To investigate if through multi task learning is possible to achieve best performance on localization task we train INCEPTION-V3 POSENET with a loss function obtained as sum of the NPP Loss function for the 3 DOF camera prediction Eq.5.2 and a cross entropy loss for classification task on the sixteen classes defined on our dataset. Furthermore we conducted experiments to analyse the relation between the characterization of the internal representation produced by the first part of INCEPTTION-V3 POSENET and the ability to regress the cameras 3 D.O.F of the network. We test if by forcing the INCEPTTION-V3 POSENET to represent near to each other images acquired by cameras close between them, in terms of position and orientation, and faraway the images far from each other the precision of the cameras poses regression could be improved. To investigate this opportunity we test two different strategy:

1. We implement a classical triplet network [30] with three additional regressive parts that take as input the three embedding representation of the triplet network (named "INCEPTION-V3 POSENTET REGRESSION AND CLASSIFICATION") (Figure 5.1). The network was trained using a loss obtained summing the Triplet loss function Eq.5.3 proposed in [30] that work on embedding space and the NPP loss for camera pose estimation presented above Eq.5.2.

2. We pretrain the not regressive part of inception-V3 POSENET network with triplet architecture using the similarity between images defined in the previous section and, starting by the weights determined through triplet training, we fine-tune the whole inception-V3 network with the NPP loss for camera pose estimation Eq.5.2.

$$Triplet \quad Loss(d_+, d_-) = \|(d_+, d_-1)\|_2^2 \tag{5.3}$$

where

$$d_+ = \frac{e^{\|Net(x)-Net(x^+)\|_2}}{e^{\|Net(x)-Net(x^+)\|_2} + e^{\|Net(x)-Net(x^-)\|_2}}$$

and

$$d_- = \frac{e^{\|Net(x)-Net(x^-)\|_2}}{e^{\|Net(x)-Net(x^+)\|_2} + e^{\|Net(x)-Net(x^-)\|_2}}$$

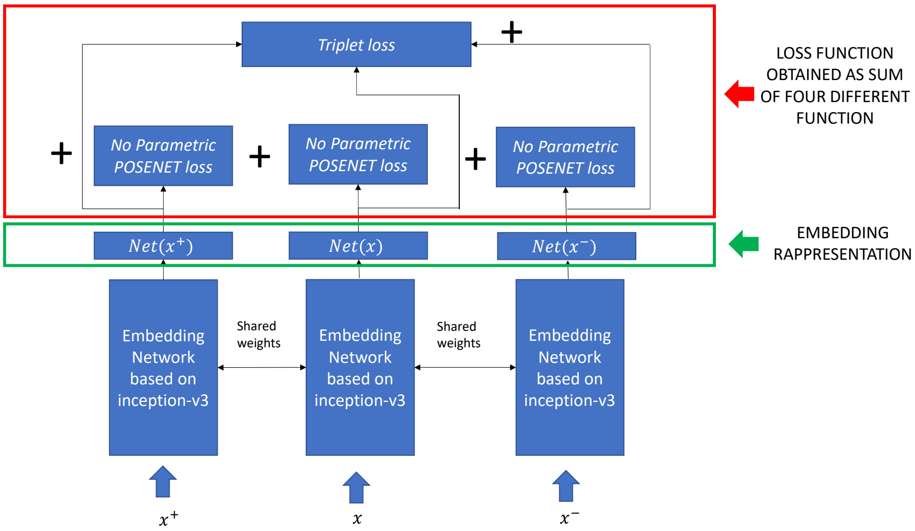

Triplet loss
+
+
No Parametric POSENET loss
+
No Parametric POSENET loss
+
No Parametric POSENET loss
LOSS FUNCTION OBTAINED AS SUM OF FOUR DIFFERENT FUNCTION
$Net(x^+)$
$Net(x)$
$Net(x^-)$
EMBEDDING RAPPRESENTATION
Embedding Network based on inception-v3
Shared weights
Embedding Network based on inception-v3
Shared weights
Embedding Network based on inception-v3
x^+
x
x^-

Figure 5.1: Graphical representation of INCEPTION-V3 POSENTET REGRESSION AND CLASSIFICATION architecture

To test if with CNN-based methods, by learning singularly position and orientation, it's possible to reach best results we conduct two experiments. We train INCEPTION-V3 POSENET, modified to produce a 2D vector output, by using as loss functions respectively the euclidean distance only between positions and only between the orientation vectors. Furthermore we perform experiments to analyse if by reducing the constraints imposed to prediction it's possible to improve performance of CNN-based approaches on position estimation. We train the INCEPTION-V3 POSENET architecture version for position prediction only to learn arbitrary positions such that the distances of the different pairs of images is preserved. To this aim we propose the following loss function:

$$Distances \quad Loss = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} d(p_i^{GT}, p_j^{GT}) - d(p_i^{PR}, p_j^{PR}) \tag{5.4}$$

Where K is the batch size dimension, p_x^{GR} is the ground truth position of the x-th frame of the batch and p_x^{PR} is the position predicted by the network for the x-th frame of the batch and d is the euclidean distance. By a geometrical point of view is possible, with the appropriate roto-translation transformation, to map the arbitrary reference system used by the network for positions prediction to the original one. To perform this mapping we computed the optimal roto-translation transformation between ground truth positions and predicted positions of the training set images by using the method based on Singular Value Decomposition (SVE) proposed in [62]. To observe how the choice of the samples present in each batch can influence the performances of this method we proposed two experiments that differ from each other for the strategies used to build the different batches. One experiment was conducted using a random sampling to form each batch, the other by inserting in each batch same reference frames and, for each one of these, a related set of frames. Each set of frames related to a reference frame was composed by selecting half of the frames randomly between the images that result to be close to the reference frame in terms of position and orientation (position distance less than 2 m and orientation distance less than 45°) and the other half randomly from the whole training set. When we use this second sampling strategy, we train the network through a variation of the Distance Loss function proposed in 5.4. In fact the loss function used in the smart sampling case take in consideration only the distance between the frames belonging to the same set of images and therefore associated to the same reference frame. This second approach, that we will name "SMART SAMPLING", try to impose to the network the same consideration to local and global relation between images to build the regressive model.
To analyse if it's possible to produce a performances improvement, by partitioning the market surface in different regions and by regressing the cameras poses separately for each part of the market, we adopt two different approaches:

1. We trained separately INCEPTION-V3 POSENET for position prediction on the images of each of the sixteen classes defined on our dataset and measured the performances of the sixteen models obtained jointly by computing mean and median errors on the whole dataset.

2. We structured a new neural network architecture FORK INCEPTION-POSENET

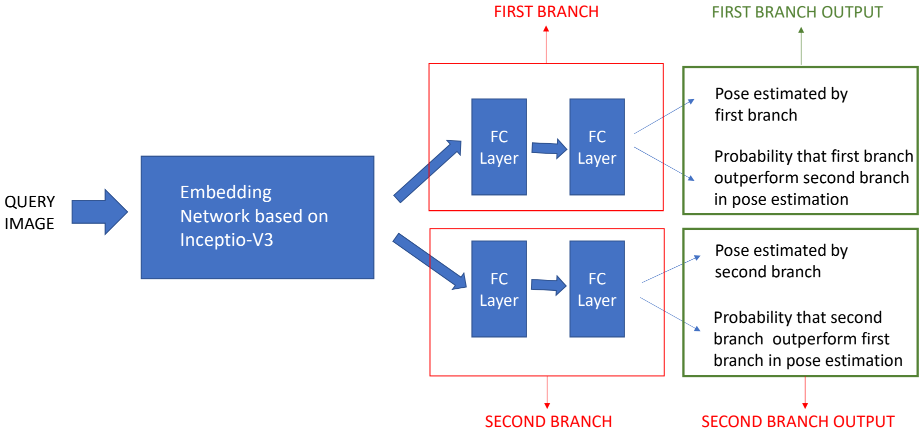

FIRST BRANCH
FIRST BRANCH OUTPUT
QUERY IMAGE
Embedding Network based on Inceptio-V3
FC Layer
FC Layer
Pose estimated by first branch
Probability that first branch outperform second branch in pose estimation
FC Layer
FC Layer
Pose estimated by second branch
Probability that second branch outperform first branch in pose estimation
SECOND BRANCH
SECOND BRANCH OUTPUT

Figure 5.2: Graphical representation of FORK INCEPTION-POSENET architecture

which differs from INCEPTION-POSENET for the regressive part of the network and the loss used with the aim to predict a partition of dataset frames and at the same time the cameras 3 DOF.

The regressive part of FORK INCEPTION-POSENET is formed by two distinct branches taking as input the same image representation vector. Each branch regress a camera pose and the probability of the branch to outperform the other branch in camera pose prediction for the input image (Figure 5.2).

The network is trained with the following loss:

$$Fork\quad Loss = NPP\quad Loss(p_{gt}, p_{bp}) + |pr_{bp} - 1| + |pr_{wp}| \tag{5.5}$$

Where $NPP\quad Loss$ is the loss function presented in Eq.5.2, that we used to train INCEPTION-POSENET, p_{gt} is the camera pose ground truth, p_{bp} is best camera pose predicted, pr_{bp} is the probability predicted by the branch that performed the best camera pose prediction and pr_{wp} is the probability predicted by the other branch. During the training phase, given a frame, the loss function try simultaneously to minimize the NPP loss function proposed in Eq.5.2, for the pose predicted by the more performing branch, the distance from one of more performing branch probability prediction and the distance from zero of the less performing branch probability prediction. During the test phase the probabilities predictions are used to select the more reliable pose prediction between the two produced by the network. Finally to investigate if classical regressive approach can be competitive respect to CNN-based methods we test performance of Support Vector Regressors on two images representation spaces:

- Representation learned fine-tuning VGG16 model (pretrained on ImageNet) with Triplet network
- Features space of the cls3_fc1 internal layer of POSENET trained on our dataset

5.3 Classification methods

To study the performances on place recognition task in grocery context, we use the sixteen classes previously defined to test classification accuracy by using different approaches. We tested the performance of modified version of inception-V3, obtained by modifying the classification layer to work on the sixteen classes of our dataset, pretrained on ImageNet and fine-tuned on our dataset. To test if the more accurate information about the 3 DOF camera poses can be useful to support classification task, we analysed the performance on classification task obtained training the INCEPTION-V3 POSENET REGRESSION AND CLASSIFICATION network (Figure 5.1). This network was trained with the sum of 3 DOF camera loss presented in Eq. 5.2 and cross entropy classification loss. Finally, to analyse how the algorithms, trained to the more constrictive 3 DOF camera estimation task, perform on the more simple place recognition task, we measured the classification accuracy by associating at each frame a class in function of position predicted by INCEPTION-V3 POSENET network.

5.4 Depth

An other aspect analysed during my studies, was related to the employment of depth images into camera localization task. We tested methods based on depth images only and methods that take as input RGB images and depth images together, both for 3 DOF camera pose estimation task and for classification task .

5.4.1 3 DOF camera pose estimation

By modifying the first convolutional layer of INCEPTION-V3 POSENET pretrained on ImageNet and adapted to the 16 classes of our dataset, we built an architecture to camera localization able for work on grayscale depth images (named INCEPTION-V3 POSENET DEPTH). The first convolutional layer was modified to work on a single channel, the weights of the one channel convolutional layer was obtained as mean of the weights of the original three RGB channels convolutional layer. Moreover we tested the opportunity to improve the performances using a network that at the same time take in input RGB images and depth images. To do this we implemented an architecture formed by two branches, one for the RGB images and one for the depth images which create two separate representation spaces, and a regressive component formed by two fully connected layers to regress the poses from the concatenation of the two features space (Figure 5.3). The branches for RGB images and depth images were obtained removing the two final fully connected layers from INCEPTION-V3 POSENET and INCEPTION-V3 POSENET DEPTH respectively.

5.4.2 Classification

To analyse the methods based on depth images for classification task we perform different experiments. We tested the performances of Inception-V3, pretrained on imagenet, on depth images as well as on RGB images and depth images together.

To evaluate the classification task on grey scale depth images we replaced the two final fully connected layers of INCEPTION-V3 POSENET DEPTH with a classification layer. To test the use of RGB images and depth images together, we modified the architecture, based on two braches, implemented to regress poses (Figure 5.3) by images and depth images, presented in the previous section. The network was

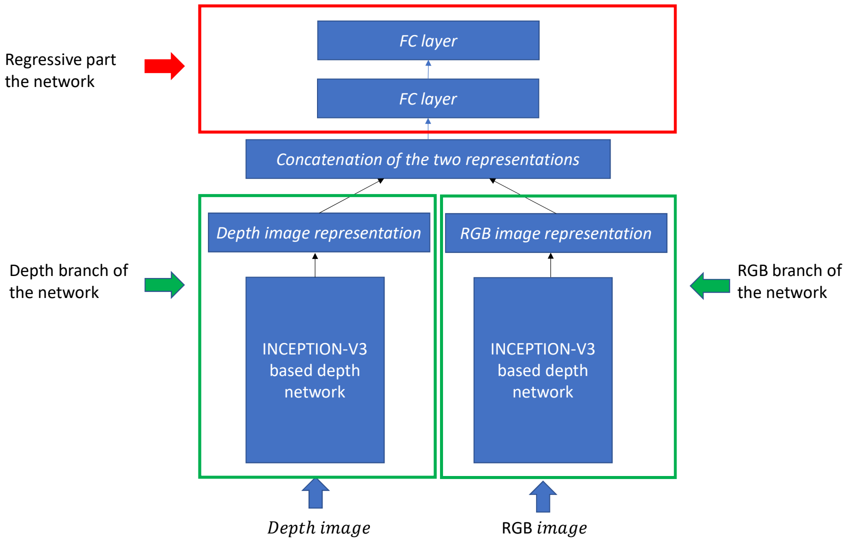

FC layer
FC layer
Regressive part the network
Concatenation of the two representations
Depth image representation
RGB image representation
Depth branch of the network
RGB branch of the network
INCEPTION-V3 based depth network
INCEPTION-V3 based depth network
Depth image
RGB image

Figure 5.3: Graphical representation of the network used to regress camera pose from RGB image and Depth image

modified by substituting the two final fully connected layers with a classification layer.

An other approach experimented to test the usage of RGB images and depth images together was the late fusion between the output on inception-v3 and the output of inception-v3 modified for depth-images. To perform the late fusion test we computed the mean of the probabilistic output of the two network trained singularly.

5.5 Experimental settings

Improved Fisher Vector was computed using gaussian mixed model with 256 components and reducing SIFT descriptor dimensionality to 80 by using PCA as suggested in [19]. The 2D version of POSENET was trained weighting position errors and orientation errors with different ratios α =500, 250, 125 and 62.5 in the loss function. The model was optimised using ADAM and with a learning rate of 10^{-3}. The α hyper-parameter is not required for the methods based on Inception-v3 that uses the NPP loss function Eq. 5.2. These methods as well as classification methods based on Inception-v3 architecture has been optimised using ADAM with a learning rate of 10^{-4}. The SVR models were trained with RBF and Linear kernels by using a grid search for parameters optimization. For both the kernels the parameter C was searched on values spaced evenly on a log scale between 10^{-3} and 10, whereas for RBF kernel the parameter γ was searched in the interval between 10^{-3} and 1. To estimate the errors for each approach we computed the mean and median distance of predicted values from camera positions and orientations ground truth.

Chapter 6

Results

In Table 6.1 we report position and orientation mean and median errors of the different K-NN approaches proposed.
All the methods based on the K-NN reached for $K = 1$ the best performances or performances very near to the best one. Consequently we report in Table 6.1 the results obtained with this parametrization. At the end of the table, the methods denoted by "TC" are those characterized by the temporal constraint. Differently Table 6.2 shows the results obtained with regression-Based Methods. In both the tables best results for each column are point out with bold number. A graphical representation of the same result was proposed in Figure 6.1(a) and (b) with position error represented on x axis and orientation error on y axis. In Figure 6.1(a) the mean errors are reported whereas in Figure 6.1(b) there are the median errors. To have a reference point for performance evaluation we plot in Figure 6.1 also the lower-bound values, obtained varying α in Eq. 4.1, for image-retrieval approaches.

6.0.1 Retrieval based methods

An analysis of 1-NN approaches shows different interesting elements. As can be expected the nearest neighbour approach on RGB image linearised space produces the worst performance: euclidean ad cosine distances result more adapt to preserve poses distance in this space compared to correlation.

1-NN on Improved Fisher vector features results show very similar performances for the two metrics analysed (mean errors 1.62 m and 13.87° and median errors 0.31 m and 3.25° using euclidean distance and mean errors 1.62 m and 13.91° and median errors 0.31 m and 3.25° using cosine distance) while 1-NN on Improved Fisher vector using spatial extended local descriptor performs better by using cosine

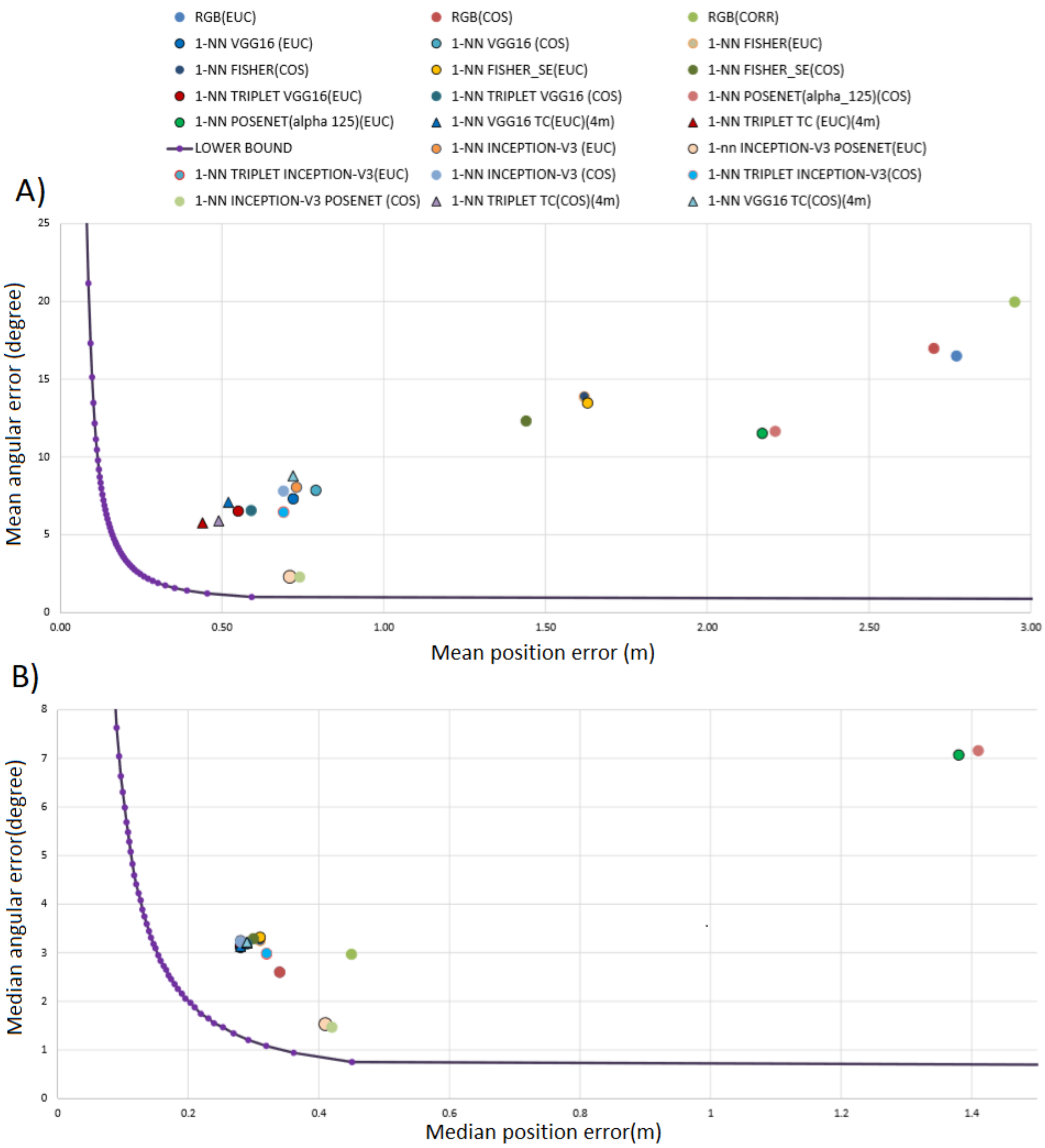


Figure 6.1: Graphical representation of mean (A) and median (B) position and orientation errors of the different 1-NN based methods.

Table 6.1: Mean and median position and orientation errors results.

	Mean		Median	
Methods	P.E.(m)	O.E.(°)	P.E.(m)	O.E.(°)
1-NN RGB (EUC)	2.77	16.5	0.34	2.6
1-NN RGB (COS)	2.70	16.99	0.34	2.6
1-NN RGB (CORRELATION)	2.95	19.98	0.45	2.97
1-NN FISHER (EUC)	1.62	13.87	0.31	3.25
1-NN FISHER (COS)	1.62	13.91	0.31	3.25
1-NN FISHER SE (EUC)	1.63	13.48	0.31	3.32
1-NN FISHER SE (COS)	1.44	12.32	0.3	3.29
1-NN VGG16 (EUC)	0.72	7.32	**0.28**	3.11
1-NN VGG16 (COS)	0.79	7.86	**0.28**	3.12
1-NN TRIPLET VGG16 (EUC)	0.55	6.52	**0.28**	3.17
1-NN TRIPLET VGG16 (COS)	0.59	6.56	0.29	3.18
1-NN INCEPTION-V3(EUC)	0.73	8.06	**0.28**	3.25
1-NN INCEPTION-V3(COS)	0.69	7.81	**0.28**	3.23
1-NN TRIPLET INCEPTION-V3(EUC)	0.69	6.47	0.32	2.98
1-NN TRIPLET INCEPTION-V3(COS)	0.69	6.44	0.32	2.99
1-NN POSENET(EUC)	2.17	11.53	1.38	7.07
1-NN POSENET(COS)	2.21	11.66	1.41	7.16
1-NN INCEPTION-V3 POSENET(EUC)	0.71	2.29	0.41	1.53
1-NN INCEPTION-V3 POSENET(COS)	0.74	**2.28**	0.42	**1.47**
1-NN TRIPLET TC (EUC)(2m)	4.65	32.31	0.43	5.13
1-NN TRIPLET TC (COS)(2m)	2.33	14.26	0.36	3.88
1-NN VGG16 TC (EUC)(2m)	3.38	27.66	0.39	4.47
1-NN VGG16 TC (COS)(2m)	0.76	10.28	0.29	3.24
1-NN TRIPLET TC (EUC)(4m)	**0.44**	5.76	0.29	3.2
1-NN TRIPLET TC (COS)(4m)	0.49	5.89	0.29	3.2
1-NN VGG16 TC (EUC)(4m)	0.52	7.09	**0.28**	3.13
1-NN VGG16 TC (COS)(4m)	0.72	8.78	0.29	3.22

distance compare to euclidean distance (mean errors 1.62 m and 13.48° and median errors 0.31 m and 3.32° using euclidean distance and mean errors 1.44 m and 12.32° and median errors 0.3 m and 3.11° using cosine distance). The results obtained with CNN features extracted from VGG16 and from Inception-v3, both trained on

Table 6.2: Mean and median position and orientation errors results.

	Mean		Median	
Methods	P.E.(m)	O.E.(°)	P.E.(m)	O.E.(°)
SVR TRIPLET (RBF kernel)	1.46	8.04	0.9	4.39
SVR TRIPLET (Linear kernel)	1.45	23.92	1.08	14.66
SVR POSENET (RBF kernel)	1.96	10.1	1.54	6.14
POSENET	1.62	7.52	1.23	4.63
INCEPTION-V3 POSENET (PP loss)	0.99	2.2	0.67	1.08
INCEPTION-V3 POSENET	0.57	1.81	0.39	1.13
INCEPTION-V3 POSENET (pretrained with triplet)	0.55	1.86	0.36	1.11
INCEPTION-V3 TRIPLET-POSENET	0.56	**1.35**	0.42	1.07
FORK INCEPTION POSENET	0.6	2.08	0.42	1.14
INCEPTION-V3 POSENET DEPTH	0.82	3.1	0.48	1.4
INCEPTION-V3 POSENET IMAGE AND DEPTH	0.62	1.52	0.40	1.14
INCEPTION-V3 POSENET REGRESSION AND CLASSIFICATION	0.66	2.38	0.47	1.32
INCEPTION-V3 POSENET ONLY ORIENTATION	-	1.4	-	**1.02**
INCEPTION-V3 POSENET ONLY POSITION	**0.42**	-	**0.29**	-
INCEPTION-V3 POSENET (DISTANCES loss)(SMART sampling)	0.44	-	**0.29**	-
INCEPTION-V3 POSENET (DISTANCES loss)(RANDOM sampling)	1.10	-	0.81	-
INCEPTION-V3 POSENET ONLY POSITION (on each class)	0.75	-	0.53	-

ImageNet for classification task, either using euclidean or cosine distance, outperform significantly the performances obtained with Improved Fisher Vector representations: the worst results obtained with features extracted from CNN trained on classification task, obtained using VGG features and cosine distance (mean errors 0.79 m and 7.86° and median errors 0.28 m and 3.12°), are strongly better than the best results obtained with Improved fisher vector features (mean errors 1.44 m and 12.32° and median errors 0.3 m and 3.29°).

The best performances obtained with CNN features can be explained as it follows: those features have a higher semantic level compared to the improved fisher vector features. This is due to the fact that these features are obtained through the use of classification task.

The results obtained, using euclidean distance, with the features extracted from VGG16 model and inception-v3 both pre-trained on ImageNet and fine-tuned with triplet architecture (mean errors 0.55 m and 6.52° and median errors 0.28 m and 3.17° with VVG16 features and mean errors 0.69 m and 6.47° and median errors 0.32 m and 2.98° with inception-v3 features) are more performing in term of mean errors respect to those obtained with the off-the-shelf VGG16 features and inception-v3 features pre-trained on ImageNet (mean errors 0.72 m and 7.32° and median errors 0.28 m and 3.11° with VVG16 features and mean errors 0.73 m and 8.06° and median errors 0.28 m and 3.25° with inception-v3 features). Also with cosine distance this improvement appears evident for VGG16 features (mean errors 0.79 m and 7.86° and median errors 0.28 m and 3.12° with VVG16 features and mean errors 0.59 m and 6.56° and median errors 0.29 m and 3.18° with TRIPLET VVG16 features) while for inception-v3 features the improvement of mean error is only in orientation prediction (mean errors 0.69 m and 7.81° and median errors 0.28 m and 3.23° with INCEPTION-V3 features and mean errors 0.69 m and 6.47° and median errors 0.32 m and 2.99° with TRIPLET INCEPTION-V3 features).

Experiment conducted using features extracted by model trained on camera relocalization task on our dataset are strongly dependent by architecture and loss function used. The 1-NN approach, that uses POSENET features, shows performances significantly less accurate respect to those discussed before, both with euclidean distance (mean errors 2.17 m and 11.53° and median errors 1.38 m and 7.07°) and cosine distance (mean errors 2.21 m and 11.66° and median errors 1.44 m and 7.16°). Differently, by using features of Inception-v3 POSENET, the 1-NN approach outperforms all the others 1-NN approaches in term of orientation error (mean error 2.29° and median error 1.53° with euclidean distance and 2.28° and 1.47° with cosine distance) but results less accurate respect to the best methods in term of position error (mean error 0.71 m and median error 0.41 m using euclidean distance and 0.74 m and 0.42 m with cosine distance) for all the investigated metrics.

As it can be expected, by imposing to the 1-NN a temporal constraint and assuming therefore the opportunity of a sequential localization of frames extracted from a video, it's possible to observe an improvement of performances both on VGG16 feature and triplet representation. Fixing the neighbourhood size to 4 m using both euclidean and cosine distance the method outperforms classical 1-NN on the same features for the mean errors and an equivalent performance for median errors (reaching a mean position error of 0.52 m and mean orientation error of 7.09° for VGG16 features with euclidean distance, a mean position error of 0.72 m and mean orientation error of 8.78° for VGG16 features with cosine distance, a mean position error of 0.44 m and mean orientation error of 5.76° for triplet features with euclidean distance and a mean position error of 0.49 m and mean orientation error of 5.89° for triplet features with cosine distance). The choice of an appropriate neighbourhood size is central for the goodness of this approach. We observe how, by fixing it to 2m, drifting effect can produce very poor results. For instance by using euclidean distance and working with triplet features we observe a mean errors of 4.65 m and 32.31° and median errors of 0.43 m and 5.13°.

It's interesting to observe how the improvement obtained imposing temporal constraint on VGG16 features it's comparable to that obtained fine-tuning the model with triplet architecture without impose any constraint. Furthermore, how it's possible to observe in Table 6.2, the effect of temporal constraint on triplet representation is less significant. These two observations suggest that network fine-tuning obtained through triplet, as described in section 5.1, is a useful instrument to reduce the representation ambiguity due to the presence of similar structure in different part of the store. This ambiguity reduction depends on the ability of triplet network to create a representation space in which images of nearby locations are mapped close and images not near are mapped not close one to each other. Finally, it should be noted that triplet fine-tuning it's always applicable, while differently the conditions to impose temporal constraint to 1-NN search are not always substantiated, in particular when low-power devices, working only at a low frame rate, are used. Moreover the absence of drift effect due to temporal constraint it's impossible to guarantee a priori.

6.0.2 Regression based methods

The methods based on regression show very different results depending on regressor type and for CNN-based methods also on architecture and loss function used. Support Vector Regressor shows the worst performances between the regressive model analysed. On the features extracted by VGG fine-tuned with Triplet, both using RBF kernel (mean position error 1.46 m, mean orientation error 8.04°, median position error 0.9 m and median orientation error 4.39°) and Linear kernel (mean position error 1.45 m, mean orientation error 23.92°, median position error 1.08 m and median orientation error 14.66°), the results are very far away from the performance of 1-NN approach with the same features analysed in the previous section. Also on features extracted by POSENET architecture trained on our dataset, SVR approach shows to be inaccurate (mean position error 1.96 m, mean orientation error 10.01°, median position error 1.54 m and median orientation error 6.14°) being outperformed by

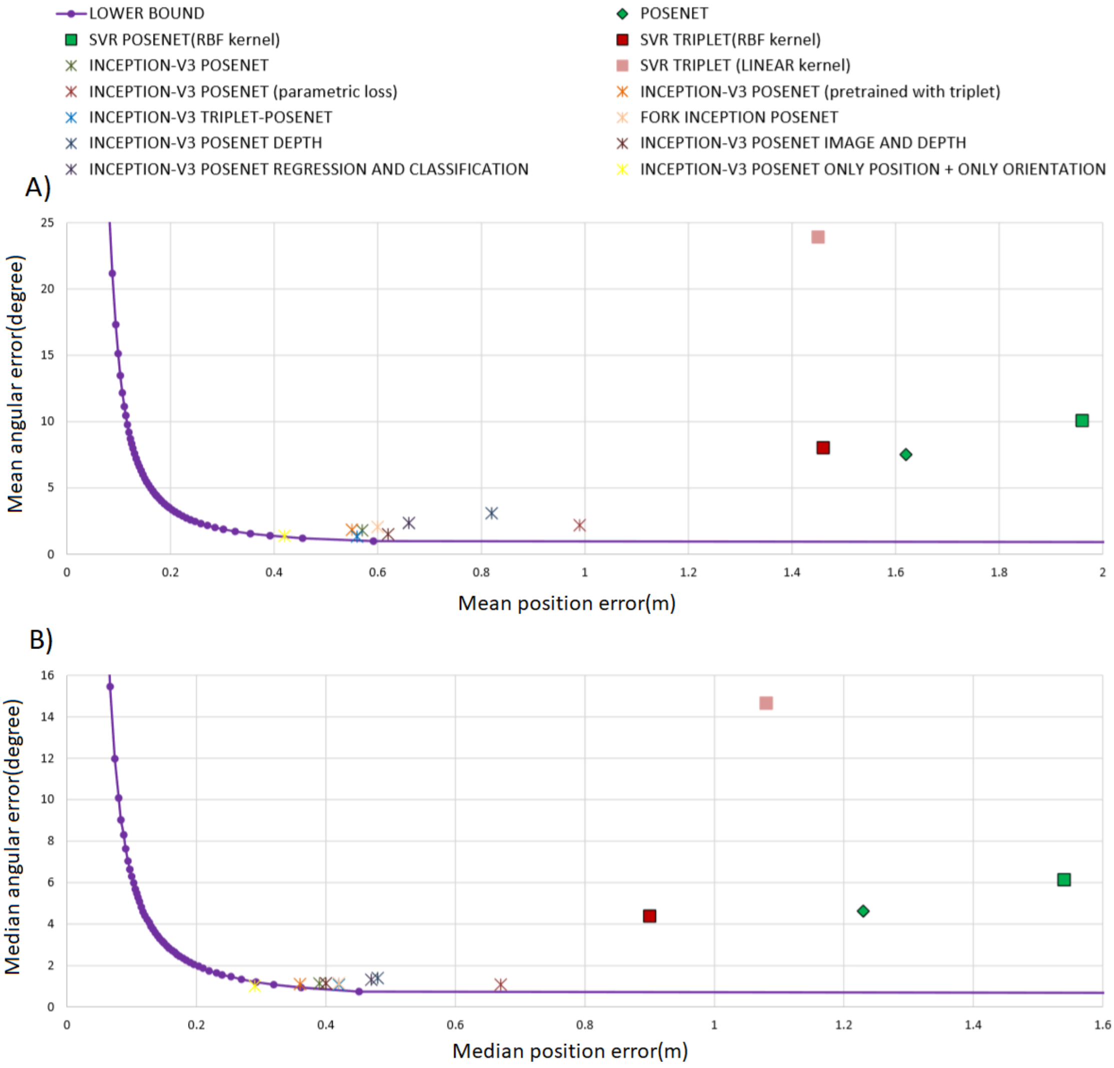


Figure 6.2: Graphical representation of mean (A) and median (B) position and orientation errors of the different regression based methods.

POSENET model in all the considered measures (mean position error 1.62 m, mean orientation error 7.52°, median position error 1.23 m and median orientation error 4.63°). The methods based on INCEPTION-v3 model produce significantly better performances respect to the other regressive approaches. The results obtained by replacing Googlenet architecture with INCEPTION-V3 in POSENET, maintaining the PP loss presented in Eq. 5.1 and fixing $\alpha = 125$ (mean position error 0.99 m, mean orientation error 2.2°, median position error 0.67 m and median orientation error 1.08°), show how the embedding part of the architecture has a central role to determine the CNN-based model performance. Using the same architecture but training it with the loss reported in Eq. 5.2 we observe an other significant reduction of position error (passing by a mean error of 0.99 m and a median error of 0.67 m with parametric POSENET Loss to a mean error of 0.57 m and a median error of 0.39 m with the no parametric POSENET Loss). It should be noted that by pre-training inception-V3 network with triplet, as discussed in section 5.1, and using this model as embedding model in INCEPTION-v3 POSENET architecture, there is an increase of position estimation performance respect to the same model trained by employing inception-v3 pre-trained with ImageNet as initial embedding model (mean position errors decreases from 0.57 m to 0.55 m, while median position error decreases from 0.39 m to 0.36 m). Differently the model trained with the loss composed by a sum between triplet loss and NPP loss, named INCEEPTION-v3 TRIPLET-POSENET outperforms INCEPTION-v3 POSENET in terms of orientation errors (mean orientation errors decreases from 1.81° to 1.35°, while median orientation error decreases from 1.13° to 1.07°) and prove to be less accurate in terms of median position errors (0.39m reach by INCEPTION-v3 POSENET and 0.42m by INCEPTION-v3 TRIPLET-POSENET). The experiment conducted with FORK INCEPTION-V3 POSENET model, presented in the previous section, shows performances (0.60 m and 2.08° mean errors and 0.42 m and 1.14° median errors) less accurate compare to those obtained with INCEPTION POSENET architecture. The partition of the space automatically produced by our architecture looks like qualitatively subdividing the dataset in two coherent parts. As it is possible to observe in Fig. 6.3, FORK INCEPTION-V3 POSENET tends to assign to the same regressor the frames acquired consecutively through the same video when the camera is moving with a constant direction (e.g. cart is moving in a corridor).

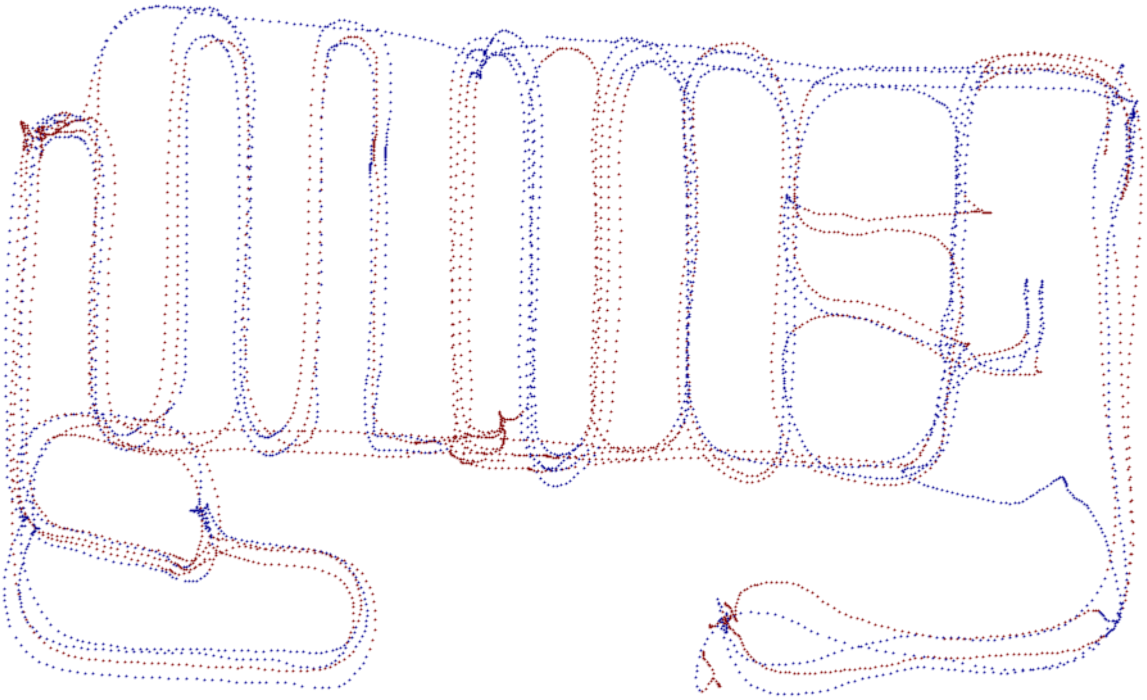


Figure 6.3: Partition produced on test set by FORK INCEPTION-3 POSENET architecture, in blue are coloured the cameras associated to the first regressor of the network in red those associated to the second regressor

Under these conditions the visual information is strongly related between different frames. Differently when the cameras turn (e.g. cart is moving by a corridor to an other perpendicular) the visual informations are quickly change and consequently also the best regressor that is associated to the query frame can change. Despite the characteristics aforementioned of space partition are intuitively useful for camera pose regression task, the worst performance obtained compared to INCEPTION-POSENET can suggest two possible explanations:

1. The shared embedding space between the two regressors can be suboptimal for both the regressors;

2. The reduction of samples on which each regressor is trained can reduce the generalization ability of the two regressors.

The experiment conducted using INCEPTION-V3 POSENET trained with depth images, as it can be expected, achieves worse results (0.82 m and 3.1° mean errors and 0.48 m and 1.4° median errors) compared to the same network trained on RGB images (0.57 m and 1.81° mean errors and 0.39 m and 1.13° median errors). The principal motivation of this gap could be the regularity of 3D structure of the market that makes the depth information less useful to disambiguate the different parts of the store compared to RGB information. Using our CNN model, based on inception-v3 to perform camera pose regression, which uses RGB images together with depth images, we observe performances (0.62 m and 1.52° mean errors and 0.40 m and 1.14° median errors) similar to those observed with INCEPTION-V3 POSENET in terms of median errors, an increase as regards position mean error and a decrease of orientation mean error. These results show questionable benefit to introduce depth informations for regression task.

The CNN model based on loss function, wich join NPP loss Eq.5.2 and classification loss, shows worse results (0.66 m and 2.38° mean errors and 0.47 m and 1.32° median errors) compared to the same model with only regression loss. This comparison point out that the classification loss prevents weight optimization for regression task. Moreover the two INCEPTION-V3 POSENET networks, trained respectively only on the position regression (mean error 0.42 m and median error 0.29 m) and only on the orientation regression (mean error 1.4° and median error 1.02°), show that tackling the two problems separately is more effective. If the results related to orientation appear similar to those obtained with TRIPLET POSENET architecture the performance in position estimation strongly outperforms all other methods in terms of mean error and is equivalent to those obtained by best methods in median position error. The appreciable performances obtained by many of the methods based on INCEPTION-V3 architecture in orientation estimation (less than 2° of mean error) bring us to focus our attention on position prediction.

Testing the INCEPTION-V3 POSENET architecture with Distances Loss Eq.5.4 for position estimation it's possible to observe that, with the random sampling, the model is unable to reach performance comparable with those observed with the NPP loss presented in Eq.5.2 (mean error of 1.10 m and median error of 0.81 m). Differently using the smart sampling discussed in section 5.2 the method based on Distance Loss reaches performances similar to those obtained with INCEPTION-V3 POSENET trained to regress only the cameras position (mean error 0.44 m and median error 0.29 m with Distance Loss and mean error of 0.42 m and median error of 0.29 m for INCEPTION-V3 POSENET using NPP loss 5.2).

Finally, we observe that, joining the results of the sixteen different INCEPTION-V3 POSENET models (each one trained to infer cameras position in one of the sixteen classes defined in 4.0.2), the performances are less accurate, in terms of both mean and median position error (mean error 0.75 m and median error 0.53 m), compared to those obtained with almost all the methods discussed based on INCEPTION-V3 architecture.

6.0.3 Retrieval based methods VS Regression based methods

In this section we will compare the most representative image retrieval based methods and regression based methods between them , through an analysis of performances and computational costs. In Figure 6.4 is reported the graphical representation of the performance of the 3 DOF camera pose estimation methods selected for the comparison. Differently in Table 6.3 we show the inference times and required amounts of memory for the same methods.

The times proposed are related to the poses predictions of 100 images and are evaluated both on CPU and on GPU. The memory required by the different models was expressed in megabytes (MB) and for image-retrieval approaches were reported both training set dimension and models dimension. The timings performances on CPU have been obtained using a machine with Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz and 32 GB of RAM. The times on GPU were computed using a NVIDIA K-40 GPU. Generally the regression-based methods are less space consuming compared with retrieval-based methods. This is motivated by the need, of the image-retrieval methods, to maintain the training set in memory. However, it should be noted that, for compact features spaces, the training sets dimension can be small (e.g., the training set represented on features space extracted from POSENET or INCEPTION-V3 consume only 104 MB). Moreover the regression-based methods result also faster both in CPU and GPU. By comparing CNN-regressive approaches with 1-NN approaches on the features extracted by these CNN architectures, it's possible to observe significant differences in terms of time performances on GPU (e.g., $2s$ for pose inference with POSENET, while $6s$ are required by 1NN POSENET. Similarly $3s$ are required by INCEPTION-V3 POSENET, while 1-NN INCEPTION-V3 requires 7 seconds); while the aforementioned differences are less

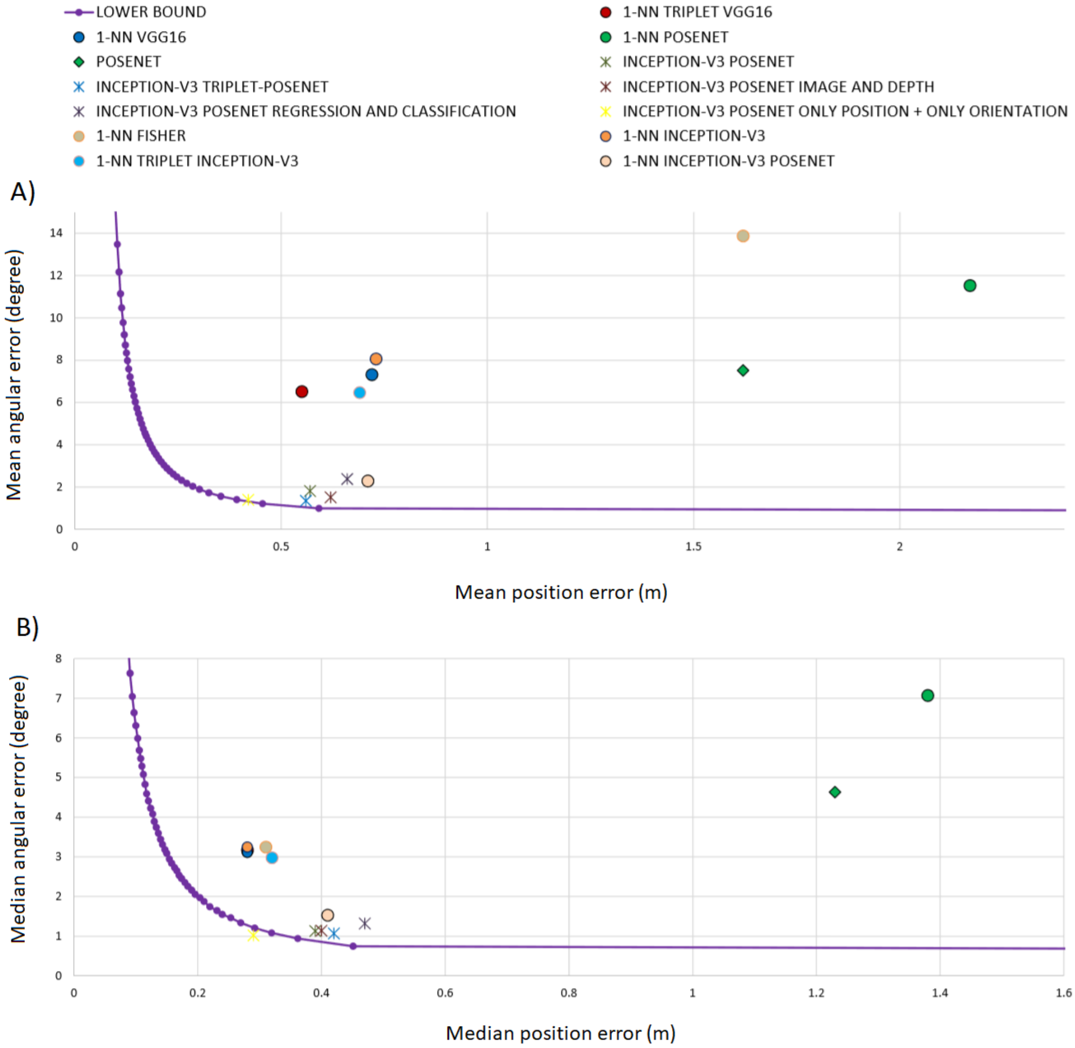


Figure 6.4: Graphical representation of mean (A) and median (B) position and orientation errors of the most representative methods.

significant working on CPU (2 : 08 minutess for POSENET against 2 : 12 minutes for 1-NN POSENET and 34s for INCEPTION-V3 POSENET against 38s needed using 1-NN INCEPTION-V3 approach). In Fig. 6.4 A) is possible to observe how, in terms of mean errors, the best regressive based methods are similar in cameras position estimation and more accurate in orientation cameras prediction compared to the best image retrieval methods. Only the model composed by two distinct CNN, one trained on position prediction and one on orientation prediction, is able to outperform all image-retrieval based approaches also in mean position errors (mean position error 0.42 m , mean angular error 1.4°, median position error 0.29 m and median angular error 1.02°). In terms of median errors 1-NN approaches are generally more accurate for the position estimation and less accurate for the orientation estimation. Moreover it's interesting to observe that, by considering median position error, only the INCEPTION-V3 POSENET ONLY POSITION + ONLY ORIENTATION regression-based method reaches performances similar to those obtained by best image-retrieval approaches (0.29 m for INCEPTION-V3 POSENET ONLY POSITION + ONLY ORIENTATION and 0.28 m for 1-NN INCEPTION-V3). However INCEPTION-V3 POSENET ONLY POSITION + ONLY ORIENTATION model results to be less compact (258MB) and less fast (0 : 06min on GPU and 1 : 08 on CPU) compared to other regression methods based on INCEPTION-V3 architecture (129MB, 0 : 03min on GPU and 0 : 34min on CPU) as well as compared to image retrieval approaches based on features extracted from INCEPTION-V3 architecture (197MB, 0 : 07min on GPU and 0 : 38min on CPU). It's also interesting to observe that 1-NN approaches based on features extracted from regressive model (1-NN POSENET and 1-NN INCEPTION-V3 POSENET) result less performing compared to the corresponding regressive model, POSENET and INCEPTION-V3 POSENET.

6.0.4 Classification

Table6.4 reports the results of the different methods proposed for classification task, in Figure 6.5 are instead represented theirs confusion matrices. It's possible to observe that using INCEPTION-V3 POSENET model to predict the cameras 3 DOF and by associating at each position predicted the associated class we obtain the 92%

Table 6.3: Time and memory requirements.

MODEL	PROCESSING TIMES		MEMORY REQUIREMENTS		
	ON GPU (mm:ss)	ON CPU (mm:ss)	TRAINING SET (MB)	MODEL (MB)	TOTAL MEMORY USED (MB)
1-NN FISHER	-	04:08	4175	**0.4**	4175.4
1-NN VGG16	00:12	01:18	209	512	721
1-NN TRIPLET(VGG16)	00:12	01:17	209	512	721
1-NN POSENET	00:06	02:12	104	37	141
1-NN INCEPTION-V3	00:07	00:38	104	93	197
1-NN TRIPLET(INCEPTION-V3)	00:07	00:38	104	93	197
1-NN INCEPTION-V3 POSENET	00:07	00:38	104	93	197
POSENET	**00:02**	02:08	**-**	73	**73**
BASED INCEPTION-V3 POSENET	00:03	**00:34**	**-**	129	129
INCEPTION-V3 POSENET REGRESSION AND CLASSIFICATION	00:03	00:36	**-**	130	130
INCEPTION-V3 POSENET IMAGE AND DEPTH	00:05	01:50	**-**	332	332
INCEPTION-V3 POSENET ONLY POSITION + ONLY ORIENTATION	00:06	1:08	**-**	258	258

of accuracy, less than all the other methods proposed. As observed for the regression task, also for classification task, the RGB images result to be more informative compared to the depth-images. In fact it's possible to observe that Inception-V3 trained exclusively on RGB images outperforms the modified version of Inception-V3 trained on depths images (95.1% of accuracy against 94.1%). Furthermore also the late fusion between prediction of INCEPTION-V3 trained with images and of INCEPTION-V3 trained with depth-images result less performing (94.8%) compared to the simpler INCEPTION-V3 model trained with images (95%). This result shows that there is a strong relation in what the two models have learned.
It should be noted that, differently from what we have seen in the previous section for regression task, in the case of classification task the inception-v3 model, trained simultaneously on regression and classification, outperforms the same network trained only for classification task (95.9% in front of 95.1%). The regressive component of the loss function probably helps to build a more structured representation space useful to discriminate the different classes. Finally, the experiment conducted by using both the RGB images and the depth images, differently from what we discussed for regression task, outperforms the RGB image-based model (96.4% in front of 95.1%) and all the other models proposed. The confusion matrices reported in Figure 6.5 show that all the classification methods tested fail principally by classifying as belonging to classes 11 and 15 images belonging to other classes. This is due to two principal motivation:

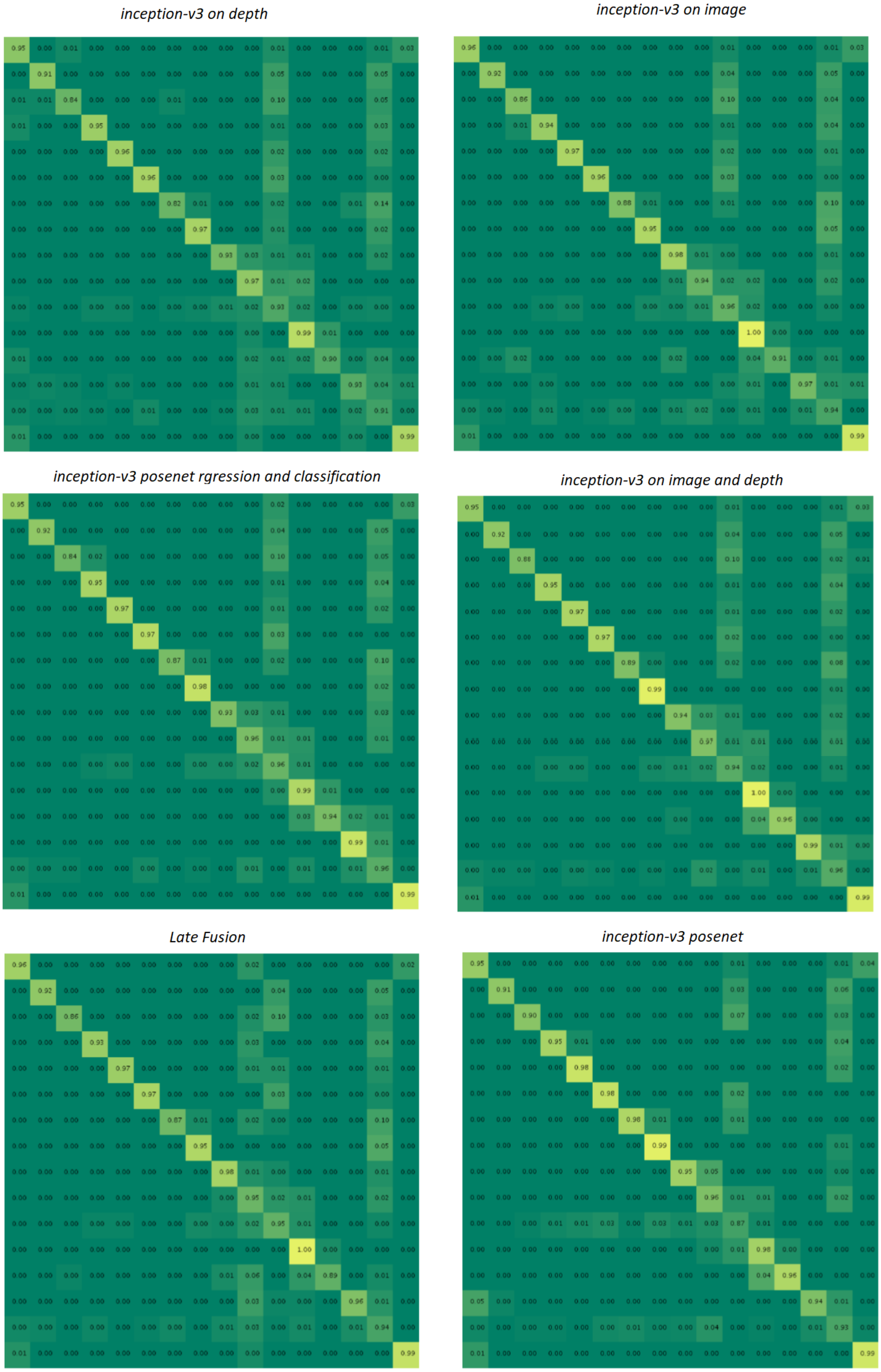
1. the classes 11 and 15 are related to market surfaces bigger than those associated to the other classes and consequently present more images

2. the parts of the market associated to this two classes are adjacent to many other parts of the market associated to the other classes and consequently

Table 6.4: Mean and median position and orientation errors results.

Methods	accuracy
INCEPTION-V3 ON IMAGE AND DEPTH	**96.4**
INCEPTION-V3 POSENET REGRESSION AND CLASSIFICATION	95.9
INCEPTION-V3 ON IMAGE	95.1
INCEPTION-V3 ON DEPTH	94.1
INCEPTION-V3 POSENET	92.0
LATE FUSION	94.8

many images belonging to these two classes are very similar to images belonging to other classes acquired in positions near to the adjacent lines.



inception-v3 on depth

inception-v3 on image

inception-v3 posenet rgression and classification

inception-v3 on image and depth

Late Fusion

inception-v3 posenet

Figure 6.5: Confusion matrix of the different methods used on classification task.

Chapter 7

Conclusion and future works

In this work, we investigated the shopping cart localization problem by egocentric images. We proposed the first dataset for shopping cart localization in retail store, correlated with RGB images and depth images labelled to face the problem both as camera 3 DOF estimation problem and as place recognition problem. To analyse the camera localization task in retail environment, we benchmark CNN-based classification approaches for place recognition task and, for camera 3 DOF estimation task, the retrieval-based and regression-based approaches. Through the classification experiments we observe that the depth images are less informative for this task compared to RGB images. Nevertheless that it's possible to use jointly the depth and RGB information to improve RGB image based classification approaches. Moreover the experiments point out that, by combining a 3 DOF regression loss function and a classification loss function, the performance on classification task improves respect to that obtained when the experiment is conducted with the same architecture and only the classification loss. Differently the improvement of the performance doesn't happen considering the regression task. Regarding the camera 3 DOF estimation methods we show that the regression based methods are generally more compact and fast. The best regression based methods outperform the retrieval based methods in orientation estimation, are similar in terms of mean position errors and less accurate by considering median position errors. The only regressive model able to perform similar to the retrieval models, in terms of median position errors, results to be that one composed by two different networks working separately on orientation and position estimation. This model requires an amount of memory and time greater than that required by the most compact retrieval based methods. An other interesting element pointed out by our analysis is the opportunity to use triplet network architecture to improve the performances both of methods based on image retrieval and, less significantly, of regression based methods. This result shows that, binding the distances between representations in the embedding features space with distances between cameras poses, is possible to facilitate the 3 DOF camera pose task. By observing the interesting results obtained using triplet network architecture to model the embedding space, in the future investigations should take into account a more systematic analysis of the relation between regressive methods performance and the characterization of the embedding space (e.g. using triplet architecture and imposing different and more sophisticate concepts of similarity between images). Moreover, to take in consideration the typical characterization of frames acquisitions in store context during the daily activity, other aspects to investigate will be: how the different models work with the presence of occlusions and the models robustness to the exchange of products position in the store.

Appendix A

A.1 Other Publications

In the following, it is reported a list of works published during my Ph.D. but not directly related to this thesis.

International Journals:

- A. Agodi, M. Barchitta, A. Quattrocchi, E. Spera, G. Gallo, F. Auxilia, S. Brusaferro, M.M. D'Errico, M. T. Montagna, C. Pasquarella, S. Tardivo, I. Mura. Preventable proportion of intubation-associated pneumonia: Role of adherence to a care bundle. In PloS one. 2017
- E. Spera, M. Migliore, N. Unsworth, D. Tegolo. On the cellular mechanisms underlying working memory capacity in humans. In Neural Network World. 2016

International Conference:

- E.Spera, G. Gallo, D. Allegra, F. Stanco, A. Maugeri, A. Quattrocchi, M. Barchitta, A. Agodi. Randomized G-Computation Models in Healthcare Systems. In IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS). 2018

Bibliography

[1] Hahnel, D., Burgard, W., Fox, D., Fishkin, K., Philipose, M. (2004, April). Mapping and localization with RFID technology. In IEEE International Conference on Robotics and Automation (ICRA), 2004.

[2] M. Sturari, D. Liciotti, R. Pierdicca, E. Frontoni, A. Mancini, M. Contigiani, P. Zingaretti. Robust and affordable retail customer profiling by vision and radio beacon sensor fusion. In Pattern Recognition Letters, 2016.

[3] A. Kara, H. L. Bertoni. Blockage/shadowing and polarization measurements at 2.45 GHz for interference evaluation between Bluetooth and IEEE 802.11 WLAN, IEEE Antennas and Propagation Society International Symposium. Held in conjunction with: USNC/URSI National Radio Science Meeting (Cat. No.01CH37229), Boston, MA, USA, 2001.

[4] C. Soto, B. Song, A.K. Roy-Chowdhury. Distributed multi-target tracking in a self-configuring camera network. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[5] M. Contigiani, R. Pietrini, A. Mancini, P. Zingaretti. Implementation of a tracking system based on UWB technology in a retail environment.In IEEE International Conference on Mechatronic and Embedded Systems and Applications (MESA), 2016.

[6] R. Pierdicca, D. Liciotti, M. Contigiani, E. Frontoni, A. Mancini, P. Zingaretti. Low cost embedded system for increasing retail environment intelligence. In IEEE International Conference on Multimedia and Expo Workshops (ICMEW), 2015.

[7] D. Liciotti, M. Contigiani, E. Frontoni, A. Mancini, P. Zingaretti, V. Placidi. Shopper analytics: A customer activity recognition system using a distributed rgb-d camera network. In Lecture Notes in Computer Science (LNCS), 2014.

[8] L. Del Pizzo, P. Foggia, A. Greco, G. Percannella, M. Vento. A versatile and effective method for counting people on either RGB or depth overhead cameras. In IEEE International Conference on Multimedia and Expo Workshops (ICMEW), 2015.

[9] E. Spera, A. Furnari, S. Battiato, G. M. Farinella. Egocentric Shopping Cart Localization. In International Conference on Pattern Recognition (ICPR), 2018.

[10] E. Spera, A. Furnari, S. Battiato, G. M. Farinella. Performance Comparison of Methods Based on Image Retrieval and Direct Regression for Egocentric Shopping Cart Localization. In 4th International Forum on Research and Technologies for Society and Industry (RTSI), 2018.

[11] A. Kendall, M. Grimes, and R. Cipolla. PoseNet: A Convolutional Network for Real Time 6-DOF Camera Relocalization. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015.

[12] A. Kendall, R. Cipolla. Geometric loss functions for camera pose regression with deep learning. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[13] F. Walch, C. Hazirbas, L. Leal-Taixé, T. Sattler, S. Hilsenbeck, D. Cremers. Image-based localization with spatial lstms. In arXiv, 2017.

[14] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In Computer Vision and Pattern Recognition (CVPR), 2013.

[15] Van Opdenbosch, G. Schroth, R. Huitl, S. Hilsenbeck, A. Garcea, and E. Steinbach. Camera-based indoor positioning using scalable streaming of compressed binary image signatures. In IEEE International Conference on Image Processing (ICIP), 2014.

[16] R. Gherardi, M. Farenzena, A. Fusiello. Improving the efficiency of hierarchical structure-and-motion. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

[17] M. Farenzena, A. Fusiello, R. Gherardi, R.Toldo. Structure-and-Motion Pipeline on a Hierarchical Cluster Tree. Proceedings of the IEEE International Workshop on 3-D Digital Imaging and Modeling (3DIM), 2009.

[18] K. Simonyan, A. Zisserman. Very deep convolutional networks for large-scale image recognition. In International Conference on Learning Representations (ICLR), 2015.

[19] F. Perronnin, J. Sanchez, and T. Mensink, Improving the Fisher kernel for large-scale image classification In Proceedings of the IEEE European Conference on Computer Vision (ECCV), 2010.

[20] J. Sanchez, F. Perronnin, T. Emìdio de Campos. Modeling the spatial layout of images beyond spatial pyramids. In Pattern Recognition Letters, 2012.

[21] http://www.fellowrobots.com

[22] T. Weyand, I. Kostrikov, and J. Philbin. Planet-photo geolocation with convolutional neural networks. In arXiv preprint arXiv:1602.05314, 2016.

[23] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: an efficient alternative to SIFT or SURF. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2011

[24] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. In IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2006.

[25] M. Farenzena, A. Fusiello, R. Gherardi, R.Toldo. SAMANTHA: a Hierarchical, Efficient, Available Structure and Motion Pipeline. In arXiv:1506.00395v1, 2015.

[26] G. Baatz, K. Köser, D. Chen, R. Grzeszczuk, and M. Pollefeys. Leveraging 3D city models for rotation invariant place-of-interest recognition. In International Journal of Computer Vision (IJCV), 2011

[27] Y. Li, N. Snavely, and D. P. Huttenlocher. Location Recognition using Prioritized Feature Matching. In Proceedings of the IEEE European Conference on Computer Vision (ECCV), 2010.

[28] K. Lin, H.-F. Yang, J.-H. Hsiao, and C.-S. Chen. Deep learning of binary hash codes for fast image retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[29] R. Hadsell, S. Chopra, Y.LeCun. Dimensionality reduction by learning an invariant mapping. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2006.

[30] E. Hoffer, N. Ailon. Deep metric learning using triplet network. In International Workshop on Similarity-Based Pattern Recognition, 2015.

[31] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. Deep Image Retrieval: Learning Global Representations for Image Search. In Proceedings of the European Conference on Computer Vision (ECCV), 2016.

[32] A. Doumanoglou, V. Balntas, R. Kouskouridas, and T.-K. Kim. Siamese Regression Networks with Efficient mid-level Feature Extraction for 3D Object Pose Estimation. In arXiv preprint arXiv:1607.02257, 2016.

[33] P. Wohlhart and V. Lepetit. Learning descriptors for object recognition and 3d pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[34] T. Sattler, A. Torii, J. Sivic, M. Pollefeys, H. Taira, M. Okutomi, and T.Pajdla. Are Large-Scale 3D Models Really Necessary for Accurate Visual Localizathow the different models work in the presence of occlusions and the models robustness to the exchange of products position in the storeion? In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[35] A. R. Zamirand M. Shah. Accurate image localization based on google maps street view. In European Conference on Computer Vision (ECCV), 2010.

[36] W. Zhang and J. Kosecka. Image based localization in urban environments. In International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT), 2006.

[37] T. Sattler, B. Leibe, and L. Kobbelt. Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization. In IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2016.

[38] R. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. In International Journal of Computer Vision (IJCV), 1994.

[39] Z. Kukelova, M. Bujnak, and T. Pajdla. Real-Time Solution to the Absolute Pose Problem with Unknown Radial Distortion and Focal Length. In IEEE International Conference on Computer Vision (ICCV), 2013.

[40] V. Santarcangelo, G. M. Farinella, S. Battiato. Egocentric Vision for Visual Market Basket Analysis. In European Conference on Computer Vision (ECCV), 2016.

[41] A. Furnari, G. M. Farinella, S. Battiato. Recognizing Personal Locations From Egocentric Videos. In IEEE Transactions on Human-Machine Systems, 2017.

[42] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovi. Going deeper with convolutions. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[43] Y. Feng, L. Fan, Y. Wu. Fast Localization in Large Scale Environments Using Supervised Indexing of Bynary Features. In Transaction on Image Processing, 2016.

[44] A. Irschara, C. Zach, J.-M. Frahm, H. Bischof. From Structure-from-Motion Point Clouds to Fast Location Recognition. In Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[45] D. Nistér, H. Stewénius, Scalable Recognition with a Vocabolary tree. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2006.

[46] J. Sivic, A. Zisserman, Video Google: A Text Retrieval Approach to Object Matching in Videos. In IEEE International Conference on Computer Vision (ICCV), 2003.

[47] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Lost in quantization: Improving particular object retrieval in large scale image databases. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008.

[48] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object Retrieval with Large Vocabularies and Fast Spatial Matching. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007.

[49] G. Schindler, M. Brown, R. Szeliski, City-Scale Location Recognition. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007.

[50] J. Knopp, J. Sivic, T. Pajdla, Avoding Confusing Features in Place Recognition. In IEEE European Conference on Computer Vision (ECCV), 2010.

[51] A. Torii, J. Sivic, T. Pajdla, M. Okutomi, Visual Place Recognition with Repetitive Structures. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013.

[52] D. Chen, G. Baatz, K. Kóser, S. Tsai, R. Vedantham, T. Pylvánáinen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, R. Grzeszczuk, City-scale landmark identification on mobile devices. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.

[53] R. Arandjelović, A. Zisserman, DisLocation: Scalable descriptor distinctiveness for location recognition. In Asian Conference on Computer Vision (ACCV), 2014.

[54] S. Cao, N. Snavely. Graph-based discriminative learning for location recognition. In Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013.

[55] S. Cao, N. Snavely. Minimal Scene Descriptions from Structure from Motion Models. In Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[56] P. Gronat, J. Sivic, G. Obozinski, P. Tomas. Learning and calibrating per-location classifiers for visual place recognition. In International Journal of Computer Vision (IJCV), 2016.

[57] I. Melekhov, Y. Juha; K. Juho; R. Esa. Image-based Localization using Hourglass Networks. arXiv:1703.07971, 2017.

[58] J. Wu, L. Ma, X. Hu. Delving Deeper into Convolutional Neural Networks for Camera Relocalization. In IEEE International Conference on Robotics and Automation (ICRA), 2017.

[59] B. Harwood, V. Kumar, G. Carneiro, I. Reid, T. Drummond. Smart Mining for Deep Metric Learning. In IEEE International Conference on Computer Vision (ICCV), 2017.

[60] V.Kumar, G. Carneiro, I. Reid. Learning Local Image Descriptors with Deep Siamese and Triplet Convolutional Networks by Minimising Global Loss Functions. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[61] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna. Rethinking the Inception Architecture for Computer Vision. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[62] P. J. Besl, N. D. McKay. A method for registration of 3-D shapes. In IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 1992.

[63] C. Cortes, V. Vapnik. Support-Vector Networks. In Machine Learning, 1995.

[64] D. Lowe. Distinctive image features from scale-invariant keypoints. In International Journal of Computer Vision (IJCV), 2004.

[65] Y. Ke, R. Sukthankar. PCA-SIFT: a more distinctive representation for local image descriptors. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2004.

[66] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool. Speeded-up robust features (SURF). In Computer Vision and Image Understanding (CVIU). 2008

[67] C. Zach, M. Klopschitz, M. Pollefeys. Disambiguating visual relations using loop constraints. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

[68] N. Jiang, P. Tan, L. F. Cheong. Seeing double without confusion: Structure-from-motion in highly ambiguous scenes. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012

[69] E. Kruppa. Zur ermittlung eines objektes aus zwei perspektiven mit innerer orientierung. In Sitzungsberichte der Mathematisch-Naturwissenschaftlichen Kaiserlichen Akademie der Wissenschaften, 1913.

[70] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. In Nature, 1981.

[71] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. In International Journal of Computer Vision (IJCV), 1992.

[72] B. Triggs, P. Mclauchlan, R. Hartley, A. Fitzgibbon. Bundle adjustment - a modern synthesis. In Vision Algorithms: Theory and Practice, 2000.

[73] O. Ozyesil, A. Singer, R. Basri. Stable Camera Motion Estimation Using Convex Programming. In SIAM Journal on Imaging Sciences (SIIMS), 2014.

[74] O. Yilmaz, F. Karakus. Stereo and kinect fusion for continuous 3D reconstruction and visual odometry. In International Conference on Electronics, Computer and Computation (ICECCO). 2013

[75] Y. Avrithis, Y. Kalantidis, G. Tolias, E. Spyrou. Retrieving Landmark and Non-Landmark Images from Community Photo Collections. In ACM Multimedia (ACM-MM), 2010.

[76] S. Gammeter, T. Quack, L. Van Gool. I Know What You Did Last Summer: Object-Level Auto-Annotation of Holiday Snaps. In IEEE International Conference on Computer Vision (ICCV), 2009.

[77] E. Johns, G.-Z. Yang. From images to scenes: Compressing an image cluster into a single scene model for place recognition. In IEEE International Conference on Computer Vision (ICCV), 2011.

[78] Y.-T. Zheng, M. Zhao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T.-S. Chua, H. Neven. Tour the world: Building a Web-Scale Landmark Recognition Engine.In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[79] Y. Li, D. J. Crandall,D. P. Huttenlocher. Landmark classification in large-scale image collections. In IEEE International Conference on Computer Vision (ICCV), 2009.

[80] J. Liang, N. Corso, E. Turner, A. Zakhor. Image based localization in indoor environments. In International Conference on Computing for Geospatial Research and Applications (COM.Geo), 2013.

[81] S. Wang, S. Fidler, R. Urtasun. Lost Shopping! Monocular Localization in Large Indoor Spaces. In IEEE International Conference on Computer Vision (ICCV), 2015.

[82] X.Sun, Y. Xie, P. Luo, L. Wang. A Dataset for Benchmarking Image-Based Localization. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.