Learning Visibility of Landmarks for Vision-Based Localization

Pablo F. Alcantarilla, Sang Min Oh, Gian Luca Mariottini, Luis M. Bergasa, Frank Dellaert

***Abstract*— We aim to perform robust and fast vision-based localization using a pre-existing large map of the scene. A key step in localization is associating the features extracted from the image with the map elements at the current location. Although the problem of data association has greatly benefited from recent advances in appearance-based matching methods, less attention has been paid to the effective use of the geometric relations between the 3D map and the camera in the matching process.**

In this paper we propose to exploit the geometric relationship between the 3D map and the camera pose to determine the visibility of the features. In our approach, we model the visibility of every map feature w.r.t. the camera pose using a non-parametric distribution model. We learn these non-parametric distributions during the 3D reconstruction process, and develop efficient algorithms to predict the visibility of features during localization. With this approach, the matching process only uses those map features with the highest visibility score, yielding a much faster algorithm and superior localization results. We demonstrate an integrated system based on the proposed idea and highlight its potential benefits for the localization in large and cluttered environments.

I. Introduction

We aim to perform fast and robust vision-based localization, using an a-priori obtained accurate 3D map of the environment. This map can be reconstructed by techniques such as bundle adjustment [1], [2] or Visual SLAM [3]. Such as setup is feasible and would be useful in many applications, among others to aid the navigation of the visually impaired, which is our primary motivation. A vision-based localization system would be able to provide an accurate pose aligned with the head orientation, which can enable a system to provide the visually impaired with information about their current position and orientation and/or guide them to their destination through diverse sensing modalities [4].

A key problem in obtaining accurate pose is associating the features extracted from the image with the map elements at the current location. Solving this matching problem becomes very challenging when the map contains a very large number of 3D landmarks, e.g., a few million, and/or has a complex spatial structure. Recent advances in appearance-based methods have improved the matching process for localization substantially. A common trend has been the efficient use of invariant appearance descriptors [5], [6]. Less attention has been paid to the effective use of the geometric relations between both the 3D landmarks and the camera pose to achieve this goal. In many cases, the only use of geometric information can be efficiently used to prune out the 3D landmarks which are beyond the proximity from the latest known camera pose.

A promising avenue that has been explored in the literature is predicting whether a feature will be visible or not in the image, based on a rough estimate of the camera pose. For example, in the visual SLAM work by Davison [3], the visibility of a 3D landmark is predicted based on the similarity of the current camera pose w.r.t. a memorized reference pose, recorded when the landmark is first seen. Although this visibility criterium works reasonably well, each 3D landmark memorizes only one reference camera pose and more complex visibility criteria can not be exploited. In the vein, Sala *et al.* [7] studied the optimal feature selection for robot navigation by means of a graph theoretical formulation where the landmarks are divided into groups with similar visibility. However, the visibility modeling for every feature is still limited since, for a given visibility region only the same set of k features is visible (being k a parameter defined by the pose-estimation algorithm). In [8] the authors show an augmented reality application in which the probability of every feature from which the feature can be tracked successfully is modeled by means of a finite set of Gaussian mixtures. The extension of this method for larger environments is difficult and the use of a Gaussian kernel makes the approach limited as explained in Section III-B. Zhu et. al. [9] studied how to build an optimal landmark database rapidly and use this database online for real-time global localization. It was shown that by using an intelligent subsampling of the landmark database the size of the database can be reduced without dropping the accuracy. In this approach landmarks are characterized by its appearance using Histogram of Oriented Gradients (HOG) [10] and the selection of putative matches for pose estimation relies vastly on appearance descriptors, not exploiting all the geometry information.

In this paper we generalize these approaches and introduce a memory-based learning framework to predict, for each feature, its visibility w.r.t. the varying camera

P.F. Alcantarilla and L.M. Bergasa are with Department of Electronics, University of Alcalá. Alcalá de Henares, Madrid, Spain. e-mail: pablo.alcantarilla, bergasa@depeca.uah.es

Sang Min Oh and F. Dellaert are with School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA. e-mail: sangmin,dellaert@cc.gatech.edu

Gian Luca Mariottini is with Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, USA. e-mail: gianluca@cs.umn.edu

pose. Our approach efficiently captures the visibility of landmarks located in structurally complex or cluttered environments. This is made possible by memorize multiple reference poses for each feature, using non-parametric learning methods. In the remainder of the paper, the overall framework of our 6-DOF vision-based localization system is shown. Then, the importance of the visibility prediction problem is discussed, and our solution based on the non-parametric framework is presented along with an efficient implementation. Finally, the experimental results demonstrate the benefits of the proposed approach. Our results show that with our method, the visibility of 3D landmarks is predicted more accurately and the algorithm improves the quality of the matches as well as of the localization algorithm accuracy.

II. Vision-Based Camera Localization

We describe the localization problem and our overall framework in this section. A map M is commonly defined as a set of high quality landmarks reconstructed from the images. Such a 3D map comprises of the location of each feature and can be obtained through visual SLAM techniques, e.g., [11]. Even though stereo camera may be used for the 3D map reconstruction, we would focus on localization problem based on monocular vision in this paper.

Our localization system can be described within the Bayesian filtering framework [12]. The posterior probability of the current camera pose θ_t (w.r.t. a world reference frame) can be estimated from the prior probability distribution recursively based on the motion model and measurement model as follows:

$$\underbrace{P(\theta_t|Z^{1:t},M)}_{\text{Posterior}} \propto \underbrace{P(z_t|\theta_t,M)}_{\text{Measurement}} \cdot \cdot \int_{\theta_{t-1}} \underbrace{P(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})}_{\text{Motion}} \underbrace{P(\theta_{t-1}|Z^{1:t-1},M)}_{\text{Prior}} d\theta_{t-1} \quad (1)$$

where $Z^{1:t} \equiv z_t$ indicates the sequence of images up to time t. While the framework in Eq. 1 is very general, we implemented the posterior as a single delta function, i.e., only the most likely camera pose is estimated. More in detail, the overall localization system works through the following steps:

a) While the camera is moving, the camera acquires an image z_t from which a set of image features $Y_t \equiv y_{t,k}|1 \geq k \leq |Y_t|$ are detected by a corner detector of choice, e.g., Harris corner detector [13].
b) Then, a promising subset of the map M'_t is chosen and re-projected onto the image plane based on the estimated previous camera pose θ_{t-1} and known camera parameters.
c) Afterwards, a set of putative matches C_t are formed where the i-th putative match $C_{t,i}$ is a pair $\{y_{t,k}, x_j\}$ which comprises of a detected corner $y_{t,k}$ and a map element x_j. A putative match is created when the error in Euclidean norm between a detected feature and a projected map element is very low.
d) Finally, the posterior function in Eq. 1 is fed into RANSAC along with the set of generated putatives C_t, where a set of promising inlier putatives are selected to produce an optimal camera pose θ_t. The RANSAC-based framework described above is very popular and has been used successfully by many authors [14], [15].

As the map M becomes larger and denser, we need to pay special attention in step (c) in order to maintain high-quality putative matches, since the high ratio of outlier putative matches will cause the RANSAC procedure in step (d) to slow down substantially or even fail to compute a correct estimate [16], which will result in under-performing system. This is due to the fact that the number of all the possible putative match pairs (even without assuming any spurious measurement), would increase exponentially w.r.t. the size of the map $|M|$. Hence, unless only the high quality putative matches are selected, a large number of ill-formed putatives will be fed into RANSAC. Traditionally, such high quality putative pairs can be found by using appearance or geometry information. For example, appearance descriptors such as SIFT [5] are used to improve the matching accuracy, or only the map elements $x_i \in M$ in the proximity of the previous pose θ_{t-1} are searched based on the distance (geometry) to reduce matching candidates . In many implemented systems, both approaches are often used simultaneously.

III. Visibility

In this paper, we address the use of geometry information to improve the selection of a promising map subset M'_t in step (b), which turns out to be crucial to improve localization. As mentioned earlier, the process of selecting a subset of map becomes more important as the map becomes larger/denser, and as the environment with substantial degree of occlusions becomes structurally more complex, which is often the case in indoor environments.

The subset of features M'_t can be selected optimally if we know the true *visibility* of features. In other words, a system that can select only the features which are truly visible w.r.t. a pose θ_t in reality is optimal. The visibility is determined by the joint function of the feature x_j and the query pose θ_t. Formally, the visibility $v_j(\theta_t)$ of the j-th 3D landmark $x_j \in M$ w.r.t. a pose θ_t is a boolean variable defined as follows:

$$v_j(\theta_t) = \begin{cases} 1 & \textit{if } x_j \textit{ is visible from } \theta_t \\ 0 & \textit{otherwise} \end{cases} \quad (2)$$

A less optimal system would select a subset which includes additional non-visible features and miss some of the truly visible features. On the other hand, the features Y_t detected from an image would comprise of the features that are in the map or the ones that are

not (due to the change of the environment, or just noisy features). Consequently, the putative match generation process in step (c) will create non-ideal matches as they associate detected image features with incorrect map elements in M'_t. It can be usually assumed that the feature detectors are engineered at its best, hence, the only factor that remains to be improved is the sub-map selection process which should eventually capture the visibility. Otherwise, a naive approach to step (b) would introduce a lot of invisible features into M'_t, which will only become more severe as the map becomes larger and denser. Consequently, if we know the true visibilities for all the features, we can generate high-quality putative matches by considering only the visible features against the detected features on the current image.

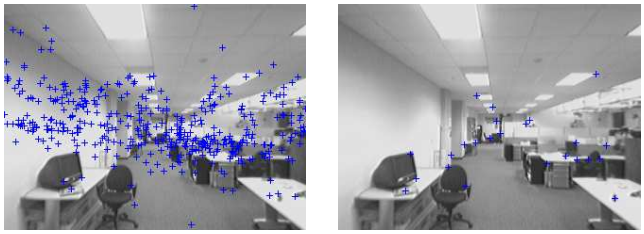
Since the true visibilities are unknown and should be predicted, we can take a probabilistic modeling approach, and pursue to model the following probabilistic visibility distribution $f_j(\theta)$ for every j-th map element w.r.t. a given camera pose θ:

$$f_j(\theta) \equiv P\left(v_j|\theta\right) \tag{3}$$

A. Non-parametric Visibility Modeling

While there can be many different candidate models for the visibility distribution $f_j(\theta)$ in Eq. 3, we propose to use a non-parametric approach [17] where every feature in the map remembers the camera poses which have seen it previously. The use of the non-parametric approach is motivated due to the complex nature of the visibility, which occurs due to the non-trivial geometric structures that exist in the environment. Once the visibility function in Eq. 3 in a non-parametric form is obtained, if a feature has been seen previously by a camera pose similar to the current pose θ_t, such feature is likely to be classified to be visible. However, a major drawback of such non-parametric approach is the computational demand incurred due to the large size of the map M and the size of the set of all the training camera poses stored for every feature in the map.

A major contribution of this paper is a computationally tractable and scalable algorithm which allows us to compute the visibilities for each feature efficiently under the non-parametric framework otherwise infeasible. For example, Fig. 1 (a) depicts an example where the entire map has been re-projected on the image plane. On the other hand, Fig. 1 (b) depicts only the projections of the map elements that are selected based on our non-parametric framework. It can be clearly seen that the often-used nearest-neighbor algorithm for putative matching would perform very poorly in generating high-quality putative matches since the number of back projections dominate the entire image plane due to the dense characteristics of the map. On the other hand, if we try to lower the number of map elements naively by selecting a low distance threshold, it can potentially remove good map elements and may deteriorate the overall quality of the matches nonetheless.



(a) (b)

Fig. 1. The blue crosses depict the re-projections of a 3D point from the map onto the image plane. In (a) the whole map is re-projected, whereas in (b) only the most highly visible features for a given pose are re-projected.

In our work, the visibility function $f_j(\theta)$ is learned by providing a set of training examples and a similarity kernel function $\mathbb{K}$. In more detail, we collect a set of training data (camera poses) Θ_j for each map element where the pose set comprises of both positive examples Θ_j^1 and negative examples Θ_j^0, i.e., $\Theta_j \equiv \left\{\Theta_j^1, \Theta_j^0\right\}$. By positive examples, we mean the camera poses from which the particular map element was visible, and by negative examples the rest of the set. With an appropriate kernel function $\mathbb{K}$, the visibility at a query pose θ_t can be estimated using Bayes' law, as shown in Eq. 4:

$$P(v_j = 1|\theta_t) = \frac{P(\theta_t|v_j = 1)P(v_j = 1)}{\sum\limits_{v_j} P(\theta_t|v_j)P(v_j)} \tag{4}$$

$$P(\theta_t|v_j) = \frac{1}{\left|\Theta_j^{v_j}\right|} \sum_i \mathbb{K}(\theta_t, \Theta_{j,i}^{v_j}) \tag{5}$$

In particular, the likelihood factor $P(\theta_t|v_j)$ is modeled by the non-parametric training examples where the likelihood factor for the positive case is shown in Eq. 5. In general, the kernel function $\mathbb{K}$ should be chosen in such a way that the similarity output value of the kernel function decreases as the query pose θ_t stays further away from the training data.

Since the size of the training data Θ_j for every feature can be still very large, which would incur substantial computational demand, we estimate an approximate visibility based on K-Nearest-Neighbor (KNN) approach w.r.t. the latest pose estimate θ_t. This is possible due to the fact that only the training data nearby the query pose θ_t are informative to correctly compute the visibility in Eq. 4 in most cases. As a consequence, we come to have a succinct visibility estimation function in Eq. 6 where only the nearest K samples $\Theta_j^K = \left\{\Theta_j^{K,1}, \Theta_j^{K,0}\right\}$ would be considered.

$$P(v_j = 1|\theta_t) \approx \frac{\sum\limits_i \mathbb{K}\left(\theta_t, \Theta_{j,i}^{K,1}\right)}{\sum\limits_{v_j}\sum\limits_i \mathbb{K}\left(\theta_t, \Theta_{j,i}^{K,v_j}\right)} \tag{6}$$

While the problem of obtaining a training dataset for the above problem does not look trivial, which seems to be almost unrealistic to collect such data manually

for every single map elements, we can notice that the training data exists already from the moment we built a map M. As either the bundle-adjustment technique or a SLAM algorithm is used to create a map, the intermediate process of these algorithms is to provide the information on the features seen by every estimated camera poses in the dataset. Hence, for every feature, we exactly know which camera poses observed it, which are collected as positive training data. The rest of the camera poses are collected as negative training dataset. Note that the map building approaches above do not guarantee that all the positive data are collected. In fact, it is expected that some portion of the positive camera poses actually slip into negative training set, since the map-building approaches fails to find all the correct correspondences.

B. Learning Kernel Functions

In this section, we describe the necessary properties of a good kernel function for visibility modeling and the details of the function we have specially developed. In particular, our kernel function is designed to output high similarity between cameras with similar views, i.e., the image features they see overlap substantially, and the parameters of the function can be optimized based on the training dataset. There may be many candidate kernel functions to define the similarity between two poses θ_0 and θ_1: $\mathbb{K}(\theta_0, \theta_1)$. Naive use of popular kernels such as Gaussian kernel would not be very useful since it does not consider the view overlap effectively. To see why, consider the case where the roll of the two cameras are substantially different while the other 3D coordinates, yaw, and pitch are identical. Although a Gaussian kernel function would produce output which shows substantial difference, actually the two cameras share very similar views, and consequently, similar set of image features. Hence, the kernel function we used incorporates (1) Euclidean distance between the cameras $d(\theta_0, \theta_1)$ and (2) the normalized inner-product between the viewing direction between the two cameras $d'(\theta_0, \theta_1)$ as the two distance inputs. Then, a logistic function was learned from the transformed two dimension data:

$$\mathbb{K}(\theta_0, \theta_1) = \frac{1}{1 + \exp\left(-(w_d \cdot d(\theta_0, \theta_1) + w'_d \cdot d'(\theta_0, \theta_1) - w_o)\right)} \tag{7}$$

Now, the parameters w_d, $w_{d'}$ and w_o (wich is an offset parameter) of the kernel function can be learned from a training dataset where each training datum comprises of two camera poses with their view similarity scores which ranges between 0 and 1. Such view similarity scores can be readily obtained as the proportion of the number of features observed by the two camera poses w.r.t. the total number of features seen by the cameras. For example, if θ_0 observed features $\{x_1, x_2\}$ and θ_1 observed features $\{x_1, x_3, x_4\}$, then the view similarity score is: $\frac{|\{x_1,x_2\} \cap \{x_1,x_3,x_4\}|}{|\{x_1,x_2\} \cup \{x_1,x_3,x_4\}|} = \frac{1}{4}$.

C. Fast Visibility Prediction for Large Maps

We introduce the core technique to decrease computational demand to compute the visibilities for all the map elements dramatically from the order of the map size $O(M)$ to $O(K)$ where K denotes the number of the neighbors in the KNN described. A naive way to compute the visibilities for all the map elements is to go over every one of them one by one computing the visibilities using Eq. 6. The computational demand for such approach increases linearly w.r.t. the size of the map $|M|$. However, the core observation is that the results of the visibility prediction using Eq. 6 will be mostly zero, since most of the map elements in a large map would not be observed at all by the KNNs of the current query pose θ_t, effectively making the numerator in Eq. 6 to be zero.

As a consequence, once we find the KNNs of the current query pose, we only need to predict the visibilities for the subset of map elements which are at least seen once by these KNNs. Then, we can set the visibilities to be zero for the rest of the map elements, even not computing them at all. More formally, the visibilities need to be computed for such map elements which are in the union set $\{\bigcup X_i\}$ where X_i denotes the set of map elements seen by the i-th neighbor pose among KNNs. If we assume that the number of map elements observed by any pose is bounded by a constant c, then the overall number of map elements to be examined is $O(cK) = O(K)$. In summary, we can prune out most of the map elements from the visibility computation process by considering only those which were seen by KNNs of the latest camera pose.

IV. Results and Discussion

We designed our experiments in such a way that we can analyze the benefits that are obtained by the *smart* use of the geometric information. In other words, any benefits to be demonstrated in this paper are expected to be obtained from the improvements on the geometric information use, isolated from any advantages due to appearance-based matching methods. A 3D map in a dense cluttered environment was computed using Visual SLAM techniques, which is used to provide testbeds for localization as well as to provide training data for non-parametric visibility functions. During the map computation process, the different training poses from which each map element is seen, are memorized so as to be able to predict the visibility by means of a non-parametric modeling during the localization stage. We show monocular vision-based localization results for the training and test datasets. Additionally, to stand out the contributions of our method, we compare our idea with two different methods:

- **Brute Force**: Under this assumption all the map features are re-projected onto the image plane for a given pose. Besides, only the features that are predicted to lie within the image plane, are consid-

ered to form the set of putatives to be used for pose estimation.

- **Length and angle heuristic**: Feature visibility is calculated considering the difference between the viewpoint from which the feature was initially seen and a new viewpoint. This difference in viewpoint has to be below some length and angle ratio, and predicted to lie within the image, in order to predict the feature as visible. Usually the feature is expected to be visible if the length ratio $|h_i|/|h_{orig}|$ is close enough to 1 (in practice between 5/7 and 7/5 and the angle difference $\beta = \cos^{-1}((h_i \cdot h_{orig})/(|h_i||h_{orig}|))$ is close to 0 (less than 45° in magnitude).

Considering the first of the above approaches, a *brute force* approach will yield in a high number of outliers and localization errors under very dense maps (thousands of features), whereas the second of the approaches can yield erroneous localization when the camera is facing occluded areas. The second approach has been widely used in the literature such as in [18], [3], since it is very easy to compute and provides good results, since after the visibility prediction, matching is performed using 2D image templates. However, one of the drawbacks of this criteria is that considering only geometry information, this heuristic can not deal properly with occlusions. This in fact, can be a problem for long term localization under cluttered environments with occlusions, such as the ones we are interested for the visual impaired (e.g. cities, underground stations, offices, etc.)

The training dataset is a large sequence of 5319 frames in which a dense map with about 1316 3D points was obtained. The test dataset is a separate small sequence recorded in the same space and comprises of 2477 frames. Fig. 2 depicts the kind of environment where we have tested our localization experiments.

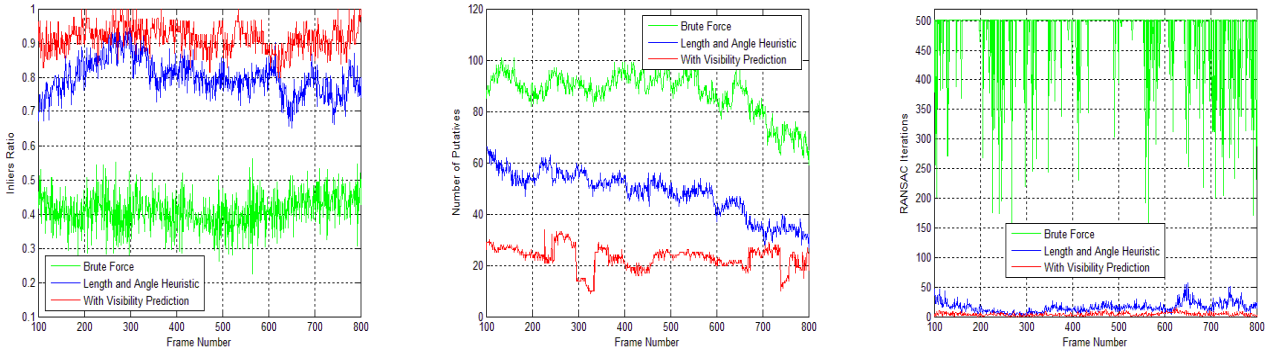
In our experiments we use an adaptive threshold version of RANSAC to automatically determine the number of RANSAC iterations needed [16]. The distance threshold that is used to create a putative match is set to 4 pixels. In addition we only consider visibility prediction of the 60 KNNs since we have found experimentally that this number of neighbors is enough for predicting visibility. Our experimental results shown in Fig. 3 highlights the benefits of our method where our result is shown to provide less number of higher-quality putative matches, which eventually leads to faster and more accurate RANSAC computation. In detail, Fig. 3 depicts the inliers ratio (a), the number of putatives per frame (b) and the number of RANSAC iterations (c) during some of the first frames of the test sequence. The highest inliers ratio is obtained using our visibility prediction approach and this ratio is normally above 80%. In our experiments, we set the number of RANSAC iterations to a maximum of 500 for computational purposes. As it can be seen in Fig. 3, the number of iterations for the *brute force* case is very close to this bound, whereas for the other experiments the number of iterations is much lower, obtaining less than 20 RANSAC iterations per frame for the visibility case.

In Table I, the information about the mean inliers ratio, mean number of putatives and RANSAC iterations per frame is shown for the training and test datasets respectively where we can again observe the computational benefits our method. Fig. 4 and 5 show the localization results for the training and test sequence respectively, w.r.t. the ground truth data obtained by Visual SLAM and bundle adjustment optimization. Map features are characterized by a vector $Y_i = \{X\ Y\ Z\}$ in a world coordinate frame (map features are represented by orange dots in the next figures). Each camera poses is parametrized by means of a vector $\theta_i = \{X\ Y\ Z\ q_0\ q_x\ q_y\ q_z\}$ (translation and orientation given by a unit quaternion). We don't show any figure results of the brute force case, since this approach fails to provide accurate localization for both datasets.

Finally, we show the overall localization accuracy of our monocular-vision system for the camera locations and rotations. In detail, Table II and III show the mean squared error with respect to the ground truth of the estimated localization for both translation and orientation for training and test sequences respectively. In the test sequence, the camera goes straight on during approximately 1000 frames into a corridor where no occlusions are present, and after that it turns right into an area with severe occlusions, yielding a wrong localization result since these two approaches are not able to estimate correctly the 180° rotation in the area with occlusions. On the contrary, with our approach, localization results are very similar to the ones obtained in the ground truth even in areas of the map with a dense level of occlusions. Besides, the number of RANSAC iterations per frame that are necessary are less than 20, showing that the method can work in real time providing good localization estimates. With our approach we have obtained the smallest errors with respect to the other two methods, both in translation and rotation components. Results are quite satisfactory, taking into account that appearance descriptors have not been used in the matching process. We think that the use of appearance descriptors combined with our visibility prediction will improve and speed up localization.

As it has been shown, our method depends on the quality of the input data. But how many training views are necessary to obtain accurate and fast localization results? Considering a huge number of training views can be an overwhelming computational burden for very large environments such as buildings, or even cities. We have done some experiments in which we reduce considerably the number of training views (sampling uniformly among the whole dataset of training views), and run our localization algorithm predicting features visibility. We have studied several localization runs with a different number of training views: %100, %70, %50, %30 and

Fig. 2. Some frames of the analyzed environment



(a) (b) (c)

Fig. 3. Test sequence results: (a) inliers ratio, (b) number of putatives and (c) number of RANSAC iterations per frame

Training	**% Inliers**	**# Putatives**	**# Iterations**
Brute Force	0.6952	67.2181	461.7096
Heuristic	0.7392	36.7744	52.7885
Visibility Prediction	0.8335	26.3897	6.4221
Test	**% Inliers**	**# Putatives**	**# Iterations**
Brute Force	0.6420	74.2782	439.0642
Heuristic	0.6817	36.9931	84.3254
Visibility Prediction	0.7368	16.7587	17.6437

TABLE I

INLIERS RATIO, NUMBER OF PUTATIVES PER FRAME FOR TRAINING AND TEST SEQUENCES

Case	**Training** ϵ_x **(m)**	**Training** ϵ_y **(m)**	**Training** ϵ_z **(m)**	**Test** ϵ_x **(m)**	**Test** ϵ_y **(m)**	**Test** ϵ_z **(m)**
Brute Force	3.4857	0.0974	1.8305	1.1825	0.1230	1.3366
Heuristic	0.5642	0.0574	0.4142	1.1549	0.0954	0.5041
Visibility Prediction	0.0476	0.0243	0.0340	0.2781	0.0785	0.2736

TABLE II

LOCALIZATION ERRORS IN TRANSLATION WITH RESPECT TO GROUND TRUTH

Training	ϵ_{q_0}	ϵ_{q_X}	ϵ_{q_Y}	ϵ_{q_Z}
Brute Force	0.2180	0.1030	0.3258	0.0497
Heuristic	0.2222	0.0415	0.1911	0.0264
Visibility Prediction	0.0068	0.0106	0.0100	0.0091
Test	ϵ_{q_0}	ϵ_{q_X}	ϵ_{q_Y}	ϵ_{q_Z}
Brute Force	0.2484	0.0488	0.2367	0.0346
Heuristic	0.1826	0.0452	0.1561	0.0304
Visibility Prediction	0.0516	0.0366	0.0476	0.0237

TABLE III

LOCALIZATION ERRORS IN ROTATION WITH RESPECT TO GROUND TRUTH

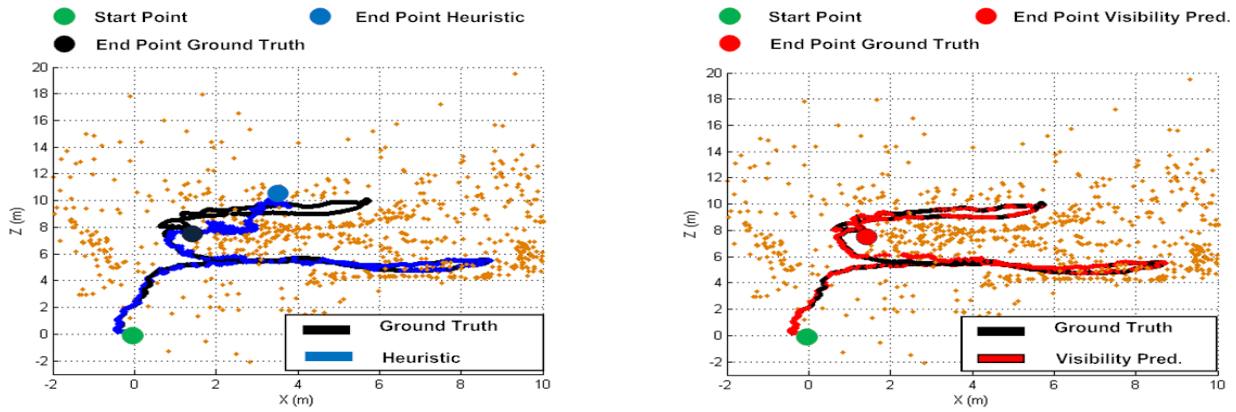


Fig. 4. Comparison of Localization Results for Training Sequence: (a) Heuristic (b) With Visibility Prediction

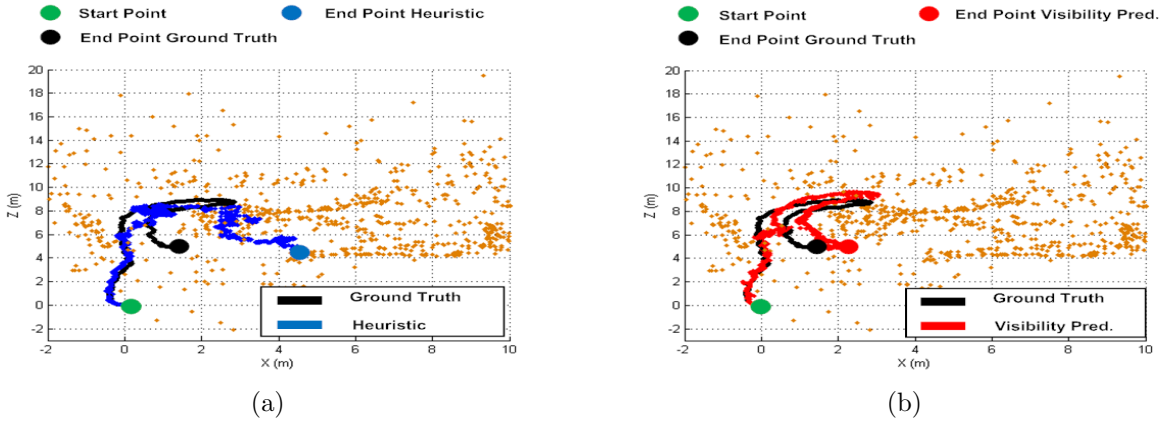


Fig. 5. Comparison of Localization Results for Test Sequence: (a) Heuristic (b) With Visibility Prediction

%10. In Tables IV ratios about the inliers ratio, number of putatives and RANSAC iterations per frame are shown for both the training and test dataset respectively.

The inliers ratio is similar for all the experiments, being higher for the %100 of training views, since this is the case in which we have the highest level of detail of the environment 3D structure. As long as we reduce the sampling rate, keeping the same number of KNNs, the distance between the current camera pose and its nearest neighbors increases so it is easier to predict a feature to be visible when in fact it is not really visible from the current camera pose. This is the reason why the number of iterations that RANSAC needs to fit the best model increases as long as we reduce the sampling rate. In terms about the difference in localization results is very similar between all the cases, and even localization results with only a %10 of the training views are better than the brute force and heuristic approaches both for training and test sequence.

In our experiments the acquisition frame rate of the training sequence was 30 frames per second, the image resolution was 320×240 pixels and camera was carried in hand by a person at normal walking speeds $(3Km/h - 5Km/h)$. According to the results it seems that our localization algorithm can provide good results with a considerably smaller number of views than the whole dataset of training views. This is an important factor for large environments since we don't have to keep in memory the whole training dataset. However, instead of performing a simple heuristic sampling, it would be of benefit just selecting the views that give more information about the 3D reconstruction just in a similar way as it is explored in [19] for data association.

V. Conclusions

We have presented an algorithm for predicting the highly visible features for real time localization under indoor environments, in which every map feature models its visibility w.r.t. the camera poses via non-parametric distributions. We have shown the benefits of our method for monocular vision-based localization. The localization errors considering our visibility prediction are very small compared to the ground truth and also the speed-up gain compared to the other analyzed methods is very significant. Besides, our procedure can provide accurate localization results even when a small set of training views is used.

As future work we are interested in integrating our

Sequence	% Camera Poses	% Inliers	# Putatives	# Iterations	# Training Views
Training	100	0.8335	26.3897	6.4221	5319
Training	70	0.8331	30.3055	6.8429	3723
Training	50	0.8158	32.2688	8.0959	2661
Training	30	0.7803	36.2432	8.6119	1569
Training	10	0.7664	42.5013	11.2367	533
Test	100	0.7368	16.7587	17.6437	5319
Test	70	0.7194	20.0803	19.5891	3723
Test	50	0.7170	25.0831	22.5293	2661
Test	30	0.6983	26.3317	27.3240	1596
Test	10	0.6510	29.4727	30.3448	533

TABLE IV

INLIERS RATIO, NUMBER OF PUTATIVES AND RANSAC ITERATIONS PER FRAME CONSIDERING DIFFERENT NUMBER OF TRAINING VIEWS

approach into a real-time SLAM system, i.e. when the map has been computed and the remaining uncertainty of the features are below a quality threshold, the map can be used for navigation purposes using the *learnt* visibility from the mapping process. In addition, the use of powerful appearance descriptors and our visibility prediction idea can provide very good localization results in cluttered environments with occlusions for indoor and outdoor environments. Obtaining the camera views that have more impact in the reconstruction, we can sample the set of training poses reducing the total size of poses to be kept in memory. In the same way we are interested in techniques such as the one described in [9] for reducing the total number of landmarks that are necessary for an accurate localization.

VI. Acknowledgements

This work was supported in part by the Spanish Ministry of Education and Science (MEC) under grant TRA2008-03600 (DRIVER-ALERT Project) and by the Community of Madrid under grant CM: S-0505/DPI/000176 (RoboCity2030 Project).

References

[1] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment – a modern synthesis," in *Vision Algorithms: Theory and Practice*, ser. LNCS, W. Triggs, A. Zisserman, and R. Szeliski, Eds. Springer Verlag, Sep 1999, pp. 298–375.

[2] K. Konolige and M. Agrawal, "Frame-frame matching for realtime consistent visual mapping," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Apr 2007, pp. 2803–2810.

[3] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, no. 6, 2007.

[4] B. N. Walker and J. Lindsay, "Navigation performance with a virtual auditory display: Effects of beacon sound, capture radius, and practice," *Human Factors*, vol. 48, no. 2, pp. 265–278, 2006.

[5] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Intl. J. of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[6] S. Se, D. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *Intl. J. of Robotics Research*, vol. 21, no. 8, pp. 735–758, Aug 2002.

[7] P. Sala, R. Sim, A. Shokoufandeh, and S. Dickinson, "Landmark selection for vision-based navigation," *IEEE Trans. Robotics*, vol. 22, no. 2, pp. 334–349, April 2006.

[8] H. Wuest, A. Pagani, and D. Stricker, "Feature management for efficient camera tracking," in *Asian Conf. on Computer Vision (ACCV)*, 2007.

[9] Z. Zhu, T. Oskiper, S. Samarasekera, R. Kumar, and H. Sawhney, "Real-time global localization with a pre-built visual landmark database," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.

[10] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[11] L. M. Paz, P. Piniés, J. D. Tardós, and J. Neira, "Large scale 6DOF SLAM with stereo-in-hand," *IEEE Trans. Robotics*, vol. 24, no. 5, 2008.

[12] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT press, Cambridge, MA, 2005.

[13] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[14] D. C. K. Yuen and B. A. MacDonald, "Vision-based localization algortihm based on landmark matching, triangulation, reconstruction, and comparison," *IEEE Trans. Robotics*, vol. 21, no. 2, pp. 217–226, 2005.

[15] S. Tariq and F. Dellaert, "A multi-camera 6-DOF pose tracker," in *IEEE and ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR)*, 2004, pp. 296–297.

[16] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[17] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2007.

[18] A. J. Davison and D. W. Murray, "Simultaneous localization and map-building using active vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 7, 2002.

[19] M. Kaess and F. Dellaert, "Covariance recovery from a square root information matrix for data association," *Journal of Robotics and Autonomous Systems*, 2009, in print. [Online]. Available: http://frank.dellaert.com/pub/Kaess09ras.pdf