Accelerated Robust Point Cloud Registration in Natural Environments through Positive and Unlabeled Learning

Maxime Latulippe, Alexandre Drouin, Philippe Giguère and François Laviolette
Laval University, Quebec, Canada
{maxime.latulippe.1, alexandre.drouin.8}@ulaval.ca, {philippe.giguere, francois.laviolette}@ift.ulaval.ca

Abstract

Localization of a mobile robot is crucial for autonomous navigation. Using laser scanners, this can be facilitated by the pairwise alignment of consecutive scans. In this paper, we are interested in improving this scan alignment in challenging natural environments. For this purpose, local descriptors are generally effective as they facilitate point matching. However, we show that in some natural environments, many of them are likely to be unreliable, which affects the accuracy and robustness of the results. Therefore, we propose to filter the unreliable descriptors as a prior step to alignment. Our approach uses a fast machine learning algorithm, trained on-the-fly under the *positive and unlabeled* learning paradigm without the need for human intervention. Our results show that the number of descriptors can be significantly reduced, while increasing the proportion of reliable ones, thus speeding up and improving the robustness of the scan alignment process.

1 Introduction

To accomplish autonomous navigation in unknown environments, a mobile robot must be able to localize itself. Proprioceptive sensors, such as odometry and inertial units, cannot be used alone to precisely keep track of the long-term robot pose, as they are subject to drift. Global localization systems like GPS can fulfill the task, but in some environments they may not be available. An alternative is to use a 3D laser scanner to exploit information from the surrounding environment in order to estimate the robot's pose. Although being well handled in structured environments (corridors, buildings), this problem remains difficult in unstructured environments like wood or planetary terrains.

To localize using laser scanners, consecutive scans (*point clouds*) taken with these sensors can be registered (*i.e.* aligned) in a pairwise manner to estimate the motion performed by the robot at each step. This motion, decomposed in a translation T and a rotation R, is often computed using the popular Iterative Closest Point (ICP) algorithm [Besl and McKay, 1992] or one of its variants [Rusinkiewicz and Levoy, 2001]. However, because ICP is subject to the problem of local minima, a prior coarse alignment of the point clouds increases the likelihood of converging towards the true solution.

This coarse alignment can be performed using measurements from proprioceptive sensors, although they may not be reliable enough in some situations (e.g. on a slippery terrain for odometry). In such cases, the coarse alignment must be determined from the point clouds themselves. For this matter, one can compute local descriptors (set of values) depicting the geometry of each point's vicinity in the scans and match the similar descriptors to establish correspondences between scans. Sample Consensus Initial Alignment (SAC-IA), presented in [Rusu *et al.*, 2009], is an adaptation of Random Sample Consensus (RANSAC) [Fischler and Bolles, 1981] to the point cloud registration problem and is based on this idea. Points are randomly picked in the source cloud and matched randomly within a list of the most potentially corresponding points (most similar descriptors) in the target cloud. For 3D point clouds, three correspondences are thus established and the geometrical transformation R and T that best aligns these corresponding points is computed. The process is repeated for N iterations and the transformation that yields the smallest alignment error on the clouds is kept. The accuracy of this coarse alignment therefore depends on the quality of the descriptors extracted, as more discriminative and robust descriptors should lead to more valid correspondences. On the other hand, with a descriptor set of poorer quality, the algorithm statistically requires a greater number of iterations to find an alignment with the same accuracy.

In this paper, we address the coarse registration problem in natural unstructured environments, including those featuring dense vegetation. Section 3 explains the issues associated with descriptors in this type of environment. We aim at increasing the robustness of SAC-IA, as well as its computation efficiency, by filtering the unreliable descriptors beforehand. Our approach, based on *positive and unlabeled* learning, is described in Section 4. Section 5 details the experimentations and results, and section 7 concludes.

2 Related work

Many types of descriptors have been proposed over the years. The interested reader can refer to [Tangelder and Veltkamp, 2008] for a survey. Some description methods extract keypoints as a prior step. The well-known Scale-Invariant Feature Transform (SIFT) descriptor [Lowe, 2004] includes such key-

point extraction. It has been extended to 3D data [Scovanner *et al.*, 2007], but requires intensity or RGB values for its computation. Normal Aligned Radial Feature (NARF) [Steder *et al.*, 2011] also contains a keypoint extraction step, but applies to range images and thus cannot be computed directly on laser scans. In this paper, we will focus on Fast Point Feature Histograms (FPFH) [Rusu *et al.*, 2009] and Signature of Histograms of Orientations (SHOT) [Tombari *et al.*, 2010], as these descriptors are state-of-the-art and directly computable on laser scans. As it will be demonstrated in Section 3, their effectiveness in natural unstructured environments is limited. With a similar goal to ours, Rusu *et al.* [2009] suggested a method to filter the FPFH descriptors after their computation, such that only the most useful are used for matching. However, this method is not well adapted to forested environments. This will be discussed in Section 3.

It is worth noting that Song *et al.* [2012] recently presented a registration method, based on tree trunks extraction, to align point clouds acquired in forested environments. While effective, this method assumes that at least a few approximately parallel trunks exist and are well visible in the environment, conditions that are not always met.

The concept of a mobile robot collecting sensor data in order to train an autonomous navigation system has already been studied. For example, methods have been proposed to learn appearance models of drivable surfaces in desert terrain [Dahlkamp *et al.*, 2006], to learn rover-terrain interaction models from experiments [Krebs *et al.*, 2010] and to perform surface slip prediction using learning from experience [Angelova *et al.*, 2006]. Our approach is similar in the sense that the robot trains itself, without the need for human intervention, based on data that is collected on the run. Machine learning has also shown to be applicable to descriptors. Sim and Dudek [1999] presented a method to evaluate and learn, for a specific environment, a set of visual landmarks useful for pose estimation. Grabner *et al.* [2007] learned keypoint descriptions to improve object tracking. On the descriptor representations side, machine learning has been used to find the optimal parameters of new image descriptor representations using the DAISY configuration [Winder *et al.*, 2009] and using a more complex set of predefined building blocks [Brown *et al.*, 2011]. Trzcinski *et al.* [2012] applied boosting to find a non-linear mapping of image intensity patches into a feature space. To the best of our knowledge, machine learning has not been employed yet for filtering 3D point cloud descriptors in order to keep the most reliable ones in certain types of environment.

3 Descriptors in natural environments

In natural unstructured environments, only a subset of all descriptors are likely to be reliable for point matching. In fact, many of them are either too common or unstable and hence lead to mismatches. In the case of forested environments, the foliage and branches are often undersampled by the laser scanner. A small viewpoint change can then result in a significant variation of the measured points. Thus, the descriptors of points located in these parts of the environment are inherently unstable. For example, Fig. 1 shows correspondence estimation results between two consecutive clouds of the *Hannover* dataset [Wulf *et al.*, 2008], from the Robotic 3D Scan Repository, using FPFH and SHOT descriptors. For both examples, the matching threshold was selected at the elbow point of the precision and recall curve ($t_{match} = t_{elbow}$). The matching precision is about 20% for the FPFH and 25% for the SHOT. These results show that few valid correspondences can be established using descriptors of points located in the foliage of the trees. Thus, filtering these unreliable descriptors can increase the matching performance. This, in turn, increases the probability of success of the SAC-IA algorithm, leading to an improvement in performance and computation efficiency.

For the FPFH descriptors, a filtering method is proposed by Rusu et al. [2009] to select the most useful descriptors for matching and registration. In their case, the useful descriptors are the less common. They suggest to approximate the distribution of all descriptors distances to the mean descriptor, μ, by a gaussian distribution $\mathcal{N}(\mu, \sigma)$. The descriptors whose distances fall outside the $\mu \pm \beta \cdot \sigma$ interval, where β is the filter threshold, are called *unique* and are retained. The others are discarded. Although this method insures that the remaining descriptors are salient in the environment, it does not guarantee their stability or reliability. In particular, for natural environments, the noisy descriptors in the foliage and branches are likely to differ significantly from each other, thus making them appear as unique. This leads to a significant amount of unreliable descriptors in the subset used for matching. For example, Fig. 2 shows the result of this method on the same point clouds as above, using $\beta = 1.2$ and $t_{match} = t_{elbow}$. In this case, the matching precision increased at about 28%, but with a recall of only 1.9%. Again, few descriptors in the foliage led to valid matches. A further analysis of the effect of this filter is presented in Fig. 3. This plot shows the proportion, as a function of β, of the unique FPFHs that established valid correspondences between the scans (valid FPFHs). For comparison purposes, the figure also presents the results for the well-known Standford Bunny[1], which has a relatively smooth and well sampled surface. In the case of the Bunny, this filtering method clearly improves the proportion of valid FPFHs as β increases. Oppositely, in the case of *Hannover*, we see that, as the value of β increases, the descriptors leading to valid matches are filtered out faster than the others. These results therefore indicate that this method is not suitable when very irregular surfaces are present, as typically seen in foliage.

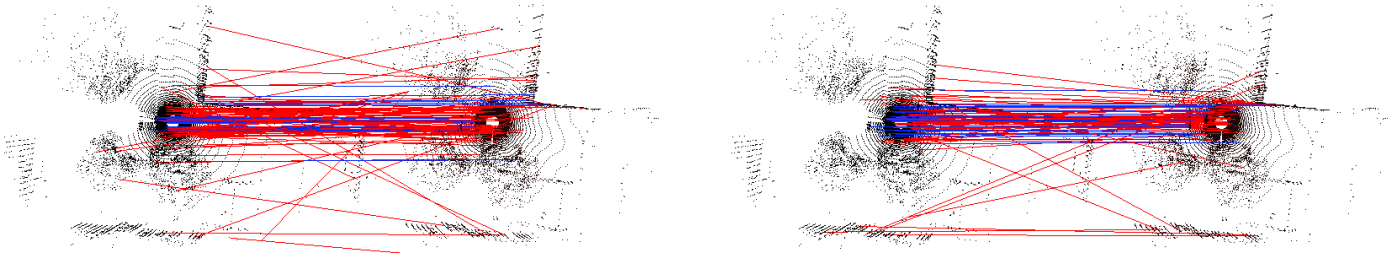
4 Method

In order to reduce the number of unreliable descriptors, we propose to use a classifier trained on-the-fly. The training process can be repeated as needed, allowing the robot to naturally adapt its classifier to a changing environment as it travels.

4.1 Training Data

In our proposed approach, the training dataset and labeling is generated by the robot itself. For very short travel distances, we assume that odometry can provide a close estimate of the robot's true motion. Consequently, two consecutive laser scans, A and B, can be registered using odometry, and this

[1] Available at http://graphics.stanford.edu/data/3Dscanrep/.



(a) FPFH. (b) SHOT.

Figure 1: Estimated correspondences between scans 22 and 23 of *Hannover*, using FPFH and SHOT descriptors. Blue and red lines show valid and wrong correspondences respectively. Shown correspondences are downsampled (1/80).

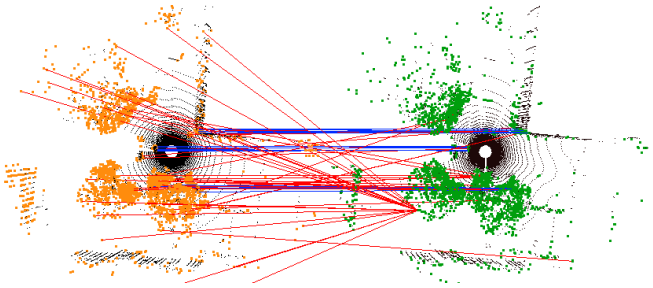


Figure 2: Estimated correspondences (downsampled to 1/10) between scans 22 and 23 of *Hannover* using unique FPFHs and $\beta = 1.2$ (orange and green dots are unique).

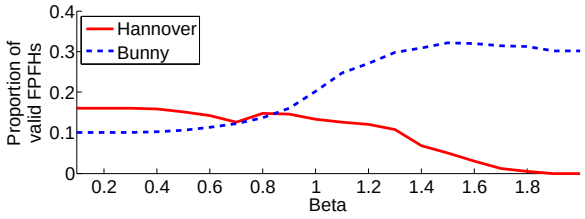


Figure 3: Effect of β on the proportion of the unique FPFHs that led to valid matches for the *Hannover* and Bunny datasets.

registration can be used to automatically label descriptors[2]. The idea behind this is to have two slightly different viewpoints of the objects present in the environment. Doing so, objects that are relatively smooth will have a stable appearance in both scans, hence similar descriptors. Oppositely, objects having an irregular surface (such as foliage) will have unstable descriptors, due to the limited spatial sampling of the laser, and will mostly lead to mismatches.

Labeling starts by matching descriptors i and k between scans A and B on a nearest neighbor basis in feature space. The corresponding point coordinates of these descriptors, with the origin defined as the position of the laser scanner in A, are $\mathbf{p}_{i,A}$ and $\mathbf{p}_{k,B}$. The descriptors in scan A are then labeled *known positive* if they are sufficiently close (below thresholds) in physical space to their corresponding descriptors in B, with a distance metric that takes into account the sensor model of the laser scanner. The two criteria used are the angular error ϕ_{err} between the points:

$$\phi_{err} \stackrel{\text{def}}{=} \cos^{-1}\left(\frac{\mathbf{p}_{i,A}\cdot\mathbf{p}_{k,B}}{\|\mathbf{p}_{i,A}\|\|\mathbf{p}_{k,B}\|}\right) < \phi_{max} \tag{1}$$

and the range error r_{err}:

$$r_{err} \stackrel{\text{def}}{=} |\|\mathbf{p}_{i,A}\| - \|\mathbf{p}_{k,B}\|| < d_{max}. \tag{2}$$

The thresholds ϕ_{max} and d_{max} are selected based on the laser scanner error model and the small odometry alignment error. Descriptors in scan B are not used for training, other than for identifying the reliable descriptors in A.

For the descriptors that fail any of the two criteria above, we argue that it would be wrong to consider them as negative examples. In fact, due to the sensor's noise, descriptors belonging to the same parts of the environment may have slightly different values. The true correspondences could fall into the second or third nearest neighbor (in feature space), a situation which is well handled by SAC-IA. Moreover, different viewpoints between scans imply that some objects may not be visible in both of them. In these cases, the true correspondences cannot be established, even if the descriptors belonging to those objects are reliable. For these reasons, we will consider the descriptors that failed our test as *unlabeled* examples, since they may contain *hidden positive* examples. A semi-supervised machine learning paradigm that is well suited to address this problem is *positive and unlabeled* learning. This paradigm is presented in the following subsection.

4.2 Positive and Unlabeled Learning

Positive and unlabeled learning (PU learning) is a variant of the semi-supervised learning paradigm that is well suited for situations where the data is available under the form of positive and unlabeled examples [Elkan and Noto, 2008].

Let $S \subseteq X \times L = \{(x_1,s_1),(x_2,s_2),...,(x_m,s_m)\}$ be a set of m training examples where X is the input space of the training examples and $L = \{l,u\}$. Examples for which $s = l$ are called *known positives* and the ones for which $s = u$, *unlabeled*. In this framework, each input example x_i is also related to a (possibly unknown) label $y_i \in \{+1,-1\}$, such that if $s_i = l$

[2]Note that odometry is used here as a shortcut to another registration technique. For example, it would be possible, albeit much slower, to compute this alignment using ICP with a high number of iterations and random restarts. Odometry is no longer used afterwards and can be arbitrarily bad (slip).

then $y_i = +1$ and if $s_i = u$, the label y_i can be either positive or negative.

We aim to perform binary classification between positive and negative examples in a supervised learning fashion. Therefore, we must take into account that some positive examples may be hidden in the unlabeled examples.

Algorithms like the One Class SVM [Schölkopf *et al.*, 2001] address this type of problem by using only the known positive examples to train an estimator of $p(y = +1 \mid x)$. However, in contrast with PU learning, this approach does not make full use of the available data, since the information contained in the unlabeled data is discarded by the algorithm.

A method called PosOnly, proposed by Elkan and Noto [2008], allows to adapt to PU learning any supervised learning algorithm that outputs probabilistic classifiers. Such a classifier provides an estimation of the probability that an example has a certain label. This method is based on the nontrivial result that:

$$p(y = +1 \mid x) = \frac{p(s = l \mid x)}{c} \tag{3}$$

where $p(s = l \mid x)$ is the probability that an example x is a known positive and $c \stackrel{\text{def}}{=} p(s = l \mid y = +1)$ is the probability that an example is a known positive given the fact that it is a positive one.

Their method goes as follows. First, a subset of examples $V \subseteq S$ is retained as a hold out set. Then, a probabilistic estimator of $p(s = l \mid x)$, $g \in X \to [0, 1]$, is trained using $S \backslash V$. Then c is estimated as follows:

$$c \approx \frac{1}{|G|} \sum_{x \in G} g(x) \tag{4}$$

where G is the set of all known positive examples in V. Finally, from Equation (3), Equation (4) and g, an estimator of $p(y = +1 \mid x)$ is obtained. We thus have a classifier of positive and negative examples that was trained using positive and unlabeled examples only.

4.3 Proposed learning algorithm

While selecting an appropriate learning algorithm, we attempt to exploit the nature of the learning problem. Suppose that the ratio of positive examples hidden in the set of unlabeled examples U is γ. Then, as the value of γ increases, the noise introduced by the positive examples hidden in the U also increases. Therefore, we can expect that the trained predictors will be affected by the value of γ.

Based on this observation, Mordelet and Vert [2010; 2013] proposed to use bootstrap aggregating (bagging) [Breiman, 1996] to learn predictors from positive and unlabeled examples. Bagging consists in training a set of predictors and aggregating their predictions. All such predictors are trained using the same learning algorithm but on different training sets. These training sets are obtained from the original dataset by sampling with replacement; this is what we call a bootstrap sample. As pointed out by Breiman [2001], this method works well when the predictors are not correlated. Random sampling with replacement from U is likely to yield different values of γ. Hence, as pointed out by Mordelet and Vert, we can therefore expect to obtain uncorrelated predictors.

The random forest algorithm, proposed by Breiman [2001], is of bagging type. It also has the appealing property of being highly scalable. This is a suitable property for our problem, since laser scans often contain hundreds of thousands of points and since calculations must be done online. Because of this requirement, algorithms that we found to be time consuming, like bagging SVM [Mordelet and Vert, 2013], were left aside in the favor of random forest.

Our algorithm slightly differs from the original random forest algorithm in its bootstrap sampling phase: we sample $|P|$ examples from P with replacement and $|P|$ examples from U with replacement, instead of randomly sampling $|P \cup U|$ examples from $P \cup U$ with replacement. This approach is similar to the one proposed by Mordelet and Vert [2010; 2013], in the sense that it gives balanced classes to the learning algorithm and that it reduces the training time when $|P| \ll |U|$, which is our case.

Although random forest does not typically produce a probabilistic classifier, we use the mean class probabilities of each tree to estimate $p(s = l \mid x)$. We then use the PosOnly method described in Section 4.2 to adapt the probabilistic classifier to PU learning.

5 Experiments

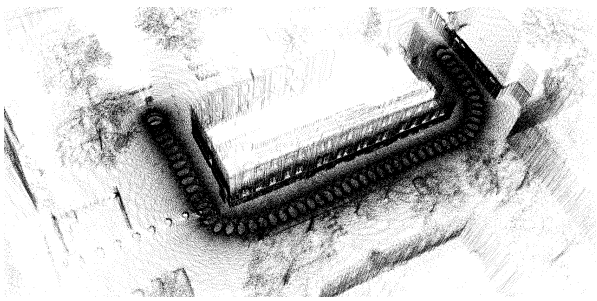
We tested our approach on the state-of-the-art FPFH descriptors. Their relatively low dimensionality[3] (33), compared to other descriptors such as SHOT (352), made them an ideal candidate to our approach. Moreover, FPFH descriptors computation relies on a single intuitive parameter (radius of the neighborhood), making the results less sensitive to tuning.

5.1 Testing framework

Our proposed method was tested on two sequences of 3D laser scans extracted from publicly available datasets. The first sequence was the scans 22 to 80 from the *Hannover* dataset (Fig. 4a), featuring trees and bushes as well as buildings (ground truth was not available for scans 1-21). The robot trajectory for these scans covered a total distance of about 84 m and contained $13,557$ points on average. The second sequence was the complete *Wood Summer* dataset (Fig. 4b), from the Autonomous Systems Lab [Pomerleau *et al.*, 2012], and consisted in 37 scans featuring dense vegetation in an unstructured environment, with a robot trajectory covering a distance of about 21 m and an average of about $182,000$ points per scan.

For each of these sequences, the classifier was trained only once, using the first two scans, according to the method described in Section 4. This trained classifier was then used to filter the descriptors for the rest of the sequence, without further training. This was done to verify that a learned model at a certain location in the environment could still give satisfactory results at a different location, provided that the type of environment had not changed significantly. The values of d_{err} and ϕ_{err} used were 0.5 m and 5° for *Hannover*, and 0.2 m and 2° for *Wood Summer*. The SAC-IA algorithm was used to

[3] These are the default implementation values in the Point Cloud Library (PCL) [Rusu and Cousins, 2011]. The dimensionality can change according to some parameters.



a) *Hannover* scans 22 to 80.

b) *Wood Summer* (scan 0 only)

Figure 4: Overview of the sequences used for testing.

perform 100 registration attempts between each pair of consecutive scans, to take into account its probabilistic nature. The number of iterations within each registration was fixed to 100.

Recall that the SAC-IA algorithm performs a coarse registration, which is a preprocessing step to a local registration algorithm such as ICP. Therefore, smaller errors and error variances indicate an improvement in convergence and robustness. This directly improves the probability of falling within the basin of convergence of ICP's global minimum.

We compared the following three approaches:

- using all descriptors;
- using reliable descriptors, classified using our method;
- using a random subset of size N_{rand} of all descriptors.

The third approach, a valid way to speed up the scan alignment, was included to demonstrate that our method did indeed increase the proportion of reliable descriptors in the remaining data. The value of N_{rand} was set to the average number of predicted reliable descriptors by our classifier for the whole series of scans. This choice insured a fair comparison with our method, as they had similar computation times.

To evaluate and compare the different approaches, we used the accuracy of scan registrations, in terms of rotation and translation error norms. We also compared the computation times for all the approaches, to estimate the speedup.

5.2 Results

Fig. 5(a) and Fig. 5(c) show the boxplot distributions of the translation and rotation error norms respectively, for the *Hannover* test sequence. For the subsampling method, we used $N_{rand} = 4,500$. Our filtering method significantly improved the quality of the alignments, both in rotation and translation, for the scans immediately after training (number 22-27). This improvement is still noticeable, albeit smaller, for scans 28-79. The histogram distributions of all the errors for scans 22-79 are shown in Fig. 5(b)) and Fig. 5(d)). These distributions show a reduction in the median error (dashed vertical lines) with our approach over the complete sequences, compared to using all or a randomly picked subset of descriptors. Finally, the number of large errors (over 2 m in translation or 10^o in rotation) was significantly reduced, as indicated by the location of the 95th percentile (solid vertical lines). This reduction can be key to improving robustness, as it increases the probability of a local alignment algorithm (e.g. ICP) to converge towards the true solution.

Fig. 6(a)) and Fig. 6(c)) show the same curves for the *Wood Summer* dataset. In this case, $N_{rand} = 25,000$ samples were used in the subsampling method. Here, the improvement in translational error is even more significant, with a reduction of about 40% in the median error (Fig. 6(b)). Again, the number of large errors (over 0.6 m in translation or 10^o in rotation) was significantly reduced. This increased robustness and accuracy indicate that our approach can offer significant improvements for challenging forested environments.

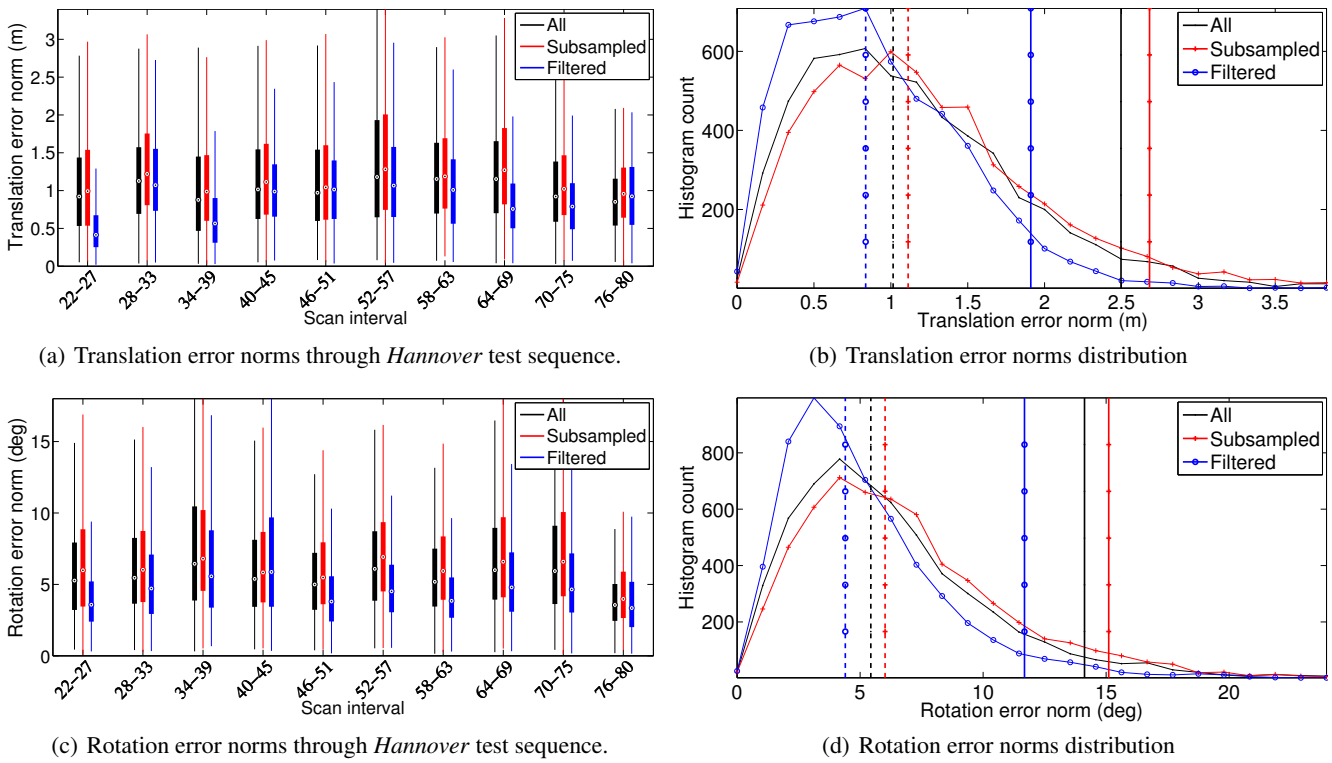
Table 1: Breakdown of average computation times (in seconds) for our approach and for scan alignments using all descriptors.

Dataset	**All**	**Our Approach**			
	Align	Gen.	Train	Filter	Align
Hannover	17.4	0.8	4.2	0.2	**2.7**
Wood S.	311	217	29.5	2.3	**15.3**

The other benefit of our approach is the significant speedup of the alignment process. Table 1 details the average time taken to generate the training dataset, train the classifier, filter the descriptors and align the scans. We can see that, once the classifier was trained, the filtering and alignment resulted in a speedup by a factor of $17.4/(0.2+2.7) = 6$ for *Hannover* and $311/(2.3+15.3) = 17.7$ for *Wood Summer*. If we take into account the training overhead (Gen.+Train), our method was still faster than using all the descriptors, and this after only a single alignment. Indeed, this overhead is less than the time needed to perform a single alignment with all the descriptors. For Hannover: $(0.8+4.2+0.2+2.7) = 7.9\ s < 17.4\ s$ and for Wood Summer: $(217+29.5+2.3+15.3) = 264.1\ s < 311\ s$. Keeping in mind that the dataset generation and training steps are executed only once, this overhead can be amortized over the subsequent scans, as long as the environment stays similar.

5.3 Implementation details

All the computations were performed sequentially, although all the steps of our proposed method can be parallelized. The computations were performed using the Point Cloud Library (PCL) [Rusu and Cousins, 2011] and the Python Scikit-learn library [Pedregosa *et al.*, 2011].



(a) Translation error norms through *Hannover* test sequence.

(b) Translation error norms distribution

(c) Rotation error norms through *Hannover* test sequence.

(d) Rotation error norms distribution

Figure 5: Alignment results for the *Hannover* test sequence. Subfigures (a) and (c) show the boxplot distributions of the translation and rotation error norms throughout the test sequence. Results were bundled in groups of six scans to simplify the plots. Subfigures (b) and (d) show the histogram distributions of all the errors over the complete sequence, in translation and rotation respectively. The vertical dashed lines indicate the median values of the distributions, whereas the vertical solid lines indicate the 95th percentiles.

6 Future Works

In our experiments, we used sequences of contiguous scans taken from the same environment. The results that were obtained are encouraging, although they are based on the assumption that the environment in which the robot operates remains similar. An interesting extension to our work would be to allow operation through changing environments. As previously mentionned, the training process can be repeated as needed, and its computation overhead is relatively small. Therefore, given a detection mechanism for a changing environment, one could trigger the retraining process, thus allowing to adapt the predictor to the new environment. Such a mechanism could consist in using a supervised anomaly detection algorithm [Chandola *et al.*, 2009]. Then, given a threshold on the number of anomalies detected, one could trigger retraining. A simpler and straightforward method could be to unconditionally retrain the predictor at given scan intervals.

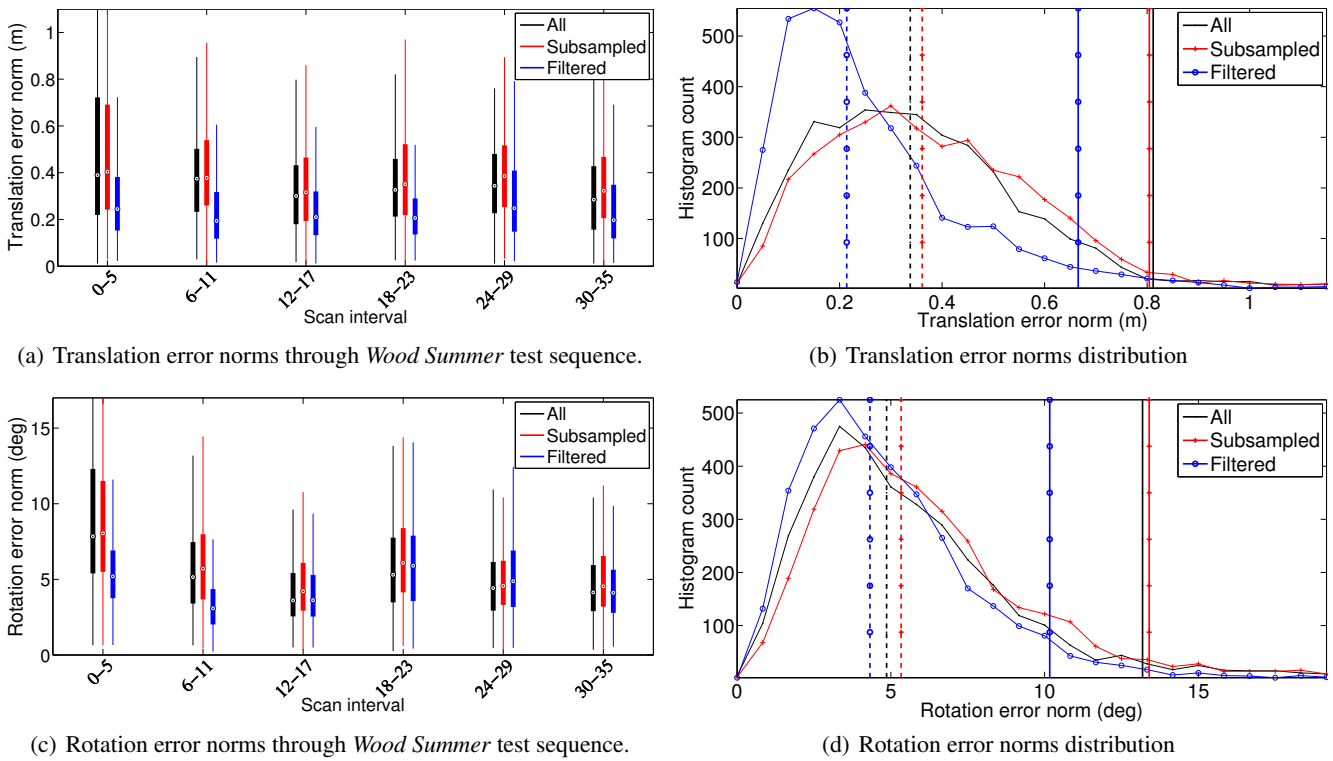
7 Conclusion

In this paper, we have addressed the issues related to the instability of descriptors in natural environments featuring trees and vegetation. We demonstrated that these unstable descriptors are unreliable for point matching and scan registration. To solve this problem, we have shown that *positive and unlabeled* learning is a well suited machine learning paradigm. We have adapted the random forest algorithm, modified in its bootstrap sampling phase to balance classes, to this paradigm. This approach gave rise to a fast and more robust learning algorithm that allows to filter out unreliable descriptors. A key contribution of our paper is having shown that classifiers can be trained using data collected by the robot itself, without any human intervention. The subset of filtered descriptors provided two benefits to the SAC-IA alignment algorithm: a better registration accuracy and robustness, and faster processing time. Our approach has been tested on two real datasets, one of them featuring dense vegetation in an unstructured environment, with encouraging results.

References

[Angelova *et al.*, 2006] A. Angelova, L. Matthies, D. Helmick, and P. Perona. Slip prediction using visual information. In *Proc. of Robotics: Science and Systems*, August 2006.

[Besl and McKay, 1992] P. J. Besl and H. D. McKay. A method for registration of 3-d shapes. *Pattern Analysis and Machine Intelligence, IEEE Trans. on*, 14(2):239 –256, Feb 1992.



(a) Translation error norms through *Wood Summer* test sequence.

(b) Translation error norms distribution

(c) Rotation error norms through *Wood Summer* test sequence.

(d) Rotation error norms distribution

Figure 6: Alignment results for the *Wood Summer* test sequence. See the legend of Fig. 5 for details.

[Breiman, 1996] Leo Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

[Breiman, 2001] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[Brown *et al.*, 2011] M. Brown, Hua G., and S. Winder. Discriminative learning of local image descriptors. *Pattern Analysis and Machine Intelligence, IEEE Trans. on*, 33(1):43 –57, jan. 2011.

[Chandola *et al.*, 2009] V. Chandola, a. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15, 2009.

[Dahlkamp *et al.*, 2006] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski. Self-supervised monocular road detection in desert terrain. In *Proc. of Robotics: Science and Systems*, Aug. 2006.

[Elkan and Noto, 2008] C. Elkan and K. Noto. Learning classifiers from only positive and unlabeled data. *Proc. of the 14th ACM SIGKDD int'l conf. on Knowledge discovery and data mining - KDD 08*, page 213, 2008.

[Fischler and Bolles, 1981] M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[Grabner *et al.*, 2007] M. Grabner, H. Grabner, and H. Bischof. Learning Features for Tracking. In *IEEE Computer Vision and Pattern Recognition or CVPR*, 2007.

[Krebs *et al.*, 2010] A. Krebs, C. Pradalier, and R. Siegwart. Adaptive rover behavior based on online empirical evaluation: Rover-terrain interaction and near-to-far learning. *J. of Field Robotics*, 27(2):158–180, 2010.

[Lowe, 2004] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int'l J. of Computer Vision*, 60:91–110, 2004.

[Mordelet and Vert, 2010] F. Mordelet and J.P. Vert. A bagging svm to learn from positive and unlabeled examples. *arXiv preprint arXiv:1010.0772*, 2010.

[Mordelet and Vert, 2013] F. Mordelet and J.-P. Vert. Supervised inference of gene regulatory networks from positive and unlabeled examples. In *Data Mining for Systems Biology*, volume 939 of *Methods in Molecular Biology*, pages 47–58. Humana Press, 2013.

[Pedregosa *et al.*, 2011] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *J. of Machine Learning Research*, 12:2825–2830, 2011.

[Pomerleau *et al.*, 2012] F. Pomerleau, M. Liu, F. Colas, and R. Siegwart. Challenging data sets for point cloud registration algorithms. *Int'l J. of Robotics Research*, 31(14):1705 – 1711, December 2012.

[Rusinkiewicz and Levoy, 2001] S. Rusinkiewicz and M. Levoy. Efficient Variants of the ICP Algorithm. In

Int'l Conf. on 3-D Imaging and Modeling, pages 145–152, 2001.

[Rusu and Cousins, 2011] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE Int'l Conf. on Robotics and Automation (ICRA)*, 2011.

[Rusu *et al.*, 2009] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA '09. IEEE Int'l Conf. on*, pages 3212 –3217, may 2009.

[Schölkopf *et al.*, 2001] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[Scovanner *et al.*, 2007] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proc. of the 15th int'l conf. on Multimedia*, pages 357–360. ACM, 2007.

[Sim and Dudek, 1999] R. Sim and G. Dudek. Learning and evaluating visual features for pose estimation. In *In Proc. of the 7th Int'l Conf. on Computer Vision (ICCV99), Kerkyra*, pages 1217–1222. IEEE Press, 1999.

[Song *et al.*, 2012] M. Song, F. Sun, and K. Iagnemma. Natural feature based localization in forested environments. In *IROS*, pages 3384–3390. IEEE, 2012.

[Steder *et al.*, 2011] B. Steder, R.B. Rusu, K. Konolige, and W. Burgard. Point feature extraction on 3d range scans taking into account object boundaries. In *Robotics and Automation (ICRA), 2011 IEEE Int'l Conf. on*, pages 2601 –2608, may 2011.

[Tangelder and Veltkamp, 2008] J. W. Tangelder and R. C. Veltkamp. A survey of content based 3d shape retrieval methods. *Multimedia Tools Appl.*, 39(3):441–471, September 2008.

[Tombari *et al.*, 2010] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In *Proc. of the 11th European Conf. on Computer Vision: Part III*, pages 356–369, 2010.

[Trzcinski *et al.*, 2012] T. Trzcinski, M. Christoudias, V. Lepetit, and P. Fua. Learning image descriptors with the boosting-trick. In *Advances in Neural Information Processing Systems 25*, pages 278–286. 2012.

[Winder *et al.*, 2009] Simon A. J. Winder, G. Hua, and M. Brown. Picking the best daisy. In *2009 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 178–185. IEEE, 2009.

[Wulf *et al.*, 2008] O. Wulf, A. Nüchter, J. Hertzberg, and B. Wagner. Benchmarking urban six-degree-of-freedom simultaneous localization and mapping. *J. of Field Robotics*, 25(3):148–163, 2008.