2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Enhancing 6D Visual Relocalisation with Depth Cameras

José Martínez-Carranza, Andrew Calway and Walterio Mayol-Cuevas
Department of Computer Science, University of Bristol, UK
Bristol Robotics Laboratory, Bristol, UK

***Abstract*— Relocalisation in 6D is relevant to a variety of Robotics applications and in particular to agile cameras exploring a 3D environment. While the use of geometry has commonly helped to validate appearance as a back-end process in several relocalisation systems before, we are interested in using 3D information to assist fast pose relocalisation computation as part of a front-end task. Our approach rapidly searches for a reduced number of visual descriptors, previously observed and stored in a database, that can be used to effectively compute the camera pose corresponding to the current view. We guide the search by means of constructing validated candidate sets using a 3D test involving the depth information obtained with an RGB-D camera (e.g. stereo of with structured light). Our experiments demonstrate that this process returns a compact quality set that works better for the pose estimation stage than when using a typical Nearest-Neighbor search over appearance only. The improvements are observed in terms of percentage of relocalised frames and speed, where the latter goes up to two orders of magnitude w.r.t. the conventional search.**

I. The Importance of Relocalisation

The ability to know the relative pose of a platform w.r.t. a previously observed scene is an essential competence for several robotic tasks. Known as *relocalisation*, this ability is commonly used to perform loop closure in mapping to manage drift, but it also has the potential to be applicable for scene-guided object search or location-based context and activity recognition.

Nowadays, maps can be obtained from a variety of sensors and structure recovery methods, for example, by using sonar or laser [1], [2], by using single cameras without additional odometry [3], and more recently, depth cameras using stereo [5] or structured light [6], [4]. From the point of view of the structure recovery and concentrating in those systems using cameras, maps can be built in a metric or topological fashion with variations of the Kalman filter [3], [7], Particle filters [8], Visual Odometry [9] or simply treated as a collection of images [10], [12].

In this varied landscape and depending on the application, relocalisation can be addressed as an image-retrieval task [31] which returns the most similar image to a query image. The latter corresponds to the image of the camera pose to be relocalised. The query image is sought in an indexed database of images collected during exploration (training) time. If a camera position is attached to every image in the database then a rough relocalised position could also be infered. On the other hand and from a geometric point of view, relocalisation can also be seen as the problem of finding a set of visual point descriptors in a database that matches a set of descriptors extracted from salient points in the query image. If each matched descriptor is linked to a 3D scene point then the three-point pose algorithm plus RANSAC is frequently used to recover the camera's 6D pose. Note that a method such as RANSAC is required to deal with outliers derived from false positive matches. This geometric-driven approach, which offers high quality pose estimates, is the one of interest in this paper.

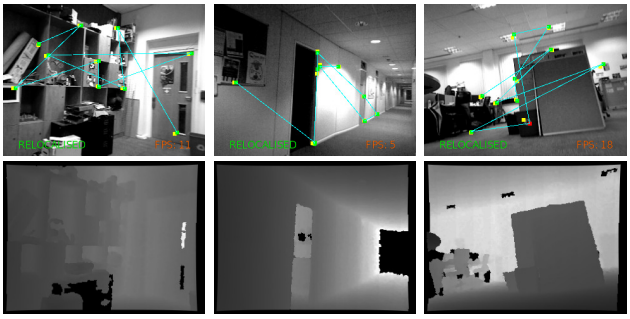


Fig. 1. Examples of quality feature sets obtained by our algorithm for various scenarios, including for a fast moving quadrotor platform.

Having a bare list of descriptors stored in a database is impractical for a search that employs a Nearest-Neigbour (NN) comparison procedure to find the best set of matches. This is addressed in, for example, bag-of-words approaches where similar descriptors can be grouped together in the same bin ('visual word') thus we only need to compare against those descriptors in the relevant bins. However, depending on how descriptors are grouped, bins may still contain a large number of descriptors, hence the NN search may be expensive and also prone to mismatches [28].

In this paper we propose a fast and relatively simple approach using a geometric 3D test that helps to reduce the number of descriptor comparisons needed within bins and independently of the procedure chosen to create such bins. Our goal is not only to reduce the number of unnecessary comparisons, but also to identify a set of quality matched descriptors whose 3D positions can be used effectively within RANSAC in order to recover 6D pose.

Our approach monotonically constructs chains of feature matches starting by finding an anchor point using visual appearance and informed by Inverse Document Frequency (IDF) analysis, but in a novel way by combining it with the geometrical information available between the stored descriptors. This assessment qualifies new elements of the chain of descriptors if the 3D distance w.r.t. the anchor point is within a tolerance margin when compared with the data coming from an RGB-D (or stereo) camera.

978-1-4673-6357-0/13/$31.00 ©2013 IEEE

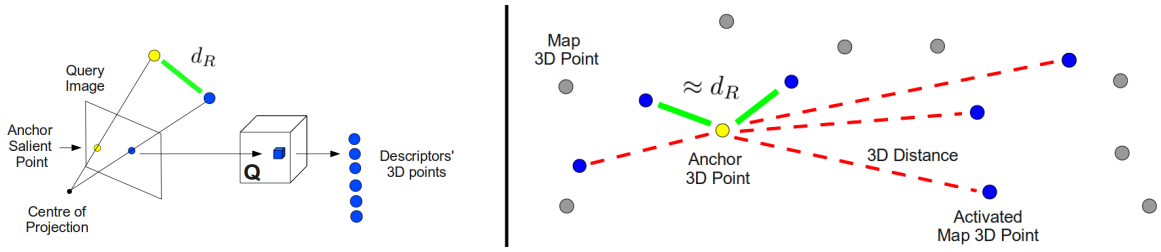


Fig. 2. Schematic example illustrating our approach: (a) 3D distance d_R in between salient points can be calculated using the depth image, this distance becomes a reference distance utilized to prune descriptors; (b) Top view of 3D Points in the map, blue points correspond to those associated to the descriptors retrieved from the hash table $\mathbf{Q}$. A retrieved descriptor is ruled out for comparison if the distance in between its associated 3D point and the 3D anchor point (in yellow) is dissimilar to d_R (note the red dash lines).

Figure 2 illustrates our iterative method where a first anchor point has been found. This anchor point is related to the 3D position of a descriptor in the database that has been matched with the descriptor of a salient point in the query image. Note that the 3D position of this salient point w.r.t. the camera's optical centre can be calculated from the depth image. This can also be done for a second salient point and thus, the 3D distance in between these two salient points can be used as a reference distance d_R. Therefore, those descriptors in the bin associated to the second salient point, whose distance is outside a tolerance margin of d_R are pruned. The surviving descriptors are used for comparison. This process can be repeated for further salient points within a bin (or visual word) of descriptors associated to it. A small chain of features is constructed in this way and then passed for further RANSAC validation.

Experiments show that, by implementing our approach, the number of descriptor comparisons is significantly reduced. Moreover, the resulting set of matched descriptors is of better quality since we observe how a smaller number of matched descriptors is still enough to produce successful relocalisations.

The rest of the paper has been organised as follows: section II describes related work; section III describes the framework used to build our approach; section IV presents a full detailed version of our algorithm while section V describes our experiments in the context of frame to frame relocalisation. Final remarks are discussed in section VI.

II. Related Work

First approaches addressing the relocalisation problem such as that of Williams et.al. [19], [16] suggests to look at it as a classification problem where visual descriptors, in their case simple image patches, are recognized using randomized trees trained on-line. However, the main drawbacks in this approach is the large memory footprint which does not scale well with the number of visual descriptors.

Klein et.al. [29] notices that relocalisation can be done very simply in the context of a key-frame base SLAM. He suggests to use a sub-sampled version of the query image (40×30 pixels) and compare it against all the sub-sampled versions of stored key frames in the map. Those 3D points related to the matched key frame are activated together with camera tracking using the key frame's camera pose as initial pose to be refined by the tracker. The method is effective when the camera is close to any key-frame pose, however, it would be expected to fail otherwise.

Eade and Drummond [15] borrowed the concept of visual bag-of-words widely used in the image-retrieval field [31], and that has been successfully used for non-metric loop closure [30]. They use this concept within a multi-node graph SLAM approach to deal with loop closure and relocalisation within the same framework. Nodes are ranked using the Term Frequency-Inverse Document Frequency (TF-IDF) of the visual words in each node in order to retrieve those k most likely nodes. A refined relocalised pose is obtained by matching each set of descriptors within the node against extracted descriptors in the query image.

One of the critical parts when using a bag-of-words approach is the construction of the visual vocabulary which intends to quantise similar descriptors into the same bin. For instance, [15] uses a conventional k-means algorithm where over 3000 clusters are found among mapped descriptors. It has been observed that the retrieval task benefits from having a large visual vocabulary [32]. However, producing a large vocabulary with k-means is not only slow, but any update attempting to increase the vocabulary can be expensive. In order to cope with the latter, some works [32], [28] propose to use a hierarchical k-means which provides a more efficient training while providing a large vocabulary in the form of a tree with k branches. Each node of the tree is seen as a word within the so called 'visual vocabulary tree'.

In contrast to the above, Chekhlov et.al. [14] propose a relocalisation method that uses a hashing technique to quantise descriptors instead of using any k-means algorithm. To achieve this, any mapped visual feature contains two types of descriptors: a coarse descriptor which consists of three Haar coefficients [25], and a Histogram of Gradients (HoG) descriptor [7]. The coarse descriptor is soft quantised in order to provide an index within a 3D hash table where each bin stores all those HoG descriptors with same Haar descriptor. A hash table is good for non-stop, real-time updating of the "vocabulary" as each one of these bins corresponds to a visual word in the visual vocabulary sense. In the above cases either using visual words or hash tables and depending on the quantisation parameters, every word may have many thousands of descriptors associated to them.

While our method is equally applicable to the other quantisation strategies such as bags of words or trees, in this paper we are interested in the relocalisation method proposed in [7] for two main reasons: (1) clustering or indexing of features can be carried out on-line at very low cost and alongside mapping; (2) updating the number of visual words only involves updating the indexes obtained from quantising the Haar descriptor, hence, it can be done in linear time in the total number of HoG descriptors stored in the hash table. Furthermore, we also chose this approach to address its *inconvenience*, which is that bins in the table may contain a large number of HoG descriptors since the separation is not performed in the HoG descriptor space, but at the coarse descriptor space, which is highly redundant.

Experiments in further sections demonstrate that when the bins contain descriptors in the order of thousands, Chekhlov's relocalisation method is just too slow. However, this gives us the opportunity to demonstrate that our proposed 3D test can rapidly and effectively prune descriptors in the activated bins, thus not only avoiding comparisons, which allows important time savings, but also producing a pool of high quality descriptors whose validated 3D geometry allows successful relocalisation, even when only few matched descriptors are passed to the pose estimation consensus stage.

Methods similar in aims to ours include that of Cadena et.al. [23] where a bag of visual words is combined with a Conditional Random Filed (CRF) to validate geometric matching in between the retrieved images and the query image. This concept resembles our goal of seeking to verify 3D geometry consistency. However, that method is generally slow due to the overhead induced by the CRFs, and authors report performances of 1 fps. Recently, Gee and Mayol [33] reported a system that builds a 3D model of the scene using RGB-D imagery and when relocalisation is needed, the system builds synthetic sub-sampled views around the camera trajectory. This samples are used in a general regression method to recover the camera pose of the query image. However it is only demonstrated for small workspaces.

III. Appearance-driven Relocalisation

We first briefly describe the relocalisation method we use here as baseline to build and showcase our proposed 3D-enhanced method. We thus refer to this baseline approach as *appearance-driven*. The system is described in more detail in [14]. It is based on visual features consisting of both: coarse descriptors, to access a hash quantization table, and finer descriptors using Histograms of Gradients (HoG) organized on a scale-stack to offer scale invariance[1]. These features are assumed to have been mapped during exploration time. As discussed before, the map could have been created in a number of ways and in fact, we have tested with maps created with standard SLAM or simple visual odometry.

During mapping, the hash table $\mathbf{Q}$ is filled out with every HoG descriptor plus its 3D position in the world. These descriptors are indexed into the table by using the coarse descriptor $\mathbf{h}$ based on Haar coefficients [25] extracted from the same image patches used to calculate the HoG descriptors. For the latter, an image patch is split into quadrants (xx, xy, yx, yy), although Haar coefficients are calculated for xx, yy and xy only. These three coefficients describe the rough appearance of the patch in those quadrants. See [14] for full details. The three coefficients are quantised to generate three indexes (i, j, k) which give access to a bin in $\mathbf{Q}$. Those HoG descriptors with same coarse descriptor $\mathbf{h} = (i, j, k)$ are allocated into the same bin. This means that $\mathbf{Q}(\mathbf{h}) = \mathbf{Q}(i, j, k) = \{\mathbf{d}_1, \mathbf{d}_2, \dots\}$ contains a list of descriptors whose associated coarse descriptor[2] is also $\mathbf{h}$. To avoid confusion in the notation, we assume that the descriptor $\mathbf{d}_i$ has a link to its mapped 3D position in the world, which often we will refer to as $\mathbf{p}_i$.

At relocalisation time, salient points $\mathbf{s}_i$ are detected in the query image using the FAST corner detector [34]. Then image patches of size 11×11, centred at $\mathbf{s}_i$, are extracted. For each patch, coarse Haar coefficients $\mathbf{h}_i$ are computed. Each $\mathbf{h}_i$ takes us to a bin $\mathbf{Q}(\mathbf{h}_i)$ containing HoG descriptors. if $|\mathbf{Q}(\mathbf{h}_i)| > 0$, a HoG descriptor $\mathbf{d}_{qi}$ is extracted from the patch centred at $\mathbf{s}_i$ in the query image. Let $\mathbf{d}_{iB} \in \mathbf{Q}(\mathbf{h}_i)$ be the best descriptor matching $\mathbf{d}_{qi}$, which is obtained using a simple 1-NN procedure. If a match is found then we pair $\mathbf{s}_i$ with the respective 3D position of $\mathbf{d}_{iB}$, which we call $\mathbf{p}_{iB}$ and add it to a list of 2D-3D pairs $\mathbf{L} = \mathbf{L} \cup \{(\mathbf{s}_i, \mathbf{p}_{iB})\}$. Once all the possible matches have been found, $\mathbf{L}$ is passed to a three-point pose algorithm plus RANSAC in order to perform a consensus and find the best camera pose that minimizes image distances in between camera projections of descriptors' 3D positions and their associated image coordinates. The pose with the biggest number of inliers is return as the relocalised pose.

IV. Quality Matches Using 3D Data

In this work we are interested in exploiting the 3D information that a depth camera provides in order to prune descriptors to compare against within a relocalisation procedure. Seeking to match a descriptor against a large list of descriptors using 1-NN is not efficient and it may become prohibitive. We could organise these descriptors using a hierarchical tree for a more efficient search [32] but we will still have to deal with false positives in appearance. We empirically demonstrate that descriptors based on appearance can be rapidly pruned by means of validating their 3D geometric consistency even if they are not organized in any other structure beyond a simple hash table.

We specifically show that matches returned once 3D validation has been adopted, are quality matches less likely to be false positives.

We assume that a hash table $\mathbf{Q}$ has been filled out as explained in section III. Thus, having a query image, with RGB and Depth data, where relocalisation is to be attempted,

[1]We use 20 scales from 0.4 to 2.4 times the size of the original patch.

[2]Note that each scaled patch produces a different HoG descriptor but also a different coarse descriptor and thus they may be allocated to different bins in $\mathbf{Q}$.

our algorithm aims at constructing a quality chain (list) $\mathbf{C}$ of 2D-3D pairs in the following manner and within a time limit parameter:

1) For a salient point $\mathbf{s}_a$ on the RGB image compute its coarse descriptor $\mathbf{h}_a$ (see section III) to access the corresponding bin in the quantization table. If $|\mathbf{Q}(\mathbf{h}_a)| > 0$ then compute the descriptor $\mathbf{d}_{qa}$ on the query image centred at $\mathbf{s}_a$ and find its best match in $\mathbf{Q}(\mathbf{h}_a)$. If found, let $\mathbf{d}_{aB} \in \mathbf{Q}(\mathbf{h}_a)$ be the *B*est match such that we initialise the chain as $\mathbf{C} = \{(\mathbf{s}_a, \mathbf{p}_{aB})\}$ (anchor point), where $\mathbf{p}_{aB}$ is the 3D point associated to $\mathbf{d}_{aB}$. Otherwise, choose another salient point and repeat 1.
2) Choose another salient point $\mathbf{s}_b$ on the RGB image, calculate its coarse descriptor $\mathbf{h}_b$ and get to $\mathbf{Q}(\mathbf{h}_b)$. If $|\mathbf{Q}(\mathbf{h}_b)| = 0$ then repeat step 2.
3) Calculate the reference 3D distance d_R in between the salient points $\mathbf{s}_a$ and $\mathbf{s}_b$ (see figure 2a) with the help of the query depth image.
4) Find which pairwise 3D distances between $\mathbf{p}_{aB}$ and descriptors in $\mathbf{Q}(\mathbf{h}_b)$ are bigger than a threshold ϵ_D i.e: let $\mathbf{p}_{bi}$ the associated 3D point of the descriptor $\mathbf{d}_{bi} \in \mathbf{Q}(\mathbf{h}_b)$, if $|dist(\mathbf{p}_{aB}, \mathbf{p}_b) - d_R| > \epsilon_D$ then mark $\mathbf{d}_{bi}$ as an 'invalid' descriptor[3]. Otherwise, the descriptor is marked as valid. If all the descriptors $\mathbf{d}_{bi} \in \mathbf{Q}(\mathbf{h}_b)$ have been marked as invalid then go to step 2.
5) From the RGB query image calculate the HoG descriptor $\mathbf{d}_{qb}$ centred at $\mathbf{s}_b$. Then compare $\mathbf{d}_{qb}$ to each 'valid' descriptor $\mathbf{d}_{bi} \in \mathbf{Q}(h_b)$. If no match is found then go to step 2. Otherwise, let $\mathbf{d}_{bB}$ be the 'best match' whose 3D position $\mathbf{p}_{bB}$ is paired and added to the quality chain: $\mathbf{C} = \mathbf{C} \cup \{(\mathbf{s}_b, \mathbf{p}_{bB})\}$
6) If $|\mathbf{C}| \geq c_{size}$ then return $\mathbf{C}$ and exit, otherwise go to step 2.

Note that the *key* step in the above procedure is the pruning of 'invalid' descriptors without having to perform any descriptor comparison at all but only the 3D distance check. This avoids unnecessary descriptor comparisons, saves computational effort and therefore time. The former is crucial especially when descriptor comparison involves comparisons of 128D vectors using the $\mathbf{L}_1$ or $\mathbf{L}_2$ norms.

If $\mathbf{C}$ is not empty then the above procedure will have monotonically constructed a chain of quality 2D-3D pairs. To retrieve the full 6D pose, we carry out a consensus procedure using the three-point pose algorithm and RANSAC over $\mathbf{C}$. Note that even when we deem $\mathbf{C}$ to contain quality matches, still there is a possibility that a mismatch occurs. Therefore the consensus helps to minimize the impact of those mismatches while estimating the camera pose with the most number of inliers.

From the above, our hypothesis, which is confirmed later by our experiments, is that the chain $\mathbf{C}$ does not have to be large (only about 10-15 elements) for robust 6D pose estimation. The procedure directly constructs feature chains that are much smaller than those in similarly inspired stereo-assisted localisation methods e.g. [22], [23] which further require training and minimization stages. Our approach aims for a bottom-up validation of quality features for a minimal set of features for pose computation.

[3]Here $dist(\mathbf{x}, \mathbf{y})$ stands for 3D euclidian distance in between 3D points $\mathbf{x}$ and $\mathbf{y}$.

A. Improvements to the algorithm

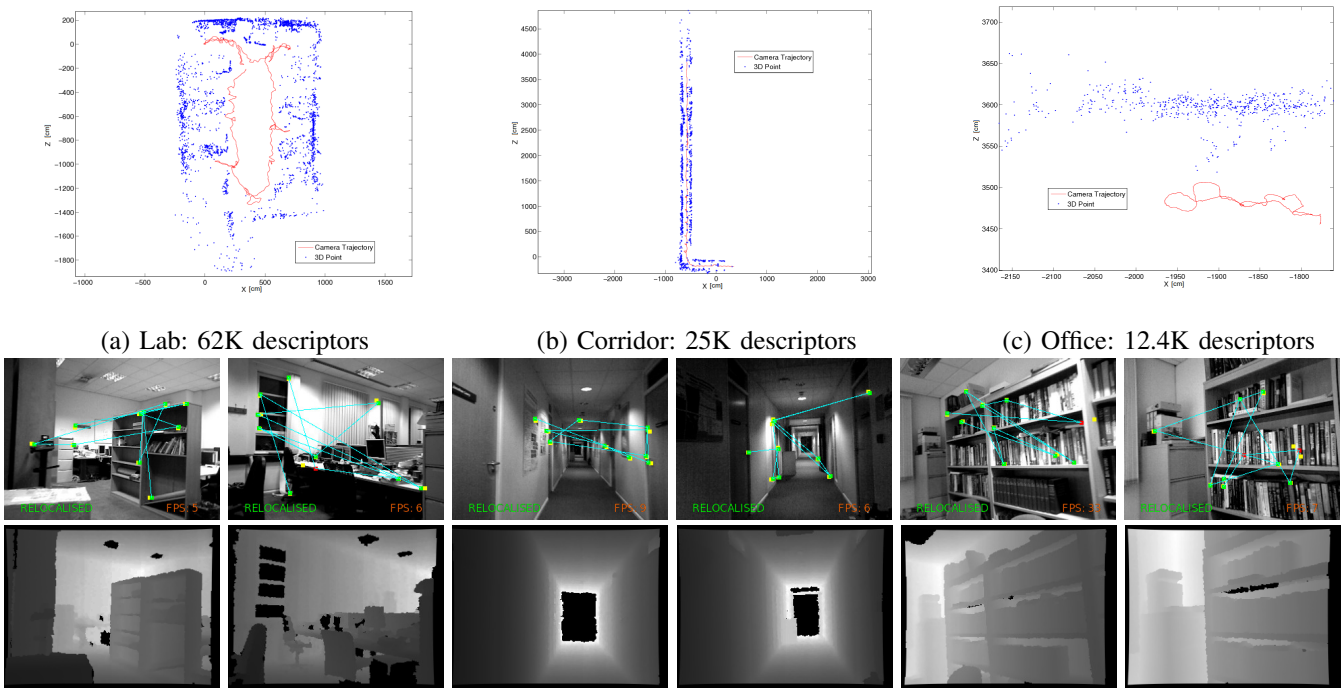
In a similar way to visual-words-based methods, to speed up even further the parsing of salient points $\mathbf{s}_i$ in the RGB query image, these are ranked using their IDF, which is obtained simply by $N/|\mathbf{Q}(\mathbf{h}_i)|$, where $N = |\mathbf{Q}|$ is the total number of HoG descriptors in the hash table (total of words in the document) and $|\mathbf{Q}(\mathbf{h}_i)|$ is the frequency of occurrence of the word $\mathbf{Q}(\mathbf{h}_i)$.

Also, step 3 of the above procedure can be extended to include more than one 3D reference distance instead of using only that obtained with the anchor point $(\mathbf{s}_a, \mathbf{p}_{aB}) \in \mathbf{C}$. For instance, assuming we use all members of $\mathbf{C}$, a list of reference distances is produced: $\mathbf{D} = \{d_{R1}, d_{R2}, \ldots, d_{R|C|}\}$, where d_{Ri} is the 3D distance in between the candidate salient point $\mathbf{s}_b$ and the image point $\mathbf{s}_i$ in $\mathbf{C}$ (remember that such distance is obtained by using the query depth image). Then for step 4, the 3D distance in between the 3D position of a candidate descriptor in $\mathbf{Q}(\mathbf{h}_b)$ and each 3D point $\mathbf{p}_{iB}$ in $\mathbf{C}$ should be similar, within the tolerance margin ϵ_D, than the corresponding distance d_{Ri}. If at least one of this distances breaches the margin ϵ_D then $\mathbf{d}_b$ is marked as an invalid descriptor.

Finally, our algorithm heavily depends on the anchor point being a true positive match, which references to the right part of the map at which the camera is currently pointing at. Our experience indicates that if the anchor point is a false positive then the chain will fail to be constructed since the map 3D geometry will not correspond to the geometry observed by the query depth image. Note that once the chaining process has started (after step 2), there is no provision for when we have run out of salient points in the query image and the chain size is less than c_{size}. The solution is simple and for this we suggest to rank the salient points as indicated before. Thus, if we have parsed all the salient points then we should go back to step 1, but we should start from the next position in the ranking where we found the previous unsuccessful anchor point. Our experiments show that a successful chain is consructed in 80% of the cases using the first found anchor point, the rest takes about three or four attempts before a successful chain is constructed.

V. Experiments

We first perform experiments for the tuning of the parameters in our algorithm before comparing the performance of our approach with the baseline appearance-driven method in terms of speed and percentage of relocalised frames. In both cases we will show the effect of using a quality set of matches produced by our 3D geometric test versus using a set obtained with a blind 1-NN search. All the experiments



(a) Lab: 62K descriptors (b) Corridor: 25K descriptors (c) Office: 12.4K descriptors

Fig. 3. Three different 3D scenarios: Lab, Corridor and Office used to construct a map treated as a topological collection of submaps adding to 100K visual features and their HoG descriptors. Image shows sample RGB and depth images with their chain $\mathbf{C}$ of quality matches plotted. The position of the HoG descriptor $\mathbf{p}_{iB}$ is shown in yellow, its associated image point is shown in green $\mathbf{s}_{iB}$ if the distance is less than the set threshold (inlier), otherwise it is red (outlier). Turquoise lines show the chain obtained. Best seen in the PDF version.

ran in a single core with clock of 2.40 GHz and without any GPU computation.

For the experiments, a map of 3D points was built using a visual odometry system similar to that in [27] but with more robust feature description as per section III. Features were initialized with an inverse depth parametrization [18] but primed by the depth obtained from an RGB-D sensor (ASUS X-tion pro live). We use 320×240 pixels intensity and depth images. For map building, the depth information from the sensor quickly leads to convergence in 3D and after a converged feature goes out of view, its corresponding HoG descriptors are quantised and stored in the hash table $\mathbf{Q}$ as explained in section III. This allows agile exploration of the environment without the burden of state growth as in e.g. full SLAM. Our saliency detection returns typically 50 to 80 features distributed across the frame.

Our testing ground is a topological collection of sub maps (see figure 3) of different size and visual ambiguity: a $200m^2$ laboratory, a $40m$ long L-shaped little textured corridor and a shelved wall in an office space. To make the environment more realistic for larger scale relocalisation all these maps are shared by the same hash table $\mathbf{Q}$ containing 100,000 HoG descriptors. We should highlight that several bins in $\mathbf{Q}$ may contain thousands of descriptors. We further use another lab environment for tests with a quadrotor.

A. Parameter Tuning and Relocalisation Criteria

We are interested in assessing the effect of parameters in the quality of the features in the chain which ultimately will be used by RANSAC to find a camera pose. Therefore we should establish whether such features are true inliers or not. For this experiments the algorithm was set to attempt to use only one anchor point for the construction of the chain $\mathbf{C}$. If the chain size is not bigger than c_{size} then the algorithm will exit without attempting to find another anchor point.

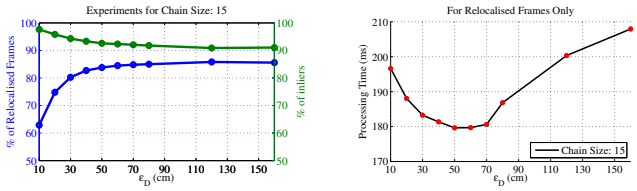
Our 6D relocalisation success criteria is necessarily more strict than when considering appearance-only relocalisation (e.g. [10]). Given a reference distance error threshold ϵ_D, a chain size c_{size} and a query image with its corresponding camera pose (obtained with visual odometry) with translation $\mathbf{t}$ and quaternion $\mathbf{q}$ for orientation, our algorithm will return a chain of 2D-3D pairs $\mathbf{C} = \{(\mathbf{s}_1, \mathbf{p}_{1B}), (\mathbf{s}_2, \mathbf{p}_{2B}), \ldots, (\mathbf{s}_n, \mathbf{p}_{nB})\}$, where n is the size of the chain. If the 3D point $\mathbf{p}_{iB}$ is an inlier then its projection on the camera pose should be close to the image point $\mathbf{s}_i$, i.e., $\mathbf{s}_i \approx \Pi(\mathbf{R}(\mathbf{q})(\mathbf{p}_{iB} - \mathbf{t}))$, where Π is the perspective projection model. We consider $\mathbf{p}_{iB}$ to be an inlier if the distance between its projection and its related image point $\mathbf{s}_i$ is less than a projection threshold, which in our experiments is set to be 2 pixels. A frame is considered relocalised only if the chain contains a minimum number of inliers, in this case, 5 points. Relocalisation success is measured as the ratio of relocalised frames over the total number of query frames which in this case is 13542 images. Note that we use every single frame in our test sequences captured during agile motion of the sensor.

B. Effect of 3D error tolerance

Figure 4a-b shows the effect of varying the reference distance error ϵ_D, which ranged from 10 to 160 cm. For these

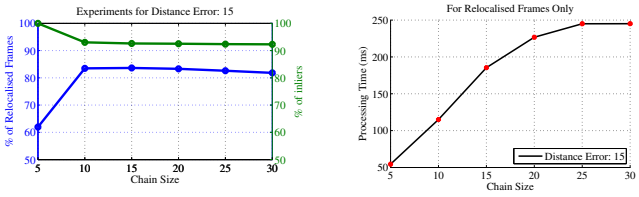
tests, we fixed the chain size $c_{size} = 15$. This distance error is one of the prime parameters affecting the relocalisation rate, the number of inliers as well as processing time. On one extreme, if this threshold is very large it is equivalent to not using any assistance from depth. It may happen that the algorithm can not find a valid chain quickly and thus we mitigate this by limiting the process to up 2 seconds. If this time limit is breached then next available frame is used. Those unsuccessful frames are excluded from the time computation but not for the relocalisation rate.

The effect of ϵ_D in the processing time (see figure 4b) is interesting since, if ϵ_D is very strict the number of inliers per chain is still high, but in contrast, the percentage of relocalisations drops. The latter means that the number of successful chains, although with high number of inliers, gets reduced when ϵ_D gets reduced. The increment in processing time in this case is due to the pruning of several candidate descriptors, which forces the algorithm to access more frequently to $\mathbf{Q}$ thus investing more processing time. On the other hand, if ϵ_D is very lax, the number of inliers is still high, however, it accepts as 'valid' too many descriptors in $\mathbf{Q}$ which clogs the algorithm. This graph alone is a good justification for the involvement of depth information in the process. This also clearly shows an optimal narrow window where processing time savings are achievable. Around $\epsilon_D >= 50cm$ the relocalisation percentage peaks. Numbers quoted are the average results for 10 runs.



(a) (b)

Fig. 4. Set of experiments where ϵ_D was set to different values. Note that the averaged percentage of relocalised frames (a) does not improve as the distance is increased. In contrast, the quality of the chain seems to decrease which is expected given that a large tolerance may introduce false positive matches. As expected, the processing time (b) increases as the distance error tolerance increases since the number of descriptors to compare to increases.



(a) (b)

Fig. 5. Set of experiments where the size of the chain is set to different values. Note that the best results are obtained for the size 10 and 15. Chains of bigger sizes do not seem to contribute to the percentage of relocalised frames nor to the percentage of inliers (a), on the other hand, there is an increment in computational effort (b).

C. Influence of chain size

Figure 5 shows tests for the effect of chain size. For this experiments we set $\epsilon_D = 50cm$. The averaged processing time showed in figure 5b indicates that the smallest size of the chain is too optimistic since the algorithm breaks as soon as it gets 5 features in the chain, but several of these chains did not contain enough inliers to accept the frame as relocalised (at least 5). However, when the chain size is increased to just 10 there is an immediate increase in the number of relocalised frames. The percentage of inliers does not increase with chains bigger than 10, but beyond this point the processing time does, see figure 5a. From this results, chain sizes between 10 and 15 appear reasonable choices.

TABLE I
RESULTS FOR SMALL SETS OF MATCHES

Method	Set Size	Reloc. %	Time (ms)	Pose Error (cm) Mean	Pose Error (cm) Std D.
Appearance Driven	7	39.4	776.3	22.6	33.1
	10	70.3	1737	20.3	27.4
	15	87.8	4811.2	19.5	25.9
	ALL	95.6	48173 .0	11.2	10.2
Using 3D Test 1 Anchor Only	7	68.9	72.2	15.8	18.8
	10	82.8	110.5	12.6	14
	15	83.7	177.9	10.2	11.6
	ALL	83.8	206.8	9.8	11.1
Using 3D Test Trying 1 or more Anchors	7	77.9	79.3	16	19.6
	10	93.6	120.9	12.5	14.8
	15	94.8	187.7	10.4	12.5
	ALL	94.8	220.1	9.9	11.7

D. Reduced Set of Quality Matches

The most important highlights from the previous analysis are threefold: (1) subject to proper tuning, our approach can potentially construct a successful chain of matches by using the first found anchor point in 80% of the cases; (2) successful chains returned by our algorithm contain around 90% inliers; (3) The previous results are obtained with relatively small chain sizes of 10-15. Therefore, we moved on to test in full our algorithm by connecting it to a three-point pose algorithm plus RANSAC.

For this experiments we compare the performance of the appearance-driven method for different number of matches, i.e.: the algorithm exits as soon as it finds a specific number of matches and then it attempts RANSAC to find the camera pose, we tested for 7, 10, 15 and 'ALL' possible matches that can be found. For our method we set c_{size} to the same number of matches mentioned before, however, we tested two configurations: (1) using only the first found anchor point; (2) In case the first anchor fails, just keep attempting until another anchor is found. Once the quality chain $\mathbf{C}$ is obtained, this is passed to the consensus procedure in order to relocalise the camera pose. The relocalised camera pose obtained with RANSAC, for both methods, was compared against that obtained during mapping. For this experiments we used the topological map and video sequences shown in figure 3.

Results are shown in table I where an immediate conclusion arises: a blind search performed by a 1-NN-like algorithm (appearance-driven) begins to drop its performance as the set of matches is reduced. This is due to the fact that the set may contain outliers in its majority and therefore,

RANSAC fails to estimate a pose. Moreover, not only the % of relocalisations drops, but the accuracy of the retrieved pose as well. In contrast, our method seems to maintain stability, except for the case with 7 matches where there is a reduction of around 14%. However, using only the first found anchor point results in around 80% of relocalisations for the rest cases. For the ALL case the algorithm did not get increased its computational time in a drastic manner, this confirms that the 3D test avoids unnecessary comparisons. Finally, an increase in relocalisation % is observed when more than one anchor point is tried (if the first fail). However, this does not over increase the computational time, which indicates that a successful chain was constructed using few attempts after the first one failed.

As a final note, observe that the relocalised camera pose error shown in table I is in average 15 cm. However, this relocalised camera pose is that returned by RANSAC without any post refinement. The latter could be done by assigning a big uncertainty to the relocalised camera pose and then using the matches in $\mathbf{C}$ to iteratively correct the pose with Kalman filter updates, such as it is done in [15], [14].

Fig. 6. Aerial vehicle used to captured challenging fast-motion sequences to test the processing time for an appearance-driven method against our approach. The RGB-D sensor was mounted on the vehicle and connected by cable to the processing laptop while flown manually.

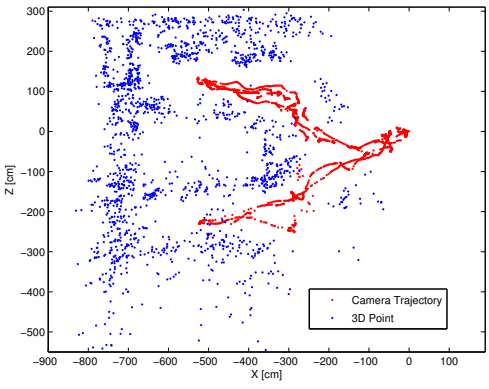


Fig. 7. Top view of the Lab scene where a map of 3D points was pre-built and later used to test the appearance-driven method against our approach. The map points are shown in blue and the trajectory of the vehicle is shown in red; notice the discontinuities in the trajectory which is due to moments of violent fast change in which tracking failed but it resumes as relocalisation was used during mapping too.

E. Performance under agile 6D motion

Our work is highly motivated by the scenario where a vehicle with fast agile motion has to explore the environment for which a 3D map representation is available. Typically, this map is used to track the 6D pose of such vehicle, but due to its erratic motion, sudden turns or vibration, the system may lose tracking and a rapid recovery method is necessary. This is the case of small aerial vehicles. We have used a small quadrotor to which we have attached the RGB-D sensor linked by an umbilical connection figure 6, to an i5 laptop. The platform is therefore not self contained but provides sample agile challenging motions when controlled manually. Similar to the previous experiments, we test relocalisation for every single frame of the test sequences, i.e. we have no tracking during testing. The attached video material also shows the performance of the system. For this experiment we set $c_{size} = 10$ and $\epsilon_D = 50$.

We have used this platform navigating in a pre-built map of a $64m^2$ environment and with 37K descriptors stored in the hash table $\mathbf{Q}$ (see figure 8). A second video sequence was used to test the relocalisation. No ground-truth was available for this experiment, however, the relocalised trajectory can be seen in figure 7. Finally, table II shows a summary of the results for this experiment. Note that strong erratic motion produced high image blur, which led both relocalisation algorithms to failure. We achieved an average time of 83 ms with our approach compared with 3000 ms when using the appearance driven method. Recall that our timings include from frame capture to 6D pose estimation. Our algorithm is left slightly behind w.r.t. the appearance-driven due to some frames where the vehicle flies too close to objects. The latter affects the generation of depth information from the sensor.

TABLE II
SUMMARY OF RESULTS FOR AGILE MOTION

Sequence	Method	# Frames	% Reloc.	Time (ms)
Parrot	3D Test	3638	73.6 %	**83.3**
	Appearance	3638	75.2 %	3000

VI. CONCLUSIONS

In this work we have presented a 3D geometric test for retrieving a set of quaility 2D-3D points which can be effectively used for camera pose relocalisation. The approach has the ability to work with maps built from Visual Odometry and full SLAM as well as with other sensors such as a stereo camera or a camera+LIDAR rig. We have compared our method with an appearance-driven approach previously used for 6D pose estimation and that is representative of some other visual methods that use quantization tables or visual words. However, our main objective has been that of demonstrating that, in the scenario where a set of matches has to be extracted from a list of candidates, it is possible to rule out 3D inconsistent candidates by using our low cost test. This leads to savings in computational time, but also to obtain a compact set of quality matches that can be used to effectively recover the camera pose.

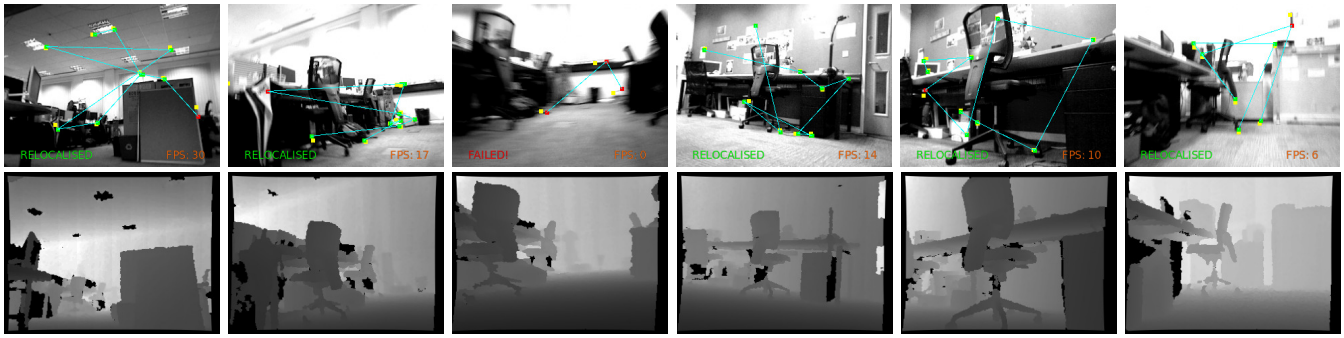


Fig. 8. Samples of the test sequence captured by the RGB-D sensor mounted on a quadrotor which flew around the scenario depicted in figure 7. The images show some examples where our algorithm found a valid hypothesis and some other examples where the algorithm fails to relocalise due to the sudden erratic motion of the vehicle.

Future work includes the adaptation of the method to other relocalisation processing pipelines that use different classifiers. We have already begun to investigate the use of different types of descriptors and organisation models [35] aiming to increase the speed of the relocalisation procedure. Other possible avenue for future work is the parallel implementation of our algorithm, which would enable the simultaneous construction of several chains thus increasing the pool of quality matches.

REFERENCES

[1] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, Monte Carlo Localization for Mobile Robots, ICRA 1999.

[2] J. Neira, J.D. Tardos and J.A. Castellanos. Linear time vehicle relocation in SLAM. ICRA 2003.

[3] A.J. Davison, Real-Time Simultaneous Localisation and Mapping with a Single Camera, ICCV 2003.

[4] R.A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A.J. Davison, P. Kohli, J. Shotton, S. Hodges and Andrew Fitzgibbon. KinectFusion: Real-Time Dense Surface Mapping and Tracking, ISMAR 2011.

[5] M. Agrawal, K. Konolige, and R.G. Bolles, Localization and Mapping for Autonomous Navigation in Outdoor Terrains: A Stereo Vision Approach, IEEE Workshop on Application of Computer Vision (WACV),2007.

[6] N. Engelhard, F. Endres, J. Hess, J. Sturm and W. Burgard, "Real-time 3D visual SLAM with a hand-held RGB-D camera", Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum. 2011.

[7] D. Chekhlov, M. Pupilli, W. Mayol-Cuevas, A. Calway, Real-Time and Robust Monocular SLAM Using Predictive Multi-resolution Descriptors. ISVC 2006.

[8] E. Eade and T. Drummond, Scalable Monocular SLAM, CVPR, 2006.

[9] D. Nister, O. Naroditsky and J. Bergen, Visual odometry, CVPR, 2004.

[10] M. Cummins and P. Newman, Appearance-only SLAM at Large Scale with FAB-MAP 2.0. The International Journal of Robotics Research. 2010.

[11] P. Rohan and P. Newman FAB-MAP 3D: Topological Mapping with Spatial and Visual Appearance. ICRA. 2010.

[12] A. Angeli, D. Filliat, S. Doncieux, J.A. Meyer, Incremental vision-based topological SLAM. IROS, 2008.

[13] B. Williams, G. Klein and I. Reid. Real-time SLAM Relocalisation. ICCV, 2007.

[14] D. Chekhlov, W. Mayol-Cuevas and A. Calway. Appearance Based Indexing for Relocalisation in Real-Time Visual SLAM. BMVC, 2008.

[15] E. Eade and T. Drummond. Unified Loop Closing and Recovery for Real Time Monocular SLAM. BMVC, 2008.

[16] B. Williams, G. Klein and I. Reid. Automatic Relocalization and Loop Closing for Real-Time Monocular SLAM. IEEE PAMI. 2011.

[17] K. Granstrom, T.B. Schon, J.I. Nieto, F.T. Ramos. Learning to close loops from range data. IJRR. 2011.

[18] J. Montiel, J. Civera and A. Davison. Unified Inverse Depth Parametrization for Monocular SLAM. Robotics Sience and Systems Conf. 2006.

[19] B. Williams, M. Cummins, J Neira, P Newman, I Reid and J Tards.A comparison of loop closing techniques in monocular SLAM. Robotics and Autonomous Systems.2009.

[20] P.F. Alcantarilla, K. Ni, L.M. Bergasa and F. Dellaert. Visibility learning in large-scale urban environment. ICRA, 2011.

[21] C. Mei, G. Sibley and P. Newman, Closing Loops Without Places. IROS 2010.

[22] L. Cadena, D. Cesar, J. McDonald, John, J. Leonard, J. Neira. Place Recognition Using Near and Far Visual Information. IFAC 2011.

[23] C. Cadena, D. Glvez, F. Ramos, J. D. Tardos, J. Neira. Robust Place Recognition with Stereo Cameras, IROS 2010.

[24] A. Howard. Real-Time Stereo Visual Odometry for Autonomous Ground Vehicles. IROS, 2008.

[25] M.Brown, R.Szeliski and S.Winder. Multi-image matching using multi-scale oriented patches. CVPR, 2005.

[26] A.P. Gee, A. Calway and W. Mayol-Cuevas. Visual Mapping and Multi-modal Localisation for Anywhere AR Authoring. ACCV Workshop on Application of Computer Vision for Mixed and Augmented Reality. 2010.

[27] J. Civera and O. G. Grasa and A. J. Davison and J. M. M. Montiel. 1-point RANSAC for EKF-based Structure from Motion. IROS, 2009.

[28] A. Irschara and C. Zach and J.-M Frahm and H. Bischof. "From structure-from-motion point clouds to fast location recognition," Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on , vol., no., pp.2599,2606, 20-25 June 2009

[29] G. Klein and D. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '07). IEEE Computer Society, Washington, DC, USA, 1-10.

[30] M. Cummins and P. Newman. Probabilistic appearance based navigation and loop closing. ICRA, 2007.

[31] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object match- ing in videos. ICCV, 2003.

[32] D. Nister and H. Stewenius. Scalable Recognition with a Vocabulary Tree. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR '06), Vol. 2. IEEE Computer Society, Washington, DC, USA.

[33] A. P. Gee, W. Mayol-Cuevas, 6D Relocalisation for RGBD Cameras Using Synthetic View Regression. Proceedings of the British Machine Vision Conference (BMVC'12). September 2012.

[34] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In Proceedings of the 9th European conference on Computer Vision (ECCV'06), Springer-Verlag, Berlin, 2006.

[35] J. Martinez-Carranza, Walterio Mayol-Cuevas. Real-Time Continuous 6D Relocalisation for Depth Cameras. Workshop on Multi VIew Geometry in RObotics (MVIGRO), in conjunction with Robotics Science and Systems RSS. Berlin, Germany. June, 2013