Boosting the computational performance of feature-based multiple 3D scan alignment by *iat*-k-means clustering

Nicola Pezzotti, Francesco Bonarrigo, Alberto Signoroni
Information Engineering Dept., DII
University of Brescia, Italy
{name.surname}@ing.unibs.it

Abstract

In this work we present a method to control and cut down the computational time required by feature-based multiple-view alignment solutions employed in modern 3D modeling pipelines. The reduction of the number of feature matches is guaranteed for each added view by means of an incremental (allowing dynamic views addition) and adaptive (variable number of clusters) implementation of a k-means clustering. The proposed method also comprises convergence quality and cluster cardinality control mechanisms, and guarantees multiple view alignment in nearly constant time with respect to the number of scans that need to be aligned for a significant class of feature descriptors. Moreover we demonstrate, on a representative experimental dataset, that the per-view alignment time can be reduced to a fraction of the corresponding pairwise alignment time without any performance degradation in terms of successful alignment. The obtained results are relevant for several 3D modeling applications where, especially for the acquisition of big and complex datasets, automation and robustness requirements are to be coupled with a quick and interactive usage of modern range scanners.

1. Introduction

The alignment of multiple 3D scans of an object (or scene) acquired from different viewpoints represents the first step of every 3D modeling pipeline [1], and can also be seen as a category of problems in the more general 'shape correspondence' domain [20]. Although the topic is a classic one in computer vision, the increasing spatial resolution attainable by high-precision scan devices nowadays employed in many professional application fields (industrial, biomedical, cultural heritage) has given rise to new criticalities that need to be addressed. In this work we focus on the first stage of a 3D modeling pipeline, where multiple independently referenced scans are collected and need to be co-referenced within a unique coordinate system. This is usually referred to as *coarse alignment*, which is then followed by *fine* and/or *global* registration that properly refine the alignment. Our objective is to boost the performance of the coarse alignment in its most challenging setting, that is when big sets of scans, acquired through highly resoluted acquisition devices, need to be aligned in an automatic, fast and reliable way. We therefore consider: 1) high alignment automatism, 2) computational speed (which allows to give immediate user feedback) and 3) absence of constraints about the acquisition path, as three pivotal factors to guarantee an efficient and interactive object acquisition. The coarse alignment phase is typically devised as a pairwise approach, implicitly assuming the availability of a concatenated acquisition path, i.e. where each newly acquired view possesses a certain overlap area with the previously acquired one. However, this is not always a viable solution, since a more unconstrained (not concatenated) acquisition path is likely to be requested in many real-life scenarios. On the other hand, straightforward multi-view extension of such pairwise approaches would quickly lead to computational issues whenever they are based on pure combinatorial or exhaustive approaches. This is especially true, as we will see, for feature-based approaches, where the number of feature matches are bound to increase proportionally with respect to the number of views that need to be aligned. In this perspective, feature space organization solutions, such as partitioning or reduction of the feature collection, are desirable in order to keep the computational burden under control. In this work we address the above issues with the objective to find new feature space clustering methods that allow the implementation of an effective multi-view coarse registration, which in turn should enable a progressive and interactive object acquisition according to the three pivotal factors previously stated.

This paper is organized as follows: after analyzing some related work (Sec.2), we consider possible multi-view extensions of feature-based pairwise coarse alignment methods. In particular, the general schemes of both 'direct' and

978-0-7695-4873-9/12 $26.00 © 2012 IEEE
DOI 10.1109/3DIMPVT.2012.32

'clustering-based' extensions are introduced in Sec.3. In Sec.4 we consider the computational issues of the direct extension, and reformulate the computational problem by introducing a variant of the classic k-means algorithm, which we address as *iat*-k-means. The proposed clustering technique is tailored in such a way so as to improve the organization of the feature space to allow for a more efficient multi-view extension of the coarse alignment. Although the proposed approach can be seen as a functional layer independent from the specific application, we will also demonstrate (Sec.5) that an implementation of our clustering technique for a reference coarse alignment technique is capable of guaranteeing multiple view alignment in a highly reduced and nearly constant time (with respect to the number of scans that need to be aligned). This will be demonstrated in Sec.6 by performance and computational comparisons on a well-assorted set of multi-view object acquisitions.

2. Related work

Despite the fact that coarse alignment methods usually employed in modeling pipelines are typically pairwise, a few multi-view solutions have already been presented, although limited by some constraints. A *pseudo*-multi-view approach has been first proposed in [6], where an exhaustive pairwise alignment is carried out, followed by a global optimization that tries to generate a graph of the reconstructed object. In [14] a single-to-multiple view alignment is proposed, based on a tensor feature indexed in a hash table and a voting mechanism used to build a spanning tree graph of rigid spatial transforms to organize the multi-view alignment. In [19] an alignment of unordered views of an object is obtained under some hypothesis: quadruples of scans on cardinal points of an object are combined in small meshes, which are then assembled with others through a PCA procedure. In all these methods single scans are always converted in meshes and the problem of the alignment is seen as a surface matching or shape correspondence problem [20]. It is important to note that all the above multi-view solutions suffer either from some limiting hypothesis (e.g. on the kind of objects or data that can be handled), or due to their computational burden (entailed by the exhaustive nature of the proposed solution).

In the past few years, pairwise feature-based multi-scale approaches such as [10, 4, 2] have proven to be effective methods for an automatic coarse alignment of scanned datasets (range images, point clouds or meshes). The first two methods are mesh-based, i.e. features are extracted from meshes generated out of single range scans, while [2] relies on features directly extracted from the original range data, and was shown to be robust (near 100% correct alignments for a variety of high-resolution, real objects) and fast enough for practical scanner usage (few seconds are required to align each view, composed by a million points each), at least in a pairwise perspective. A recent direct multi-view extension of [2] clearly shows that the alignment time is dependent with respect to the number of views (which in real life acquisition settings can range from dozens to hundreds of scans per object) [3]. To the best of our knowledge, no effective multiple view coarse alignment technique has been presented up to now that is capable of satisfying all three requirements described in Sec.1. Moreover, in a feature space organization perspective, we also did not find any clustering technique suitable to guarantee the necessary flexibility to provide incremental and adaptive reorganization of the feature space, as well as the complexity reduction mechanisms that we introduce in this paper.

A popular approach to organize the feature space is represented by the Bag of Words (BoW) model [16]. BoW solutions have been proposed in the field of similarity matching (e.g. object retrieval in large databases), they have also been used in one case [9] for view alignment purpose. However, it is worth noting that there is little in common between the BoW approach with respect to our solution, despite the fact that both aim to some kind of feature space organization. In fact, BoW approaches are not suitable to an incremental and interactive acquisition pipeline because a) they require that all the scans should be available in advance, b) they present moderate to high computational complexity for the codeword dictionary construction (this is why off-line dictionary and database matching make them suitable for retrieval applications) and c) 'feature vs codeword' matches are sufficient for similarity seeking, while for alignment purpose accurate feature localization is essential and 'feature vs feature' matching is always necessary at least within a cluster-of-interest (where cluster centroids are instead used for faster clusters-of-interest search).

Other application-specific clustering techniques have already been derived from the popular k-means [12] algorithm. A review of the variants proposed, such as the large scale clustering, clusterization of heterogeneous data, or semi-supervised clustering, has been proposed in [7]. The *iat*-k-means approach described in Sec.4 presents some similarities with respect to the global k-means [11], as well as with the sequential k-means [12]. However, it also detaches from them in some essential and distinctive aspects: in fact our approach tries to minimize the standard deviation of the cluster dimension, as well as keep the average cluster dimension fixed while adding new elements in the feature space.

3. Multiple view alignment extension

A direct and intuitive extension of a feature-based pairwise alignment pipeline, capable of aligning a set of scans without the severe constraint related to concatenated acquisition path, is shown in Fig.1. Here each new view is only required to have a partial overlap with respect to at least one

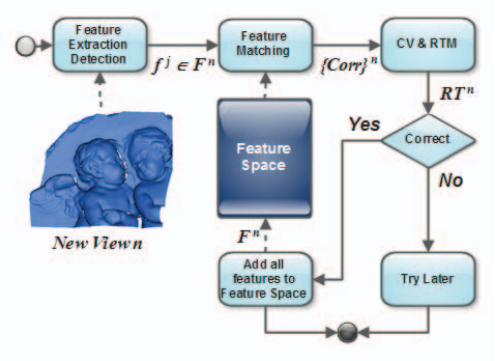


Figure 1: General multi-view alignment pipeline.

of the previously aligned views, rather than requiring overlap with the previous one. This kind of approach has been already considered in a recent work [3], where a linear dependency of the alignment time with respect to the number of previously aligned views was clearly visible in the experimental results. This is caused by the fact that features belonging to any of the previously acquired views need to be stored (possibly avoiding spatial duplicates) in a suitable structure (some kind of feature table), and for each new feature an exhaustive match has to be performed against all the features in the table, leading to a progressive growth of the matching time. In this work we aim to tackle the computational issue described above while trying to preserve the working principles of the scheme portrayed in Fig.1, which demonstrated to be both flexible and practical[1]. An extension of such general scheme that considers a feature space organization is presented in Fig.2(a), where two new blocks are evidenced in red. The upper block has the scope of identifying the most similar cluster-of-interest, which is obtained by matching each new feature with respect to all the cluster centroids, while the subsequent 'feature vs feature' matching is limited to the ones within the cluster of interest. The lower red block represents the feature space update, which is now organized in clusters. Although an extension of the matching process that foresees feature space clusterization can be quite intuitive, it does not automatically involve a computational gain because the feature space management will also introduce an overhead, especially considering that an update of the feature space has to be performed at every aligned view. This is, however, a necessity since leaving the feature space statically organized would imply to lose control on the number of features that populate each cluster, thus giving again rise to the problem of linear growth of computation time.

[1]For example, unsuccessfully aligned views could be easily handled by moving them in a waiting list and reconsidered them at a later time ('Try Later' block in the scheme), until alignment is achieved.

4. Proposed solution

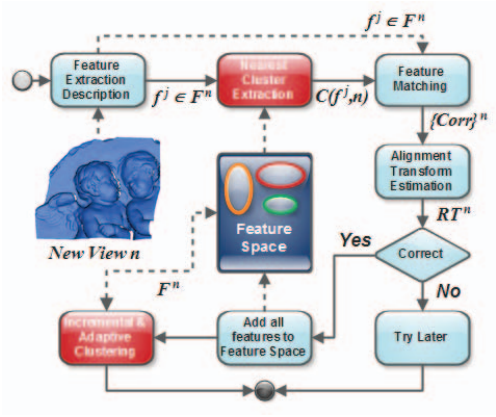
We propose a solution which allows to reformulate the computational load of an interactive multi-view alignment process from the one associated to the direct extension (Sec.4.1), to one obtained through a particular clustering-based extension. The reduction of feature matches can be guaranteed, at each added view, by means of an *incremental* (allowing online views addition) and *adaptive* (variable k) implementation of a k-means clustering (Sec.4.2). As clustering here is simply functional to reduce the number of feature matches (that is, not strictly oriented to optimal space partitioning), in Sec.4.3 we introduce a quality verification mechanism that allows early *termination* before complete (and possibly computationally expensive) convergence is reached, thus limiting the computational burden associated to the clustering update process. We address our approach as *iat*-k-means, by the initials of the above three keywords.

4.1. Computational complexity issues

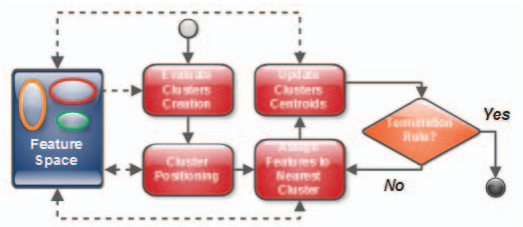
As stated, our main objective is the limitation of the feature matching time in a multiple view alignment pipeline. In fact, the direct extension approach described in Fig.1 needs an exhaustive matching for each of the features extracted at the n-th view (constituting the set F^n) against the set F_{db} representing all the non-repeating features collected up to view $n-1$, therefore giving rise to a problematic dependency of the matching time $T_{F^n \to F_{db}}$ from the number of views. In average, this can be estimated as:

$$T_{F^n \to F_{db}} \approx \alpha \tilde{f}\,(n-1)\,\tilde{F} \tag{1}$$

where $\tilde{f} = |F^n|$ is the average number of features present in the view currently being added, while $\tilde{F} = |F^n| - |F^{n-1}|$ is the average number of non-repeating features collected in F_{db} at each new view, and the coefficient α represents the cost of a single feature match. In principle, a clustering of the feature space at each stage n could be used to reduce the number of feature matches from $\tilde{f}(n-1)\tilde{F}$ to $\tilde{f}\tilde{C}$, where $\tilde{C}$, at the moment, represents a general idea of the average dimension of feature clusters in the feature space. However, several problems arise about (i) the control of $\tilde{C}$ at each added view, (ii) the computational load associated to the search of the nearest cluster for each feature of the added view, (iii) the cost of updating the cluster at each added view and, most importantly, (iv) the reduction of feature matches must not determine any sensible degradation of the overall automatic alignment performance. We are, therefore, interested in a solution that addresses all these problems (from (i) to (iv)) by guaranteeing multiple view alignment in a reduced time, irrespective of the number of added views.



(a)



(b)

Figure 2: (a) Proposed multi-view alignment pipeline; (b) expansion of the Incremental and Adaptive Clustering block.

4.2. Incremental and adaptive k-means clustering

To separate features into subsets we adopt a clusterization approach consisting in a modified version of the classic k-means algorithm [12] on the space of d-dimensional feature signatures. First, the clustering must necessarily be *incremental*, in that it must be capable to handle the insertion of new features at certain points of the 'means' (i.e. the cluster centroids) migration. To do so, after each successful alignment, new features are added to the set F_{db} and associated to existing clusters, and a k-means update is performed (more details on how to do this in a balanced way with respect to both computational load and convergence issues will be given in the next subsection). With the aim of cutting down the computational complexity of the feature matching with respect to the elements within a cluster, we are also strongly interested in exerting a control on the average dimension of each cluster. Therefore, we also devise an *adaptive* clustering where the number of clusters at the n-th view k^n is varied in order to guarantee:

$$F_{db} = \bigcup_{i=1}^{k^n} C_i \qquad \text{where} \begin{cases} \sum_{i=1}^{k^n} \frac{|C_i|}{k^n} \leq C_{size} \\ C_j \cap C_i = \varnothing \quad \forall j \neq i \end{cases} \tag{2}$$

where C_i is one of the k clusters that partition the set F_{db}. In practice, k should increase when C_{size} new features need to be added in the feature database, and this can be obtained by splitting the largest clusters. To do so, a ranking of the clusters' dimension is performed, then the splitting is repeated on as many clusters as needed to satisfy (2). It is important to note that this approach does not guarantee that every cluster dimension stays below C_{size}: in fact some clusters are allowed to grow a little bigger, while the biggest are split. With respect to a hard policy where every partition bigger than C_{size} gets split, the one proposed allows to reduce the overall number of partitions created, thus speeding up the matching. Once a cluster is split, for each of the new clusters a random feature is selected as centroid, and the k-means update process is performed. The incremental and adaptive k-means approach described up to now will be referred to as *ia*-k-means, in contrast to the previously stated *iat*-k-means which also include the termination rule that we will introduce in the following section.

4.3. Complexity analysis and control

A potential reduction of the matching time through feature clustering does not come without costs. Using the above *ia*-k-means approach, the matching time (at scan n) can now be estimated as:

$$T_{F^n \to F_{db}} \approx \alpha \tilde{f} C_{size} + \beta \tilde{f} k^n + t_C^n \tag{3}$$

where problems (i) to (iii) described in Sec.4.1 are clearly visible in the right hand term. In the first term of the sum, α represents the cost of 'feature vs feature' matching, which is performed for each feature within the set F^n (which average size is $\tilde{f}$) with respect to the features that populate the selected cluster (with average size C_{size}). In the second term, β is the cost of 'feature vs cluster centroid' matching, where each feature within F^n has to be matched with the cluster centroids, which number at scan n is k^n, and eventually t_C^n is the cost of cluster update at scan n.
Considering again the first term we can observe that, if $C_{size} < \tilde{F}$, we can expect a reduction of the matching time even with respect to the reference pairwise alignment time. The second term of the sum depends from n, and therefore this introduces again a linear dependency with respect to n. However, as we will see in the next section, the second contribution can be kept marginal with respect to the first one since we will show (under proper assumptions) that we can reach $\beta \ll \alpha$ with no alignment performance degradation, thus greatly reducing the rate at which the linear dependence on n impacts on the overall alignment time (as we shall see, this contribution is negligible for the practical application highlighted in this work). The third term t_C^n would also increase with n because of the increasing number of features, however two aspects are worth noting: 1) in our incremental approach an update is performed after each added view n, therefore requiring only minor adjustments to the clustering; 2) we are not interested in true convergence

at each step n (we are not performing classification) and therefore an early termination rule can be introduced (under proper quality constraints). Actually, this turns out to be an effective way to keep t_C^n low and almost constant, as we will also see experimentally. To this end, we define two threshold values, $\hat{T}$ and $\hat{Q}$, which respectively constraint the minimum execution time of the *ia*-k-means updating and the minimum clustering accuracy according to the following quality measure:

$$Q_C^n = \frac{100}{|F_{db}|} \cdot \sum_{i=1}^{|F_{db}|} Q(f_i)$$
$$Q(f) = \begin{cases} 1 & C^j(f) = C^{j-1}(f) \\ 0 & C^j(f) \neq C^{j-1}(f) \end{cases} \quad (4)$$

where $C^j(f)$ indicates the cluster that contains the feature f during iteration j of the k-means update performed after aligning the n-th scan. In short, Q_C^n represents the percentage of features which do not change cluster at the j-th k-means update iteration (when true convergence is reached, Q_C^n reaches the value 100%). We then continue to execute k-means iterations until the following termination rule is satisfied:

$$((t_C^n > \hat{T}) \wedge (Q_C^n > \hat{Q})) \vee (Q_C^n = 100\%) \quad (5)$$

According to (5), the algorithm continues to improve the quality of the solution until a time threshold is reached. If t_C^n reaches $\hat{T}$ without a minimal quality $\hat{Q}$, the iterations continue until the quality threshold is reached. In Sec.6 we will experimentally see how this is enough to guarantee good alignment performance even for relatively low values of $\hat{Q}$. The *ia*-k-means which also comprise such an early *termination* mechanism will be referred to as the *iat*-k-means.

5. Proposed Implementation

We now specialize our general method to the common case of circular feature descriptors, composed by N angular sectors and M radial sectors. In order to match two features in the most reliable way, a circular correlation should be computed in order to determine the best relative orientation between the two features, requiring a number of sector comparisons equal to $M \cdot N^2$. One way to reduce this computational complexity is to determine the orientation of each feature through the estimation of a principal direction ([17], [18]), thus obtaining directly comparable 'oriented' features (this problem has been addressed to as the "Local Reference Frame" search in [15]). However, a direct exploitation of this complexity reduction is likely to cause a degradation of the correspondences' reliability, since the principal direction may not always be correctly estimated. This is why in our implementation we exploit principal feature orientations (estimated through an algorithm presented in [13]) only to speed up the 'feature vs cluster centroid' matching, thus allowing us to obtain $\beta = M \cdot N$. In our implementation we adopt the descriptors proposed in [2] ($N = 32$, $M = 3$), where in order to handle undefined sectors of the feature descriptor due to incomplete representation of the object (holes, undercuts, ...), we treat them as zero value sectors and adopt the following modified 'feature vs cluster centroid' distance function between a feature descriptor A and a centroid descriptor B:

$$D(A,B) = \frac{\sum_{i=1\ldots N, j=1\ldots M} U(A_{i,j}, B_{i,j}) \cdot |A_{i,j} - B_{i,j}|}{\sum_{i=1\ldots N, j=1\ldots M} U(A_{i,j}, B_{i,j})} \quad (6)$$
$$U(s_1, s_2) = \begin{cases} 0 & \text{undefined}(s_1) \vee \text{undefined}(s_2) \\ 1 & \text{defined}(s_1) \wedge \text{defined}(s_2) \end{cases}$$

For the identification of the corresponding feature within the cluster (i.e. the 'feature vs feature' matching), in order to obtain the best possible discrimination, we resort to a circular correlation that is resilient to undefined sectors, as described in [2].

Another way we followed to reduce the 'feature vs cluster centroid' matching time led us to exploit the high distinctiveness between the cluster centroids by means of an angular subsampling of their feature signature by a factor S, therefore reducing the number of comparisons for each matching to $M \times \lfloor \frac{N}{S} \rfloor$, leading to $\beta = \frac{\alpha}{N \cdot S}$ in (3). In our implementation we used $S = 4$, thus obtaining a speed-up factor of 128, leading to the desired result $\beta \ll \alpha$. As we will see in Sec.6, this simplified cluster search did not degrade the overall alignment performance, giving our approach a tremendous computational time reduction. Some of the original saliency-based feature descriptors are presented in Fig.3(a), while some examples of how our centroid descriptors look like are presented in Fig.3(b).

The two factors that allowed us to obtain $\beta \ll \alpha$ in our implementation (that is, avoiding the circular correlation and

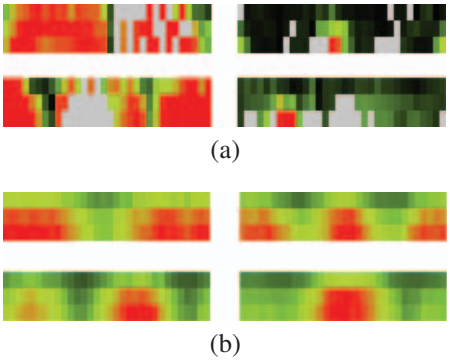


Figure 3: (a) Feature signatures obtained according to [2], gray sectors are undefined; (b) Cluster centroids signature obtained with *iat*-k-means algorithm.

Figure 4: Datasets used in the experiments, from left: Cupid, Shell, Carter, Slide, Neptune, Hurricane, Capital, Venus.

subsampling the feature signatures), may not be feasible for different types of features ([5], [8]). In the worst case (that is, when $N = S = 1$, leading to $\beta = \alpha$), the matching time for our approach would grow linearly with respect to the number of features in the database, but thanks to the clustering this linear increment is reduced by a factor of C_{size} with respect to an exhaustive matching.

Once the correspondence matches have been obtained through the described approach, in order to filter out possible outliers from the correspondence set we follow the algorithm described in [2]. The approach turned out to be robust, particularly for cases when the percentage of inliers in the correspondence set was very low.

Once a new view has been successfully aligned, its feature points are integrated into the feature table. In order to prevent the inclusion of features that are already present in the table, for each new feature to be added its spatial position is evaluated to identify whether it is already included in the table. If this is the case, only a single feature is maintained, while the others are discarded. In order to retain the feature with the most significant signature, the one which has its normal better aligned with respect to the acquisition point of view is kept, supposing that in such conditions the acquisition should better capture the surface, minimizing occlusions.

6. Experimental results

Tests have been made on a well-assorted variety of scan datasets (see Fig.4) in terms of object attributes and application fields, some of which have been directly provided by the authors of [2], while others have been acquired with a high-resolution structured-light scanner. Each dataset is composed by numerous range images, each containing a high number of features (see first 4 cols of Tab.1), as well as many repeated details (especially for the Shell and Carter datasets) which give rise to similar feature signatures, possibly causing misalignments. We compare the performance of our matching technique (which works according to the scheme of Fig.2) with respect to the direct multi-view extension introduced in [3], for which an exhaustive search is performed with respect to all the features present in the database (as in Fig.1). As shown in Tab.1, for a typical selection of parameters, our algorithm allows to incrementally align sets of range images in a greatly reduced time with respect to the exhaustive approach, proving the effectiveness of our *iat*-k-means solution. We have also verified the absence of degradation in terms of correct alignment percentage. Surprisingly, we also observed that for the Neptune dataset we had an increased percentage of correct alignments. This is probably due to the fact that the exhaustive search could potentially generate many false positives caused by features with lots of invalid sectors. On the contrary, our space clusterization tends to isolate all these features into some clusters, actually avoiding these misleading matches. After some empirical tests on a veriety of datasets, we have determined that a good value for the C_{size} is around 50.

In Tab.1 we also present the computation time required by our approach, split into two terms: the matching time (col. 6) and the *iat*-k-means update time (col. 7). We discriminate the two terms since, while the matching directly impacts on the time the user has to wait before the system can show the resulting alignment, the *iat*-k-means update can be performed while the user adjusts the scanning head in order to acquire the following scan, thus resulting transparent to the user. In Fig.5 we show the evolution of the alignment time for the Shell dataset. For the exhaustive matching approach (in yellow), we can note a linear increase in the alignment time as long as new, non-repeating features are to be included into the feature database, that is until the entire object surface has been acquired (range image from 1 to 50). From this point on, thanks to the feature duplicates avoidance mechanism, the number of features in the database does not increase significantly. Peaks and valleys in the graph are generated by range images that present a very high, or low, number of features extracted with respect to the average parameter $\tilde{f}$.

Thanks to the early termination rule, *iat*-k-means updates are executed in nearly constant time, as shown in Fig.6. It is worth noting that, as Fig.7 testifies, such termination rule does not lead to alignment performance degradation even for low values of quality threshold $\hat{Q}$. This fig-

Dataset	RI #	Points	#Extr. feats	I) % RI aligned	I) Avg t match [s]	I) Avg t update [s]	II) % RI aligned	II) Avg t direct [s]	III) Avg t pairw. [s]
Hurricane	32	22,032k	7626	100%	**0.473**	0.560	100%	21.8	2.163
Capital	36	20,012k	3320	100%	**0.442**	0.385	100%	6.5	2.242
Cupid	45	13,645k	7729	100%	**0.657**	0.552	100%	17.9	0.746
Slide	48	6,939k	1026	95.83%	**0.048**	0.106	95.83%	0.39	0.290
Venus	61	50,961k	9932	100%	**0.424**	0.459	100%	16.3	1.701
Carter	70	38,316k	10481	100%	**0.517**	0.562	100%	29.6	1.185
Shell	98	74,125k	11195	100%	**0.699**	0.651	100%	46.1	5.285
Neptune	79	54,310k	15251	**96.20%**	**0.759**	0.639	93.67%	23.3	1.995

Table 1: Matching results with $C_{size} = 50$, $\hat{Q} = 75\%$, $\hat{T} = 600msec$. Percentage of correctly aligned views (I), average matching and *iat*-k-means update time (per view) for the proposed method and for the direct extension [3] (II). Pairwise alignment time (III).

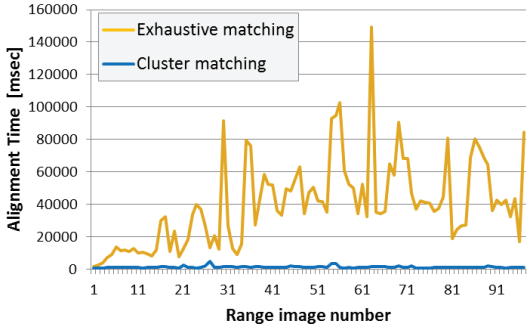


Figure 5: Comparison between alignment time required by the exhaustive matching based on [2] and our cluster matching solution.

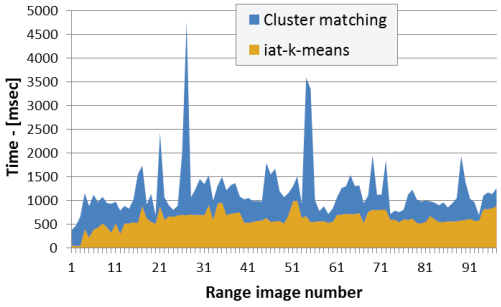


Figure 6: Alignment time decomposition between the cluster matching and the *iat*-k-means updating for our solution.

ure also shows that the time threshold $\hat{T}$ set to 600 msec grants that the iterative *iat*-k-means update process is run long enough to obtain an adequate accuracy of the feature clustering, which is suitable for our scan alignment purpose. While the value $\hat{T}$ should be tuned with respect to the performance of the hardware at hand, the threshold $\hat{Q}$ can be set according to the features distribution within the space:

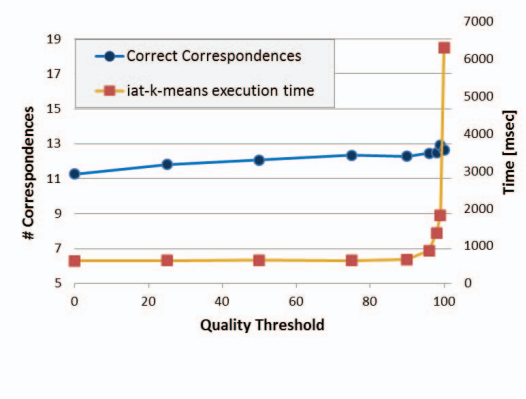


Figure 7: Evolution of *iat*-k-means execution time and number of correct correspondences with respect to $\hat{Q}$, with fixed $\hat{T} = 600msec$, on Shell dataset.

if an object presents many similar features, then it may be necessary to increase $\hat{Q}$. This said, for all the datasets presented no fine-tuning was necessary (not even for the Shell or Carter datasets, which presents many repeating features). On average, the matching time required by our technique for each new scan alignment is well below one second even for a feature database composed by more than 15000 features. Although the tests presented in this section have been performed on high quality range scans we can, however, infer some considerations about the approach resilience with respect to scan quality. While high-frequency noise (i.e. outliers) can be easily filtered out beforehand and should not affect the alignment performance, low-frequency deformations (such as the ones that usually affect Kinect scans) would cause the quality of the feature descriptors to degrade, thus compromising performance. In conclusion, it is important to notice that while 'feature vs feature' matching time remains constant due to the fixed average size of the

clusters, as well as the *iat*-k-means update time, which is bound through the termination rule, 'feature vs cluster centroid' matching time presents a linear increase due to the progressive increment in the number of clusters. However, the time required by a single 'feature vs cluster centroid' match has been significantly decreased (in our implementation, 128 of such matches require the same computational overhead of a single 'feature vs feature' match), as we described in Sec.5, therefore remaining well balanced with respect to the other two time factors. The results presented in this section let us conclude that we have achieved the objective of obtaining multiple view alignment in nearly constant time, regardless to the number of previously aligned views.

7. Conclusions

In this work we investigated the problem of feature organization with the aim of reducing the computational burden required by an incremental, feature-based, multi-view alignment system employed to align sets of numerous and dense 3D scans. We presented the *iat*-k-means, an effective variant of the k-means algorithm which succeeds in maintaining the computational complexity of the matching process nearly constant, despite the increase of feature numerosity. We also presented a specific implementation of our clustering for an automatic alignment technique that, in its pairwise version, demonstrated to be particularly robust and accurate, as well as competitive with respect to computational time, but increasingly (linearly) slower in its multi-view version due to the exhaustive matching it required. In contrast, our adaptive clustering is capable of maintaining the matching time practically constant throughout the incremental alignment procedure, regardless to the number of views previously aligned. Through the modifications proposed for the feature matching procedure, we also substantially reduced the time required by the feature matching itself without any degradation of alignment accuracy. We conclude that our clustering technique is particularly suited to be integrated into manual or robotic acquisition pipelines where precision, robustness and automatism are of the utmost importance.

References

[1] F. Bernardini and H. Rushmeier. The 3D model acquisition pipeline. *Computer Graphics Forum*, 21(2):149–172, 2002. 1

[2] F. Bonarrigo, A. Signoroni, and R. Leonardi. A robust pipeline for rapid feature-based pre-alignment of dense range scans. In *ICCV 2011*, pages 2260–2267, 2011. 2, 5, 6, 7

[3] F. Bonarrigo, A. Signoroni, and R.Leonardi. Multi-view alignment with database of features for an improved usage of high-end 3d scanners. *EURASIP Journal on Advances in Signal Processing*, 2012. in press. 2, 3, 6, 7

[4] U. Castellani, M. Cristani, S. Fantoni, and V. Murino. Sparse points matching by combining 3D mesh saliency with statistical descriptors. *Computer Graphics Forum*, 27(2):643–652, 2008. 2

[5] C. S. Chua and R. Jarvis. 3d free-form surface registration and object recognition. *International Journal of Computer Vision*, 17(1):77–99, 1996. 6

[6] D. F. Huber and M. Hebert. Fully automatic registration of multiple 3d data sets. *Image and Vision Computing*, 21(7):637 – 650, 2003. 2

[7] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651 – 666, 2010. 2

[8] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999. 6

[9] S. Khoualed, U. Castellani, and A. Bartoli. Semantic shape context for the registration of multiple partial 3d views. In *BMVC 2009*, 2009. 2

[10] X. Li and I. Guskov. Multi-scale features for approximate alignment of point-based surfaces. In *Symp. on Geom. Processing, SGP 2005*, 2005. 2

[11] A. Likas, N. Vlassis, A. Likas, N. Vlassis, and J. Verbeek. The global k-means clustering algorithm. *Pattern Recognition*, 36:451–461, 2001. 2

[12] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967. 2, 4

[13] C. Maes, T. Fabry, J. Keustermans, D. Smeets, P. Suetens, and D. Vandermeulen. Feature detection on 3d face surfaces for pose normalisation and recognition. In *Biometrics: Theory Applications and Systems (BTAS)*, 2010. 5

[14] A. Mian, M. Bennamoun, and R. Owens. Automatic multiview coarse registration of range images for 3d modeling. In *IEEE Conference on Cybernetics and Intelligent Systems, 2004*, volume 1, pages 158 – 163 vol.1, dec. 2004. 2

[15] A. Petrelli and L. Di Stefano. On the repeatability of the local reference frame for partial shape matching. In *ICCV 2011*, 2011. 5

[16] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering objects and their location in images. In *ICCV 2005*, volume 1, pages 370 – 377 Vol. 1, oct. 2005. 2

[17] B. Steder, R. B. Rusu, K. Konolige, and W. Burgard. Narf: 3D range image features for object recognition. In *Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics, IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 8, 2010 2010. 5

[18] Y. Sun and M. Abidi. Surface matching by 3d points fingerprint. In *ICCV 2001*, volume 2, 2001. 5

[19] F. ter Haar and R. Veltkamp. Automatic multiview quadruple alignment of unordered range scans. In *IEEE International Conference on Shape Modeling and Applications, 2007*, pages 137 –146, june 2007. 2

[20] O. van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. A survey on shape correspondence. *Computer Graphics Forum*, 30(6):1681–1707, 2011. 1, 2