

Accurate and automatic alignment of range surfaces

Simone Fantoni, Umberto Castellani and Andrea Fusiello
 Dept. of Computer Science - University of Verona, Italy
 name.surname@univr.it

Abstract

This paper describes an automatic pipeline that is able to take a set of unordered range images and align them into a full 3D model. A global voting scheme is employed for view matching, inspired by 2D techniques for image mosaicing. Then a multiple view registration approach is introduced, which aims at optimizing the alignment error simultaneously for all the views. Experiments demonstrate the effectiveness of the method.

1. Introduction

Three-dimensional (3D) registration of range images acquired by scanners is still a critical issue for obtaining a complete 3D model of objects or buildings [3]. In particular, as defined in the taxonomy suggested by [14], when several (> 2) scans are involved and initial pose estimates are unknown, the problem is called *multiview surface matching*. Three main interrelated sub-problems need to be solved: i) determining which views overlap, ii) determining the relative pose between each pair of overlapping views, and iii) determining the absolute pose of the views.

Many works have been proposed that tackle these issues but few of them address the three sub-problems at the same time. The first sub-problem aims at improving the automation of the process, and it is relevant especially when the acquisition sequence is not available. The output of this stage can be encoded in an *adjacency* matrix that contains overlap information. Pairwise registration (sub-problem ii) is customarily solved with the Iterative Closest Point (ICP) algorithm [4], which represent the gold standard for this problem. As observed in [14], there is a mutual dependency between the overlap and relative poses: If the relative poses are known the overlap can be easily computed, and vice-versa. Hence, in general, these two phases are carried out in a cooperative fashion. Finally, once the adjacency matrix is known and relative poses are available, multiple view registration (sub-problem iii) is addressed, where the solution of the absolute pose estimation is computed simultaneously for all views.

In this paper we tackle all the three aforementioned sub-problems by proposing a multiview surface matching pipeline which deals with automation of the process and accuracy of results. More specifically, our main contribution is twofold:

- we propose a new *feature-based* scheme for 3D partial view matching which makes the computation of adjacency matrix fully automatic, and
- we extend a pairwise registration algorithm [12] in order to cope simultaneously with multiple views.

2. Previous work

Roughly speaking *multiple-view* registration techniques cope with two main issues: (i) error accumulation and (ii) the automation of the process. In the following we report the state-of-art according to these two dimensions.

Reducing error accumulation. When the ordering of the sequence of views V_1, \dots, V_n is available, the registration can be performed pairwise between consecutive views (i.e., between views V_i and V_{i+1}) and therefore each view can be moved to the global reference system by concatenating pairwise transformations. In general, even if all the pairs are apparently well registered, misalignments appear when the full model is reconstructed due to the accumulation and propagation of the registration error. The general idea of multiple-view registration techniques is to solve *simultaneously* for the global registration by exploiting the interdependence between all views at the same time. This introduces additional constraints which reduce the global error. A comparative study of similar multiple-view registration methods was performed in [11]. In [26] a method is described that first performs registration pairwise, and then uses the estimated pairwise transformations as constraints in a global multi-view stage. In [10] a method that distributes registration errors evenly across all views was presented. It works in the transformation space, and the ordering of the views is required. More recently, in [5] a non-linear optimization method is proposed which minimizes the registration error over the manifold

of rotations. Moreover, a subsampling procedure is introduced to improve the speed of corresponding points estimation. Finally, correspondence-estimation and multiview-optimization steps are iterated until convergence is reached. In [29] the authors extended [26], by representing the motions as dual quaternions, and by casting the multiview registration problem into a diffusion of rigid transformations over the graph of adjacent views.

It should be noted that [5, 29] do not optimize correspondences, which are considered fixed or allowed to vary in alternation with the optimization of the rigid transformations (but the convergence of such procedure is not discussed).

Automating registration. Especially when the full model is composed of a large number of scans the view order might not be available and therefore should be manually specified. Many methods were proposed to improve the automation of multiple-view registration. In [14] a global optimization process searches a graph constructed from the pairwise view matches for a connected sub-graph containing only correct matches. This phase is implemented by computing a Minimum Spanning Tree (MST) to eliminate incorrect but locally consistent matches. In [20] it is proposed to address the automatic alignment as a location-recognition-problem by integrating range data with 2D intensity images. In [24] each range image is represented by multiple tensors and a global voting scheme is proposed by allowing tensors of each view to be simultaneously matched with the tensors of the remaining views. In [2] a fast and robust technique is introduced for pairwise registration based on the alignment between coplanar random 4-points sets. Other approaches use both *global* and *local* pre-alignment techniques to select the overlapping views by computing a coarse alignment between all the pairs. In [22] the pre-alignment is performed by extracting global features (namely, extended Gaussian images) from each view. Conversely, in [19] the pre-alignment is computed by comparing local signatures of feature points. Then, the best view sequence is estimated by solving a standard Travelling Salesman Problem (TSP). Similarly, in [6] a local feature point approach is proposed which emphasizes the speed of the procedure rather than accuracy. Note that in [6] the ordering of the sequence must be known in advance. An exhaustive comparison between local feature-based methods for partial shape matching is reported in [25].

Our approach addresses both the aforementioned issues. It automatically estimates the adjacency matrix of the views by employing a local feature-based technique, and solves the simultaneous registration among all the views by minimizing a global alignment error. Point correspondences can vary during the optimization procedure.

3. Method

Our pipeline is composed of three main steps: i) keypoint detection and description, ii) overlap estimation by keypoint matching, and iii) global registration.

In step i) keypoints are extracted using [9] (see also [7] for a review). In general, these keypoints are associated to distinctive parts of the shape such as anatomical parts (i.e., eyes, nose, fingers) or articulated junctions (i.e., for articulated objects). In step ii) keypoint descriptors are compared in order to estimate the overlap among views. The idea is that two views are overlapping if they share a sufficient number of similar keypoints. Being inspired by methods for 2D mosaicing [8], we propose a *global* voting scheme. Instead of computing the keypoint matching between view pairs separately, we enable a keypoint to be compared with keypoints of all the views at the same time. This drastically reduces the computational complexity of the adjacency matrix estimation. In step iii) we propose a multiview optimization method by extending the one described in [12] to work on multiple views. In this way the absolute orientation of the views is estimated by minimizing the global alignment error.

3.1. Keypoint extraction and description

We adopt a *feature*-based approach which is composed of two main phases: i) keypoint extraction, and ii) keypoint description.

3.1.1 Keypoint extraction

Keypoint extraction aims at detecting few and distinctive feature points from the shape. We employ the method proposed in [9], that consists of three main steps: (i) multiscale representation, (ii) 3D saliency measure definition, and (iii) keypoint detection (for a complete explanation of the method see [9]).

The multiscale representation is obtained by applying N Gaussian filters on the mesh, obtaining N multidimensional filtering maps $\{F_i\}, i = 1, \dots, N$. The *neighborhood region* of a vertex \mathbf{v} , over which the filtering is applied, is built by expanding an n -rings search starting from v , and collecting all those vertices displaced within a distance equal to 2.5σ , where σ is the standard deviation of the Gaussian kernel. The Difference-of-Gaussians (DoG) operator is defined as:

$$F_i(\mathbf{v}) = g(\mathbf{v}, \sigma_i) - g(\mathbf{v}, 2\sigma_i) \quad (1)$$

where σ_i is the value of the standard deviation associated to scale i . Six scales of filtering have been fixed, corresponding to standard deviation values $\sigma_i \in \{1\epsilon, 2\epsilon, 3\epsilon, 4\epsilon, 5\epsilon, 6\epsilon\}$, where ϵ amounts to 0.1% of the length of the main diagonal located in the bounding box of the model.

It is worth noting that $F_i(\mathbf{v})$ is a 3D vector which denotes how much the vertex \mathbf{v} has been moved from its original

position after the filtering, and this can be taken as a saliency measure. In order to reduce the displacement vector $F_i(\mathbf{v})$ to a scalar quantity it is projected to the normal $n(\mathbf{v})$ of the vertex \mathbf{v} . In this fashion the *scale* map M_i is obtained as:

$$M_i(\mathbf{v}) = \|n(\mathbf{v}) \cdot (g(\mathbf{v}, \sigma_i) - g(\mathbf{v}, 2\sigma_i))\|. \quad (2)$$

Each map is then normalized by following the approach of [16]: normalize the values of the map to a fixed range; find the global maximum t ; find all the other local maxima and compute their average \hat{t} ; multiply the map by $(t - \hat{t})^2$. The effect of this normalization is to increase the prominence of the highest peaks. A saliency map is obtained by summing the contributions of each scale map.

Finally, salient points are obtained as maxima of the saliency map: a point is detected if it is a local maximum and its value is higher than 30% of the global maximum. An example of such keypoints is shown in Fig. 1.



Figure 1: Two views from the Bunny datasets with the extracted keypoints in red.

3.1.2 Keypoint description

Keypoint description aims at attaching a descriptor to each keypoint that must be: i) distinctive of the point, ii) invariant to rigid transformations, and iii) resilient to nuisances such as clutter, noise, partial views, sampling rate. We use spin-images [17], a well-known surface representation that have been successfully employed in shape matching and object recognition.

For the sake of self-containedness we will briefly summarize the approach.

A 3D point \mathbf{p} with its normal \mathbf{n} defines a local coordinate system (α, β) , where α is the radial distance to the surface normal line and β is the axial distance above the tangent plane.

Given an oriented point (\mathbf{p}, \mathbf{n}) a *spin-map* S_O is defined as the function that projects 3-D points \mathbf{x} to the local coordinate system defined by (\mathbf{p}, \mathbf{n})

$$S_{\mathbf{p}}(\mathbf{x}) \rightarrow (\alpha, \beta) = (\sqrt{\|\mathbf{x} - \mathbf{p}\|^2 - (\mathbf{n} \cdot (\mathbf{x} - \mathbf{p}))^2}, \mathbf{n} \cdot (\mathbf{x} - \mathbf{p})) \quad (3)$$

A *spin-image* for point \mathbf{p} is generated by applying the spin-map to all the points \mathbf{x} of the surface and then accumulating

the results in a discretized (α, β) space. In order to make the descriptor more local and less sensitive to clutter a support distance and a support angle are introduced that limits the contribution to the spin image to those points \mathbf{x} that meets some criteria based on distance from \mathbf{p} and angle between \mathbf{p} and the surface normal of \mathbf{x} .

If surfaces are uniformly sampled, the spin-images of two corresponding points on different instances of the same object will be the “similar” in some sense. In the original paper the authors proposed to use the cosine distance to measure similarity; in this work we are forced to use instead the Euclidean distance, in order to be able to employ a k-d tree for matching.

In the light of the above description, there are three parameters that control the generation of spin-images: bin size, support distance, and support angle. Following [17] the bin size is set to 1.5 times the mesh resolution, and the support angle to 75deg. The support distance have been set to 1/10 of the size of the model, in order to compel the spin-image to act as a local descriptor.

3.2. Pairwise view matching

The objective is to find matching keypoints in pairs of views with some overlap. As the range views are unordered, the first step is to identify in a computationally efficient way pairs of views that potentially share a good number of keypoints, instead of trying to match keypoints between every view pair, as they they are $O(n^2)$ (where n is the number of views). We follow the approach of [8] for 2D image mosaicing. In this *broad phase* we consider only a constant number of descriptors in each view (we used 100, where a typical view contains thousands of keypoints). Then, each keypoint descriptor is matched to its ℓ nearest neighbors in feature space (we use $\ell = 6$). This can be done efficiently by using a k-d tree to find approximate nearest neighbors (we used the ANN library¹). A 2D histogram is then built that record in each bin the number of matches between the corresponding views: we call it the *keypoint co-occurrence matrix* H (see Fig. 2).

Then, in the *narrow phase* every view will be matched only to the m views (we use $m = 8$) that have the greatest values in its row of the keypoint co-occurrence matrix H . Hence, the number of views to match is $O(n)$, being m constant. Precise view-to-view matching follows a nearest neighbor approach as in [21], with rejection of those keypoints for which the ratio of the nearest neighbor distance to the second nearest neighbor distance is greater than a threshold (set to 1.5 in our experiments). These matches are then used to compute the rigid transform that aligns the view pairs using MSAC [28]. Some view matches can be rejected at this stage, if MSAC fails to compute a valid alignment or if the number n_i of remaining inlier matches between two

¹Ann library is available at <http://www.cs.umd.edu/mount/ANN>

views is less than a threshold:

$$n_i > 5.9 + 0.22n_f \quad (4)$$

where n_f is the number of original matches. The derivation of the formula can be found in [8].

Finally, the rigid transform between the two views is refined with LM-ICP [12] (see Sec. 3.3.2) on the whole set of points, whereas before we were considering keypoints only.

After registration, outlier points are singled out using a robust statistics called X84 [10, 13]. Let e_i be the residuals after ICP, the final set of inliers are those points such that

$$|e_i - \text{med}_j e_j| < 3.5\sigma^*, \quad (5)$$

where $\sigma^* = 1.4826 \text{ med}_i |e_i - \text{med}_j e_j|$ is a robust estimator of the scale of the noise.

For partially overlapping views the inliers correspond to the area of overlap, hence we can assign a weight $W(i, j)$ in the range $[0, 1]$ to the pair (view i , view j), corresponding to the fraction of the overlapping points over the total number of points. The $n \times n$ matrix W is called the weighted adjacency matrix (see Fig. 2).

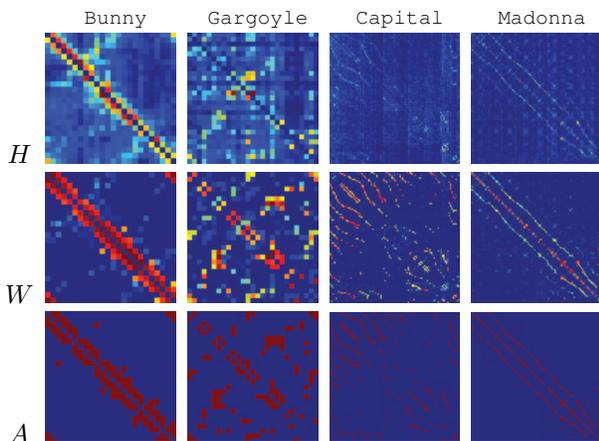


Figure 2: From top to bottom: keypoint co-occurrence matrix H , weighted adjacency matrix W , and adjacency matrix A for the four datasets described in Sec. 4.

3.3. Global registration

There are two stages of global registration: first a global alignment is produced by combining the pairwise rigid transformations found in the previous section; then this alignment is refined with a multiview registration that considers all the views simultaneously.

3.3.1 Graph-based alignment

A weighted graph is constructed, whose vertices are the views and whose (weighted) adjacency matrix is W . Given

a *reference* view chosen arbitrarily, which sets the global reference frame, for each view i , the transformation that aligns it with the reference view r is computed by chaining transformations along the shortest weighted path from i to r . This is equivalent to computing the (weighted) MST with the root in r .

The idea (as in [27, 23]) is that this yields a global alignment of the views with the least accumulation error among the solutions based on chaining pairwise registrations.

3.3.2 Multiview LM-ICP

Some global alignment methods like [14] consider the graph-based alignment to be the optimum, however the solution found with the MST can be further improved by defining a global registration schema which estimates all the absolute orientations at the same time. As an example, Figure 3 shows a detail where the benefit brought in by a global registration can be appreciated.

In this section we shall review the so called Levenberg-Marquardt Iterative Closest Point (henceforth LM-ICP) introduced by [12] for pairwise registration, and then we shall extend it to deal with multiple views.

Let $\{\mathbf{d}_i\}_{i=1}^{N_d}$ be the *data* view and $\{\mathbf{m}_i\}_{i=1}^{N_m}$ be the *model* view: pairwise registration aims at estimating the transformation T which moves the data into alignment with the model. The optimal alignment is given by minimizing the following error function:

$$E(\mathbf{a}) = \sum_{i=1}^{N_d} e_i(\mathbf{a})^2 \quad (6)$$

where

$$e_i(\mathbf{a}) = \min_j \|\mathbf{m}_j - T(\mathbf{a}; \mathbf{d}_i)\| \quad (7)$$

is the closest point distance, and \mathbf{a} is the vector of p parameters that defines a particular T . In our case $T(\mathbf{a}; \cdot)$ is a rigid transformation where the rotation is represented by unit quaternions, hence $p = 7$.

Equation (6) is the error function that gets minimized in ICP [4], which can be regarded as a specialized minimization procedure. Fitzgibbon [12] instead proposed to minimize it with a general-purpose technique such as LM, and has shown that performances in terms of speed and convergence are (surprisingly) better.

The goal of each iteration of LM is to choose an update to the current estimate of \mathbf{a} , that reduces the registration error. To this end the the Jacobian matrix $[J_{i,j}] = [\partial e_i / \partial a_j]$ is needed. These derivatives can be determined efficiently via the pre-computation of the *distance transform* D of the model:

$$D(\mathbf{x}) = \min_j \|\mathbf{m}_j - \mathbf{x}\|, \quad (8)$$

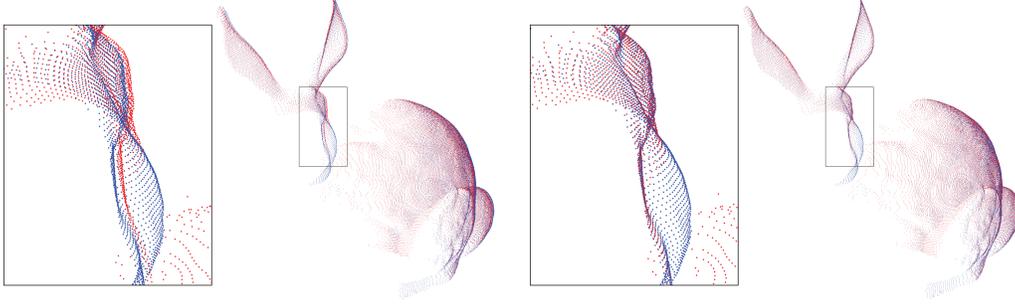


Figure 3: Two views of Bunny (in red and blue) have been aligned with the graph-based alignment approach only (left) and with our Multiview LM-ICP (right). The detail of the face shows a misalignment that gets fixed after Multiview LM-ICP.

where $\mathbf{x} \in X$ being X is a discrete volume grid enclosing the model.

By combining equation (7) with equation (8) one obtains $e_i(\mathbf{a}) = D(T(\mathbf{a}; \mathbf{d}_i))$, and derivatives are readily computed by the chain-rule:

$$J_{i,\cdot} = \partial e_i / \partial \mathbf{a} = \nabla_{\mathbf{x}} D(T(\mathbf{a}; \mathbf{d}_i)) \nabla_{\mathbf{a}}^{\top} T(\mathbf{a}, \mathbf{d}_i). \quad (9)$$

Please note that $\nabla_{\mathbf{a}} T$ is analytical, whereas $\nabla_{\mathbf{x}} D$ can be computed with finite differencing in advance and will remain constant throughout the minimization.

In order to make the method robust one may attenuate the influence of outliers by replacing the square loss function in Eq. 6 with a *robust* loss function [15]. This can be easily incorporated into the distance transform, by defining the ϵ -distance transform:

$$D_{\epsilon}(\mathbf{x}) = \epsilon(\min_j \|\mathbf{m}_j - \mathbf{x}\|), \quad (10)$$

where ϵ is the *square root* of the loss function. In our case we used the Huber loss function with a tuning parameter $k = 1.345\sigma$ that guarantees 95% efficiency:

$$\epsilon^2(x) = \begin{cases} x^2/2, & \text{if } |x| \leq k \\ k|x| - x^2/2 & \text{if } |x| > k \end{cases}$$

Now we are ready to introduce our *Multiview* LM-ICP, by extending the LM-ICP framework to the simultaneous alignment of more than two views. Let V^1, \dots, V^n be the set of views that are to be brought into alignment.

Let us consider the adjacency matrix A such that $A(h, k) = 1$ if view V^h can be registered to view V^k , 0 otherwise. A is obtained by thresholding the weighted adjacency matrix W with $\theta = 0.35$ (see Fig. 2).

Let $\mathbf{a}_1, \dots, \mathbf{a}_n$ be the vectors of parameters that encode the rigid transform that is applied to each view *in the common reference frame*. The alignment error between view V^h and V^k writes:

$$E(\mathbf{a}_h, \mathbf{a}_k) = \sum_{i=1}^{N_h} A(h, k) (D_{\epsilon}^k(T(\mathbf{a}_h \mathbf{a}_k^{-1}, \mathbf{d}_i^h)))^2, \quad (11)$$

where D_{ϵ}^k is the ϵ -distance transform of V^k , $\mathbf{d}_i^h \in V^h$, and $T(\mathbf{a}_h \mathbf{a}_k^{-1}, \cdot) = T(\mathbf{a}_k, \cdot)^{-1} T(\mathbf{a}_h, \cdot)$.

The overall alignment error is defined by summing up the contribution of every pair in the set of overlapping views: $S = \{(h, k) : A(h, k) = 1\}$:

$$\begin{aligned} E(\mathbf{a}_1, \dots, \mathbf{a}_n) &= \sum_{(h,k) \in S} \sum_{i=1}^{N_h} (D_{\epsilon}^k(T(\mathbf{a}_h \mathbf{a}_k^{-1}, \mathbf{d}_i^h)))^2 \\ &= \sum_{(h,k) \in S} \sum_{i=1}^{N_h} (e_{k,h,i}(\mathbf{a}_1, \dots, \mathbf{a}_n))^2 \end{aligned} \quad (12)$$

The derivatives are computed as follows:

$$\begin{aligned} \partial e_{h,k,i} / \partial (\mathbf{a}_1, \dots, \mathbf{a}_n) &= \\ \nabla_{\mathbf{x}} D_{\epsilon}^k(T(\mathbf{a}_h \mathbf{a}_k^{-1}, \mathbf{d}_i^h)) \nabla_{(\mathbf{a}_1, \dots, \mathbf{a}_n)}^{\top} T(\mathbf{a}_h \mathbf{a}_k^{-1}, \mathbf{d}_i^h). \end{aligned} \quad (13)$$

In particular, the Jacobian J is a sparse matrix (see Fig. 4) composed by $q \times n$ blocks $J^{s,r}$, where $q = \#S$. The block $J^{s,r}$ is associated with the s^{th} view pair in S : let (h, k) be such view pair. In formulae, the components of $J^{s,h}$ are defined as:

$$J_{i,j}^{s,h} = \nabla_{\mathbf{x}} D_{\epsilon}^k(T(\mathbf{a}_h \mathbf{a}_k^{-1}, \mathbf{d}_i^h)) \nabla_{[\mathbf{a}_h]_j}^{\top} T(\mathbf{a}_h \mathbf{a}_k^{-1}, \mathbf{d}_i^h). \quad (14)$$

The components of the block $J^{s,k}$ are computed likewise. Please note that in the block row s , only blocks $J^{s,h}$ and $J^{s,k}$ are non-zero.

4. Results

In this section we report results for both synthetic views and real scans acquired by a 3D scanner. The Stanford Bunny model², for which a complete model from laser scanning is given was used as a ground truth, and 24 partial views were synthetically created. Other real data sets (courtesy of IST-CNR) are composed of unordered range views

²available from the Stanford 3D scanning repository (<http://graphics.stanford.edu/data/3Dscanrep/>)

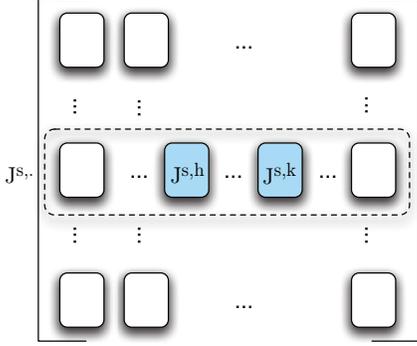


Figure 4: Jacobian matrix. The matrix is composed of $q \times n$ blocks that are associated to view pairs. Given a view pair (h, k) whose index is s , the only non-zero components of the Jacobian are those associated to the unknowns \mathbf{a}_h and \mathbf{a}_k . Indeed, only blocks $J^{s,h}$ and $J^{s,k}$ are non-zero.

of fairly heterogeneous objects: Gargoyle (27 views), Capital (100 views) and Madonna (170 views). Models are shown in Fig.5.

Besides Bunny, all the other datasets come from real-world acquisition, hence noise is intrinsically present and the sampling is not uniform across views (as the scanner moves backward and forward). In the Madonna dataset each scan covers a very small portion of the whole object, with a small overlap between views, thereby making the alignment very challenging.

We evaluate our method in two aspects: the capability of automatically pre-aligning the views, and the accuracy of the final registration after Multiview LM-ICP.

The evaluation of the first aspect is qualitative, and Fig.5 shows that the models have been correctly recovered in cases that ranges from 24 to 196 views. For display purposes only, we show the surface reconstructed with the Poisson method [18].

As for the accuracy, we compared our Multiview LM-ICP with [26], as implemented in Scanalyze[1].

Table 1: Rotation and translation errors for Bunny

Method	rotation [deg]	translation [mm]
Scanalyze	0.086	1.23
Multiview LM-ICP	0.072	0.77

Table 1 reports rotation and translation errors for Bunny for both methods (starting from the graph-based alignment). The figures show that Multiview LM-ICP clearly improves on [26].

This evaluation was possible because for Bunny the ground truth was available. In all the other cases we can only report the registration error (i.e., average closest points

distance), as in Tab. 2. Unfortunately Scanalyze crashed for all the datasets larger than 30 views, hence we have data only for Bunny and Gargoyle.

Table 2: Average registration error [mm].

Dataset	Scanalyze	Multiview LM-ICP
Bunny	0.095	0.093
Gargoyle	0.465	0.459

Please note that, even if the improvement in term of the average registration error appear to be small, this can translate into an appreciable rotation or translation error, as the case of Bunny testifies.

5. Conclusions

The contribution of this paper is twofold. First we proposed a completely automatic method for 3D registration of multiple range views. Although there are methods in the literature that address the same problem, very few work describe fully automatic system which require no manual intervention in any phase and no parameter tuning. Moreover, we are able to deal with much more views than what was proposed in previous work. The second contribution is *Multiview LM-ICP*, the extension of LM-ICP to multiple views in order to minimize a *global* registration error. Although this had been already envisaged in [12], it has never been done before. Results have shown a clear advantage in comparison with a standard method.

Acknowledgments

This paper was supported by the LIRE ‘‘Joint-Project’’ between University of Verona and Gexcel srl. Thanks to M. Corsini (IST-CNR, Pisa) for providing the range scans. A. F. is now with the University of Udine. S. F. is now with 3Dflow srl.

References

- [1] <http://graphics.stanford.edu/software/scanalyze/>. 6
- [2] D. Aiger, N. J. Mitra, and D. Cohen-Or. 4-Points Congruent Sets for Robust Pairwise Surface Registration. In *SIGGRAPH: International Conference on Computer Graphics and Interactive Techniques*, 2008. 2
- [3] F. Bernardini and H. Rushmeier. The 3D model acquisition pipeline. *Computer Graphics Forum*, 21(2):149–172, 2002. 1
- [4] P. Besl and H. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. 1, 4

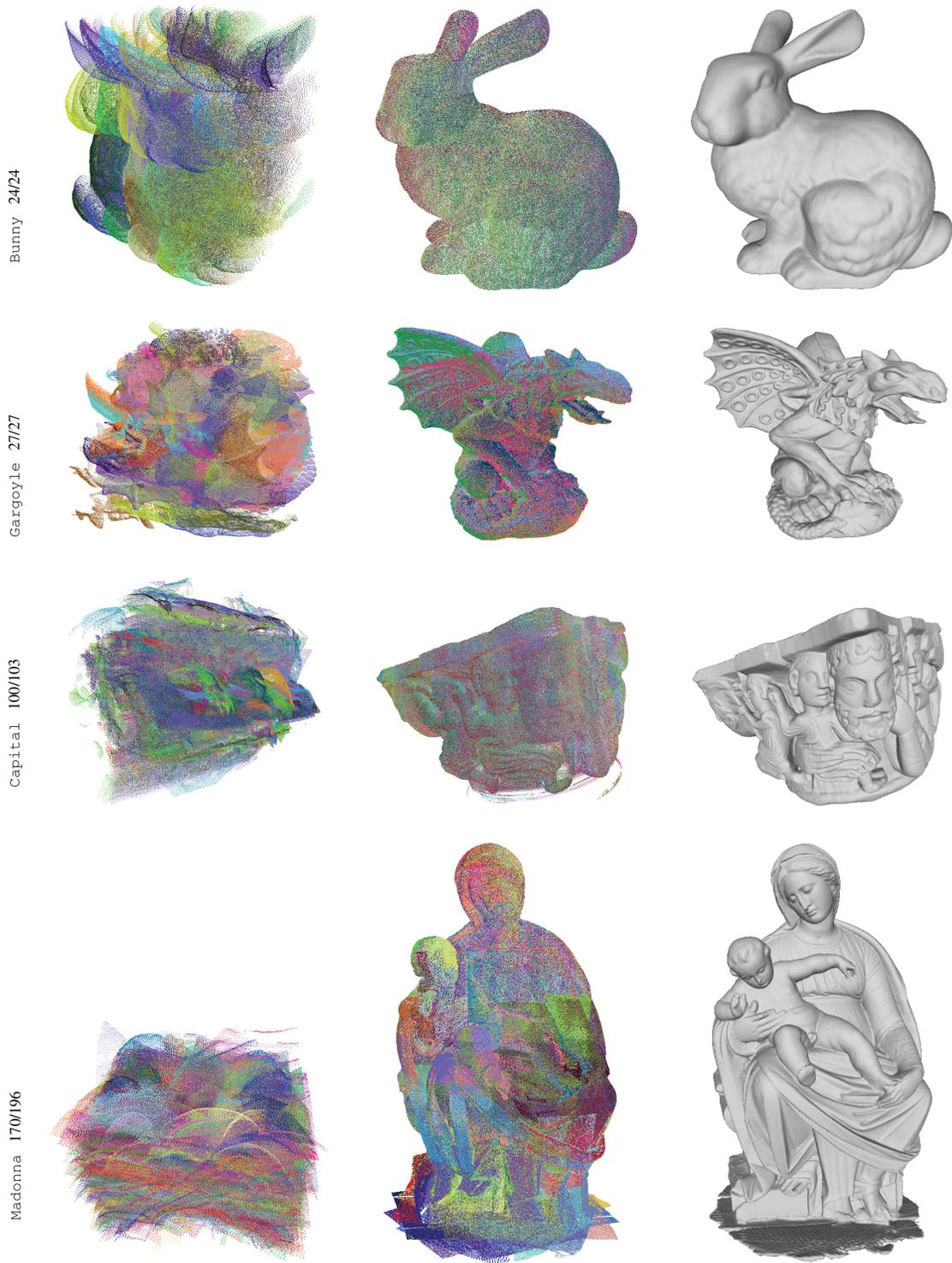


Figure 5: Reconstructed models. Starting pose (left), aligned views (center), and reconstructed surface (right). Points belonging to different views are shown in a different color. On the left, the name of the model and the number of views aligned over the total number.

- [5] F. Bonarrigo and A. Signoroni. An enhanced optimization-on-a-manifold framework for global registration of 3d range data. In *International Symposium on 3D Data Processing, Visualization and Transmission*, 2011. 1, 2
- [6] F. Bonarrigo, A. Signoroni, and R. Leonardi. A robust pipeline for rapid feature-based pre-alignment of dense range scans. In *International Conference on Computer Vision*, 2011. 2
- [7] A. M. Bronstein and et al. SHREC 2010: robust feature detection and description benchmark. In *Proc. Eurographics Workshop on 3D Object Retrieval*, 2010. 2
- [8] M. Brown and D. Lowe. Recognising panoramas. In *International Conference on Computer Vision*, pages 1218–1225, 2003. 2, 3, 4
- [9] U. Castellani, M. Cristani, S. Fantoni, and V. Murino. Sparse points matching by combining 3D mesh saliency with statistical descriptors. *Computer Graphics Forum*, 27:643–652, 2008. 2
- [10] U. Castellani, A. Fusiello, and V. Murino. Registration of multiple acoustic range views for underwater scene reconstruction. *Computer Vision and Image Understanding*, 87(3):78–89, 2002. 1, 4
- [11] S. Cunningham and A. Stoddart. N-view point set registration: A comparison. In *British Machine Vision Conference*, 1999. 1
- [12] A. Fitzgibbon. Robust registration of 2D and 3D point sets. *Image and Vision Computing*, 21(13-14):1145 – 1153, 2003. 1, 2, 4, 6
- [13] F. Hampel, P. Rousseeuw, E. Ronchetti, and W. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. Wiley, 1986. 4
- [14] D. Huber and M. Hebert. Fully automatic registration of multiple 3D data sets. *Image and Vision Computing*, 21(7):637–650, 2003. 1, 2, 4
- [15] P. J. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35(1):73–101, March 1964. 5
- [16] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11), 1998. 3
- [17] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999. 3
- [18] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, pages 61–70, 2006. 6
- [19] S. Khoualed, U. Castellani, and A. Bartoli. Semantic shape context for the registration of multiple partial 3D views. In *British Machine Vision Conference*, 2009. 2
- [20] B. J. King, T. Maliziewicz, C. Steward, and R. J. Radke. Registration of multiple range scans as a location recognition problem: Hypothesis generation, refinement and verification. In *International Conference on 3-D Digital Imaging and Modeling*, 2005. 2
- [21] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 3
- [22] A. Makadia, A. Patterson, and K. Daniilidis. Fully Automatic Registration of 3D Point Clouds. In *International Conference on Computer Vision and Pattern Recognition*, pages 1297–1304, 2006. 2
- [23] R. Marzotto, A. Fusiello, and V. Murino. High resolution video mosaicing with global alignment. In *International Conference on Computer Vision and Pattern Recognition*, 2004. 4
- [24] A. Mian, M. Bennamoun, and R. Owens. Automatic multiview fine registration of range images acquired from unknown view points. In *Proc. International Conference on Modeling, Simulation and Applied Optimisation*, 2005. 2
- [25] A. Petrelli and L. D. Stefano. On the repeatability of the local reference frame for partial shape matching. In *International Conference on Computer Vision*, 2011. 2
- [26] K. Pulli. Multiview registration for large data sets. In *International Conference on 3-D Digital Imaging and Modeling*, pages 160–168, 1999. 1, 2, 6
- [27] H. S. Sawhney, S. Hsu, and R. Kumar. Robust video mosaicing through topology inference and local to global alignment. In *European Conference on Computer Vision*, 1998. 4
- [28] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000. 3
- [29] A. Torsello, E. Rodola, and A. Albarelli. Multiview registration via graph diffusion of dual quaternions. In *International Conference on Computer Vision and Pattern Recognition*, 2011. 2