INCREASING TRACKING ROBUSTNESS FOR LOW-COMPLEXITY REAL-TIME RECONSTRUCTIONS WITH HANDHELD OPTICAL SCANNERS

M. Lombardi[1,2], A. Riccardi[2], A. Signoroni[1,*]

[1] DII, University of Brescia, via Branze, 38, 25123 Brescia, Italy - alberto.signoroni@unibs.it
[2] Open Technologies srl, Via Giacomo Matteotti 161, Rezzato (BS), Italy

Commission II

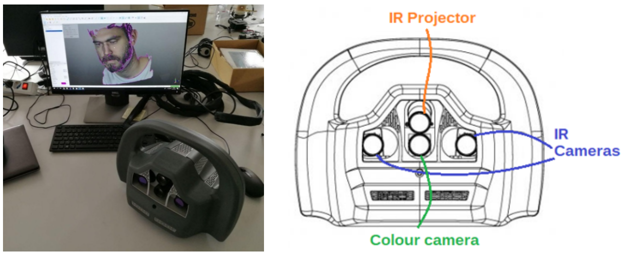
KEY WORDS: Object modelling, 3D surface registration, Range image registration, Real-time scanners

ABSTRACT:

By offering fast and flexible solutions to create 3D models, handheld scanners are currently under the focus of many research activities in various 3D data processing fields. The real-time constraint is still challenging to achieve especially when it comes with concurrent needs, such as level of accuracy in the data acquisition, easiness of recovering from scanning interruptions or loop closure abilities... Among them, object/scene tracking quality is one of the most critical. In this work, we describe two issues that affects its performance, focusing on the robustness of the process. Specifically, we encounter such issues at to two different steps while moving through the working pipeline of a prototype handheld scanner, *i.e.* (1) the data pre-processing before running a pairwise alignment between a frame and the model representation, called key-frame, and (2) the temporal and quality criteria that govern key-frame updates. Our approach simply consists in substituting the use of a rigid (uniform) pattern for sampling, with a random distribution of points. We then implement an adaptive statistical method to select suitable timing steps for key-frames refreshing, comparing this solution with a previous static one based on regular updating rate. We run experiments on a dataset created with our own scanner and we show that the adoption of such alternatives reduce the number of tracking failures, consequently increasing the robustness of the system, improving the quality of the alignments and preserving the real-time behavior of the device.

1. INTRODUCTION

In a classic 3D scanning and modeling workflow, the amount of generated data and computational needs make the reconstruction of the 3D model to take place asynchronously and delayed with respect to acquisitions. In recent years, increasingly powerful and affordable processors come on the market one after the other, de facto enabling 3D reconstructions in a real-time fashion. However, this cannot be considered a self-driving progress, since modern 3D reconstruction applications require solutions characterized by unprecedented degrees of responsiveness, robustness, accuracy and flexibility. This started new research directions (often re-opening more classical ones) and it defines scenarios where diversified combinations of technological and application-related potentialities and limitations can be found. The flexibility introduced by real-time scanners can be exploited adjusting the scanning on-the-fly, to avoid the frustrating situation of finding poorly represented or lacking parts in the model only after post-processing. This can prevent unwanted waste of time or even the worst case of information loss due to some unfeasibility of measure repetition. This is why the interest and also the availability of real-time scanners is increasing in many fields, such as construction industry and civil engineering, robotics, geomatics, design process, reverse engineering, quality assurance and industrial metrology, cultural heritage, entertainment (cinema and video games), medical CAD/CAM (orthotics and dentistry). However, a high degree of flexibility requires fast, accurate and robust acquisitions, which, in turn, calls for high performance of the views alignment processes that work on the stream of acquired partial scans. In this work, we identify and propose solutions for two criticalities that emerge from a low-complexity real-time 3D alignment pipeline by focusing on both pairwise alignment and camera tracking robustness, which are both crucial to guarantee quality reconstructions. The data flow considered here is generated during acquisitions with a commercial handheld 3D scanner (Figure 1).



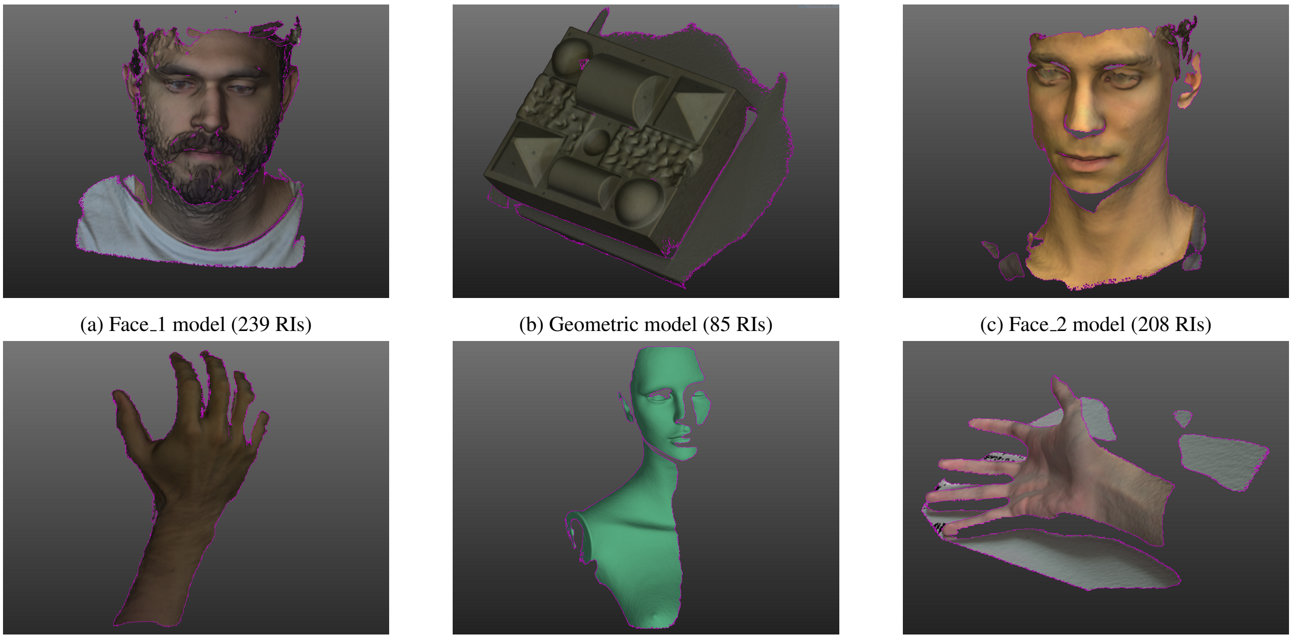
(a) Setup (b) InSight CAD

Figure 1: Acquisition hardware: *InSight* (Open Technologies, Italy) handheld structured-light scanner.

2. RELATED WORK AND CONTRIBUTION

A good alignment of fairly close 3D views (as they typically are in real-time scanning with sufficiently high frame-rate and/or sufficiently moderate scanning speed) can be typically provided by the Iterative Closest Points (ICP) algorithm (Besl, McKay, 1992) and its variants (Chen, Medioni, 1992, Rusinkiewicz, Levoy, 2001). In our specific case, we have pairs of dense sets of 3D points to align, meaning that they are redundant and computationally heavy to process in their entirety. This is why, as an input for ICP, working on a sub-sampled version of the acquired data is a common practice. While the uniform 2D sub-sampling, facilitated by the regular reference grid of the

[*] Corresponding author



(a) Face_1 model (239 RIs)

(b) Geometric model (85 RIs)

(c) Face_2 model (208 RIs)

(d) Forearm model (102 RIs)

(e) Dummy model (161 RIs)

(f) Hand model (145 RIs)

Figure 2: Examples of some 3D reconstructions from our dataset (the total number of Range Images is indicated for each dataset) acquired with the optical scanner of (Figure 1).

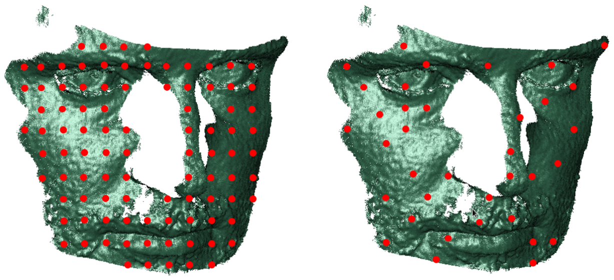
acquired range images, seems to be the most natural way to lighten the data load in a computationally workable way, here we investigate whether and how a random sub-sampling could be adopted instead, by measuring its impact on the alignment performance. Intuitively, this also relates to the tracking robustness of the scanner, which is the second aspect under our attention. Being able to determine rigid pairwise transformations allows recomputing and tracking the location in space of the scanning device. However, alignments can fail and so tracking recovery strategies must be devised. This can be done by adding flexibility in the proposition or computation of alternate targets to prevent or react to possible alignment failures. Moreover policies that look beyond frame-to-frame alignment are also functional to counteract error accumulation drifts. To this purpose, frame-to-model alignment solutions (Izadi et al., 2011, Newcombe et al., 2011a) are of interest, however they do not come without possibly severe additional computational load (Nießner et al., 2013). To cut complexity, instead of using the whole model, new scans can be aligned to *key-frames* (Newcombe et al., 2011b, Steinbrucker et al., 2013), built through ray-casting, which can be used to represent the model under reconstruction in a given temporal window. The decision about when a given key-frame should be updated is critical. Good strategies are those able to guide decisions considering the trade-off between reconstruction accuracy/robustness and computational/memory load. A first trivial but practical solution is to wait a fixed number of new frames before updating the key-frame. In this work, we investigate a smarter approach we derive from the context of dense visual odometry. Inspired by (Kerl et al., 2013a, Kerl et al., 2013b) we propose a statistical method to dynamically update the key-frames according to an entropy ratio criterion.

3. METHODS

We introduce the two focus of our research for the improvement of the tracking robustness of a 3D real-time scanner. We show issues related to each point and describe the solutions we adopted to overcome them.

3.1 Points Sampling for ICP

Many choices can be made to sub-sample the data before aligning via ICP (Rusinkiewicz, Levoy, 2001). Since our range image has been created via stereo triangulation by using two infrared cameras, we have a 2D domain (the reference grid) defined onto the field of view of such cameras, that can be used to retrieve the 3D position of each point. When a new range image is acquired, the first trivial approach is to reduce the number of points by exploiting a regular sub-sampling grid, and then try to align this new subset of points with the current key-frame. The resolution of each camera is of 1280×1024 pixels and we typically work with around 500k-700K valid points per range image. Then, for instance, by using a reducing factor equal to 5 (on both rows and columns), we usually end up working with a smaller set of about 20K-30K valid points. We need to keep the subsampling factor relatively low because of the risk we incur in using a fixed pattern to filter the 2D domain: given the possibly high geometric complexity of the object/scene, we could fall into many sectors mainly containing invalid points (*i.e.* empty space) and consequently fail the tracking. Leveraging on a fixed pattern is a poor choice when we need a good level of generalization, as it is in our case, since we have no clue *a priori* about the shape we are going to acquire. Moreover, this approach is quite easy but it behaves critically also in case of data redundancy, for which a lot of sampled points could be, at best, not helpful or even could cause drifts during the alignment. Therefore it is interesting to investigate a solution that pick the points using different criteria with respect to a predetermined grid. Ideally, a feature based approach should lead to the extraction of salient key-points. However, their detection and selection likely involves a computational load which is not sustainable in our real-time working scenario, so we need an alternative that has to be fast and lightweight as the uniform sampling was. The idea then is to sub-sample again the natively dense data-grid but now extracting the points according to a random distribution. It is reasonable to expect that a randomly covered valid area will need less redundancy with respect to the previous solution, so we can set a smaller number of valid points to use. In Figure 3 we show a qualitative example of both methods applied to a range image inside our dataset. We expect that with points not belonging to a predefined regular grid we could reduce the number of required iterations when the alignment task is more challenging or that we might be more effective in solving some ICP alignment failure cases.



(a) Uniform sub-sampling (b) Random sub-sampling

Figure 3: Qualitative example for two different sub-sampling approaches. In red the hypothetical points extracted from a range image, in (3a) by using a uniform distribution and in (3b) by adopting a random sampling.

3.2 Key-frame selection

In the frame-to-model solution, ICP is performed between the incoming range image and the already acquired model. Similarly to (Newcombe et al., 2011b), the *model* in our pipeline is not the whole set of points obtained from the previous registrations (it would be unfeasible to use it in a real-time scenario) but a representation of it, obtained through ray-casting from a specific position of the camera to the volume. Such representation is then again a range image that we also call *anchor*.

During the scan process, we must update the key-frame many times in order to pick a newer position of the camera, closer to the current one, both in time and in space so that the two range images have a bigger overlapping region and the ICP can successfully run. However, the ray-cast is computationally expensive so that we can not do it constantly during frame acquisitions. The first trivial solution was to update the keyframe at a regular, *a priori* defined, rate. This is a static approach which does not consider how fast the topology of the target under test is changing, both because of its shape and because of how rapidly the scanner is moving. Indeed a slower variation of the topology will require less key-frames and viceversa. In Figure 4 we show how the two approaches (uniform and adaptive) behave differently.

In order to be responsive, we need to define a parameter to describe the difference between the range image and its keyframe. Common strategies employ a threshold for instance on the rigid transformation allowed (Stückler, Behnke, 2012) or on the variance and the total value of the alignment error (Meilland et al., 2011). We found promising results in the work of Kerl *et al.* (Kerl et al., 2013a, Kerl et al., 2013b), coming from the context of visual dense odometry, so we implemented and tested it on our own data.

The method aims to estimate the camera motion g^* between two range images, parametrized by a six-dimensional vector $\boldsymbol{\xi} \in \mathbb{R}^6$ that contains the three components of the linear velocity of the motion (v_1, v_2, v_3), and the three components of its angular velocity $(\omega_1, \omega_2, \omega_3)$. The estimate is done by minimizing the photometric and the geometric residuals, respectively r_I and r_Z, defined as:

$$r_I = I_2\big(\tau(\mathbf{xT})\big) - I_1(\mathbf{x}) \tag{1}$$

$$r_Z = Z_2\big(\tau(\mathbf{xT})\big) - \big[\mathbf{T}\pi^{-1}\big(\mathbf{x}, Z_1(\mathbf{x})\big)\big]_Z \tag{2}$$

Here $I_i(\mathbf{x})$ and $Z_i(\mathbf{x})$ represent the intensity and the depth values at pixel $\mathbf{x}$ for a generic frame i. $\mathbf{T}$ is the rigid body transformation, π^{-1} represents the inverse of the projection function $\pi(\mathbf{p})$ which projects a 3D point $\mathbf{p}$ from homogeneous coordinates space onto the pixel $\mathbf{x}$, while τ is the *warping function* which maps the location of a pixel from one image to the other. Finally, $[\cdot]_Z$ returns the Z component of a point. Assuming both the geometric and photometric consistencies, the residuals differ from zero with a distribution that follows the probabilistic sensor model $p(\boldsymbol{r}_{I,Z}|\boldsymbol{\xi})$. Moreover the noise is assumed to be independent and identically distributed (*i.i.d.*) for all n pixels, then using the Bayes' rule the camera motion $\boldsymbol{\xi}^*$ can be determined by maximizing the probability given the pixel-wise error $p(\boldsymbol{\xi}|\boldsymbol{r}_{I,Z})$:

$$\boldsymbol{\xi}^* = \arg\max_{\boldsymbol{\xi}} \frac{p(\boldsymbol{r}_{I,Z}|\boldsymbol{\xi})p(\boldsymbol{\xi})}{p(\boldsymbol{r}_{I,Z})} \tag{3}$$

Using the *i.i.d.* property, dropping the $\boldsymbol{\xi}$-non dependent elements and converting the maximization to a minimization of the negative log-likelihood, Equation (3) can be written as:

$$\boldsymbol{\xi}^* = \arg\min_{\boldsymbol{\xi}} \Big(-\sum_i^n \log\big(p(\boldsymbol{r}_{(I,Z)_i}|\boldsymbol{\xi})\big) - \log\big(p(\boldsymbol{\xi})\big)\Big) \tag{4}$$

From such equation, by dropping the motion prior $\log\big(p(\boldsymbol{\xi})\big)$ and setting the derivative of the log likelihood to zero we end up with the equation:

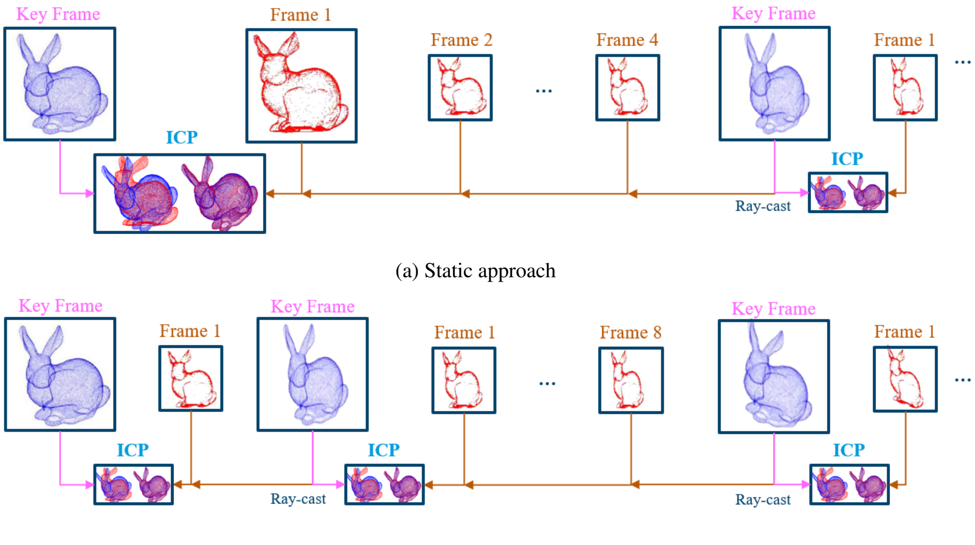
$$\frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{\xi}} \boldsymbol{w}(\boldsymbol{r}_i)\boldsymbol{r}_i = 0 \tag{5}$$

where $\boldsymbol{r}_i$ is an easier to read notation for $\boldsymbol{r}_{(I,Z)_i}$ and $\boldsymbol{w}(\boldsymbol{r}_i)$ ($\boldsymbol{w}_i$ in the follow) is the *weight function* defined as $\boldsymbol{w}(\boldsymbol{r}_i) = \partial \log(p(\boldsymbol{r}_i)/\partial \boldsymbol{r}_i \cdot 1/\boldsymbol{r}_i$.

We intentionally omitted few steps in the computation in order to avoid an overhead of formalization, however one can refer to (Kerl et al., 2013b) for more details on the estimate of $\boldsymbol{\xi}^*$ and on the assumptions made on the distribution of the error. In particular, inspired by (Lange et al., 1989), such distribution results to be better approximated by a t-distribution which in our case is bivariate (because of the bivariate random variable $\boldsymbol{r}_{I,Z}$), *i.e.* $p_t(\mathbf{0}, \boldsymbol{\Sigma}, \nu)$, with 0 mean, scale matrix $\boldsymbol{\Sigma}$ and ν degrees of freedom. In the bivariate case and with all the assumptions made above, the Equation 4 now can be re-written as:

$$\boldsymbol{\xi}^* = \arg\min_{\boldsymbol{\xi}} \Big(-\sum_i^n \boldsymbol{w}_i \boldsymbol{r}_i^T \boldsymbol{\Sigma}^{-1} \boldsymbol{r}_i \Big) \tag{6}$$

This is a non-linear problem in the motion parameter $\boldsymbol{\xi}$, therefore Kerl *et al.* used a first order Taylor expansion to linearize it around the current estimate $\boldsymbol{xi}_k$ and such non-linear least square problem results into a normal equations $\mathbf{A}\Delta\boldsymbol{\xi} = \mathbf{b}$



(a) Static approach

(b) Dynamic approach

Figure 4: Different policies for the key-frame update inside the registration pipeline. The static approach (4a) performs the update every N new range images, while the adaptive method (4b) tries to do it when the new range image is sufficiently *different* from the current key-frame.

where:

$$\mathbf{A} = \sum_{i}^{n} \boldsymbol{w}_i \mathbf{J}_i^T \mathbf{\Sigma}^{-1} \mathbf{J}_i \tag{7}$$

$$\mathbf{b} = -\sum_{i}^{n} \boldsymbol{w}_i \mathbf{J}_i^T \mathbf{\Sigma}^{-1} \mathbf{r}_i \tag{8}$$

Here $\mathbf{J}$ is a 2×6 Jacobian matrix containing the derivatives of $\mathbf{r}_i$ with respect to $\boldsymbol{\xi}$. The normal equations are iteratively solved for increments $\Delta\boldsymbol{\xi}$. Moreover, $\mathbf{A}$ can be seen also as an Hessian matrix, or, equivalently, as the negative of the *observed Fisher information matrix* (Kerl et al., 2013a).

Now it is finally possible to define the metric that express the relation between a new generic frame and the key-frame. In (Kerl et al., 2013a) the use of the ratio between the differential entropy $H(\cdot)$ of the estimate of the motion occurred between the key-frame k and the current frame j, defined as $\boldsymbol{\xi}_{k,k+j}$, and the differential entropy of the estimate of the motion from the same key-frame to the very next frame, *i.e.* $\boldsymbol{\xi}_{k,k+1}$, is proposed. The last assumption is that the estimated parameters $\boldsymbol{\xi}$ are normally distributed with mean $\boldsymbol{\xi}^*$ and covariance $\mathbf{\Sigma}_{\boldsymbol{\xi}}$ such as $\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{\xi}^*, \mathbf{\Sigma}_{\boldsymbol{\xi}})$. According to how the Fisher information matrix $\boldsymbol{A}$ is defined in (7), this gives the lower bound to the variance of $\boldsymbol{\xi}$, *i.e.* $\mathbf{\Sigma}_{\boldsymbol{\xi}} = \mathbf{A}^{-1}$. Moreover, the differential entropy of a generic multivariate normal distribution $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$ with n dimensions is defined as $H(\mathbf{x}) = 0.5 \cdot \big(n\big(1 + \ln(2\pi)\big) + \ln(|\Sigma|)\big)$. Dropping the constant terms, the entropy is proportional to the natural logarithm of the determinant of the covariance matrix, *i.e.* $H(\mathbf{x}) \propto \ln(|\mathbf{\Sigma}|)$, so the information about uncertainty encoded in the covariance matrix is converted into a scalar value. Therefore, we first compute the Hessian matrix $\mathbf{A}$ by iteratively solving the normal equations for increments of $\Delta\boldsymbol{\xi}$ using a standard expectation maximization algorithm for the t-distribution (Liu, Rubin, 1995), then we can compute $H(\boldsymbol{\xi})$. The new metric becomes the *entropy ratio* α:

$$\alpha = \frac{H(\boldsymbol{\xi}_{k,k+j})}{H(\boldsymbol{\xi}_{k,k+1})} \tag{9}$$

The reason behind this metric evaluation is the relation between the error of the estimated trajectory and $H(\boldsymbol{\xi})$, such that when the first new frame is matched against the key-frame, the distance is the best possible for that frame and the estimate is accurate. The accuracy then degrades with the increasing distance between matching frames and so the entropy (which is a *negative* differential entropy) increases, *i.e.* getting closer to zero. So, in absolute value, the ratio α decreases until it reaches a threshold we set, telling that the accuracy is too low to continue with the current key-frame and that an update is recommended. This method is adaptive, as opposite to the uniform update rate shown above and then we expect to reduce the number of updates during the alignment process and even to recover additional range images that could be tried to align with a suboptimal selected key-frame.

4. RESULTS

We run our experiments on a dataset created with the considered optical 3D scanner (few model examples are given in Figure 2) and we evaluate the system performance with and without one or both the adopted solutions, comparing them to the basic pipeline. The tests are performed offline, so that we can tune the parameters on the same data for all the evaluations. Through experience with the dataset we found a configuration that is usually able to generalize quite well to new acquisitions. Among the main parameters to set, the *scaling factor* used for uniform sampling has been set to 5, implying working with around 10k-30k points. Conversely, for our tests, we set the maximum number of randomly sampled points to 500 (thus much less than in the uniform case). Moreover, while the key-frame update when performed uniformly has been done with a rate equal to $N = 5$ (to guarantee decent robustness), in the dynamic case, for the update policy based on entropy ratio, we used a threshold α set to 0.9.

For the evaluation we focused on:

- The total number of correct alignments,
- The average number of iterations occurred during ICP (evaluated on the positive alignments only),

Models	**Total Alignments**		**Avg. Iterations**		**Avg. KF Updates**		**Avg. Distance [mm]**	
	U&S	**R&A**	**U&S**	**R&A**	**U&S**	**R&A**	**U&S**	**R&A**
Dummy	134/161	161/161	3.97	2.27	0.20	0.14	1.05	0.34
Face_1	204/239	218/239	4.30	4.75	0.20	0.05	0.98	0.49
Face_2	119/208	205/208	1.10	1.25	0.20	0.13	0.27	0.31
Forearm	93/102	98/102	3.80	2.23	0.20	0.19	0.66	0.36
Geometric	81/85	85/85	1.77	1.81	0.20	0.20	0.67	0.36
Hand	145/145	145/145	1.07	1.02	0.20	0.04	0.21	0.21

Table 1: Results run on 5 models in our test set. We compare the initial solution (uniform sampling and static key-frame update, classified as *U&S*) with the new solution (random sampling and adaptive key-frame update, namely *R&A*).

Models	**Total Alignments**		**Avg. Iterations**		**Avg. KF Updates**		**Avg. Distance [mm]**	
	U&A	**R&S**	**U&A**	**R&S**	**U&A**	**R&S**	**U&A**	**R&S**
Dummy	145/161	161/161	5.03	2.76	0.07	0.20	0.88	0.37
Face_1	202/239	211/239	5.63	4.38	0.05	0.20	0.72	0.59
Face_2	119/208	200/208	1.18	1.25	0.06	0.20	0.29	0.31
Forearm	93/102	98/102	2.76	2.31	0.10	0.20	0.38	0.41
Geometric	85/85	85/85	1.73	1.78	0.20	0.20	0.37	0.34
Hand	145/145	145/145	1.01	1.09	0.04	0.20	0.22	0.21

Table 2: Results run on 5 models in our test set. We compare the remaining combinations of the methods (uniform sampling with adaptive key-frame update, labeled as *U&A*, and random sampling with static key-frame update, *R&S*).

- The average number of key-frame updates (again computed over the positive alignments only),
- The average of the final distance returned after the ICP pairwise alignment.

Table 1 shows the results obtained on four representative sample models, by comparing the two opposite solutions: the one using the original policies for sampling and updating the key-frames, and the other adopting both the new solutions. We complete the comparison by showing, in Table 2, the results on the same datasets using the other two available mixed combinations of the presented methods.

Overall we see that changing the initial solutions (either only one or both) we always have an increment of total alignments. The *Face_2 model* is the best case: because of redundancy and noise in the data, the uniformly sampled points resulted to be meaningless, so the ICP minimization algorithm diverged when reaching the range image number 120 and the pipeline was not able to recover the tracking afterwards. The problem was instead overcame by selecting less points but randomly and in fact the test shows an increment of 86 new alignments.

The *Dummy* model case shows how using a non-fixed grid to sample the data, combined with a dynamic policy for updating the key-frames, reduces the number of iterations required to perform the alignment and reduces also the number of ICP failures by concurrently improving the alignment accuracy without impairing real-time reconstruction with the same hardware (with few data-dependent variations the complexity remains of the same order of magnitude). Although in the other study cases the average number of iterations is not considerably modified, we are always adding to the evaluation new aligned range images, which resulted to be critical in the first attempt (indeed the alignment failed), so to us is not surprising to see ICP requiring few more iterations in these situations. The timing analysis showed no particular differences between the two versions: both perform a single alignment within 90 ms. The additional computation of the random distribution is then absorbed and does not affect the overall real-time experience.

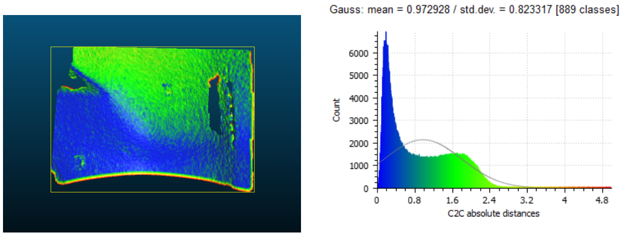
Moreover, regarding the number of key-frame updates, is clear how the previous policy (*i.e.* to update every 5 frames) was sub-optimal: using an adaptive method we were able to align at least the same amount of range images with less than half of the key-frames. This means that we can reduce the amount of ray-casting operations, which cost around 15 ms every time. A significant example is the *Hand* dataset. In this case the acquisition were performed by moving quite slowly the scanner, so the dataset contains a lot of very similar range images. In the end, the new method achieves the same result of total alignments (all the set were successfully aligned), with the same average distance (it is hard to go below the 0.2 mm because of the intrinsic error coming from the acquisition) but with just a fifth of the total updates.

In addition, because of the smarter decision about the timing for the update of the key-frame, each alignment is performed between pairs of range images that shares enough overlapping area. Then together with the random selection of points, the whole new process also produces a better alignment overall, as pointed out by the final average distance that is always smaller then the original method. The only exception happens for the *Face_2 model*, but we point out that its averages are computed on a much bigger pool of correct alignments and so there is a bias to consider for the comparison of such measurements. Indeed if we average on the first 119 range images only also for the new implementation, we get the same result for the average distance, *i.e.* 0.27 mm.

Finally, we add a comparison on how the pairwise alignment performs when the same range image is aligned to two different key-frames: in the former case (Figures 5 and 7) it was updated by using a fixed rate, while the latter (Figures 6 and 8) leveraged on the adaptive solution. These examples highlight the fact that the adaptive update can select a key-frame closer to the current data with a smaller average point-to-point distance and less outliers (as shown by the increased standard deviation on the distance) with respect to the static counterpart.

5. CONCLUSION

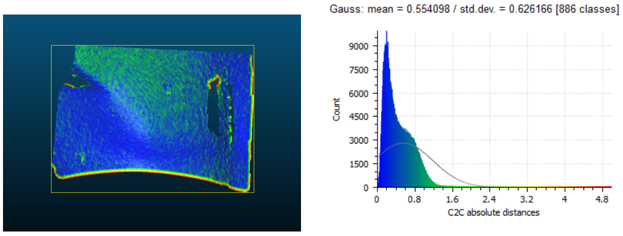
We conclude that the use of random sampling and dynamic updating strategies outperformed former fast solutions while maintaining low computational complexity. In particular, the new sampling method, despite being slower, allowed maintaining the timing performance, concurrently solving critical alignment failure occurrences in the original solution. Moreover, the new key-frame update method resulted to be more flexible and more efficient, since the update is performed only when it



(a) Superposition of aligned frames

(b) Distribution of point-to-point distances

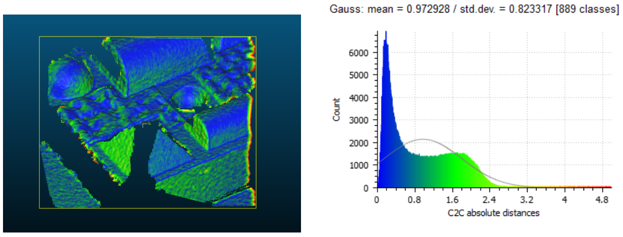
Figure 5: Distance comparison between two range images from *Dummy model*, aligned with the original methods. This and the following figures was produced with *CloudCompare (version 2.9.1)* http://www.cloudcompare.org/



(a) Superposition of aligned frames

(b) Distribution of point-to-point distances

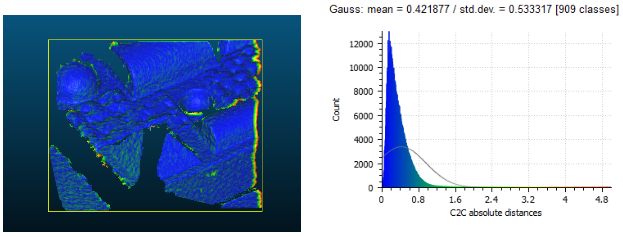
Figure 6: Distance comparison between two range images from *Dummy model*, aligned with the new methods.



(a) Superposition of aligned frames

(b) Distribution of point-to-point distances

Figure 7: Distance comparison between two range images from *Geometric model*, aligned with the original methods.



(a) Superposition of aligned frames

(b) Distribution of point-to-point distances

Figure 8: Distance comparison between two range images from *Geometric model*, aligned with the new methods.

is deemed necessary (which translates into optimized resource consumption) and, since it is forced when there is not enough overlap between the frame and the key-frame, this allowed to solve other alignment failures that we experienced with the original solution.

REFERENCES

Besl, P. J., McKay, N. D., 1992. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2), 239-256.

Chen, Y., Medioni, G., 1992. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3), 145 - 155.

Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., Fitzgibbon, A., 2011. Kinectfusion: Real-time 3D reconstruction and interaction using a moving depth camera. *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, 559–568.

Kerl, C., Sturm, J., Cremers, D., 2013a. Dense visual SLAM for RGB-D cameras. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2100–2106.

Kerl, C., Sturm, J., Cremers, D., 2013b. Robust odometry estimation for RGB-D cameras. *2013 IEEE International Conference on Robotics and Automation*, 3748–3754.

Lange, K. L., Little, R. J. A., Taylor, J. M. G., 1989. Robust Statistical Modeling Using the t Distribution. *Journal of the American Statistical Association*, 84(408), 881-896.

Liu, C., Rubin, D. B., 1995. ML estimation of the t-distribution using EM and its exstension, ECM and ECME. *Statistica Sinica*, 5(1), 19–39.

Meilland, M., Comport, A. I., Rives, P., 2011. Dense visual mapping of large scale environments for real-time localisation. *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4242–4248.

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A., 2011a. Kinectfusion: Real-time dense surface mapping and tracking. *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, 127–136.

Newcombe, R. A., Lovegrove, S. J., Davison, A. J., 2011b. Dtam: Dense tracking and mapping in real-time. *2011 International Conference on Computer Vision*, 2320–2327.

Nießner, M., Zollhöfer, M., Izadi, S., Stamminger, M., 2013. Real-time 3D Reconstruction at Scale Using Voxel Hashing. *ACM Trans. Graph.*, 32(6), 169:1–169:11.

Rusinkiewicz, S., Levoy, M., 2001. Efficient variants of the ICP algorithm. *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, 145–152.

Steinbrucker, F., Kerl, C., Cremers, D., Sturm, J., 2013. Large-scale multi-resolution surface reconstruction from rgb-d sequences. *2013 IEEE International Conference on Computer Vision*, 3264–3271.

Stückler, J., Behnke, S., 2012. Integrating depth and color cues for dense multi-resolution scene mapping using RGB-D cameras. *2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 162–167.