

Handheld scanning with 3D cameras

Enrico Cappelletto, Pietro Zanuttigh, Guido Maria Cortelazzo

Dept. of Information Engineering, University of Padova

Via Gradenigo 6B, 35131 Padova, Italy

enrico.cappelletto, zanuttigh, corte@dei.unipd.it

Abstract—Novel real-time depth acquisition devices, like Microsoft Kinect, allow a very fast and simple acquisition of 3D views. These sensors have been used in many applications but their employment for 3D scanning purposes is a very challenging task due to the limited accuracy and reliability of their data. In this paper we present a 3D reconstruction pipeline explicitly targeted to the Kinect. The proposed scheme aims at obtaining a reliable reconstruction that is not affected by the limiting issues of these cameras and is at the same time simple and fast in order to allow to use the Kinect sensor as an handheld scanner. In order to achieve these targets a novel algorithm for the extraction of salient points exploiting both depth and color data is firstly used. The extracted points are then used within a modified version of the ICP algorithm that exploits both geometry and color distances to precisely align the views produced by the Kinect even when the geometry information is not sufficient to constrain the registration. Experimental results show how the proposed approach is able to produce reliable 3D reconstructions from the Kinect data.

I. INTRODUCTION

The reconstruction of the three-dimensional shape of complex objects has always been a very challenging task. Most approaches rely on the acquisition of a set of different 3D views of the object and then fuse the views into a complete 3D shape representation. The acquisition of the 3D views has traditionally been considered a difficult problem but the recent introduction of real-time 3D acquisition devices, like the Kinect or Time-Of-Flight cameras, has made the acquisition of 3D views much faster and simpler than before. Unfortunately these devices have also several limiting issues (as pointed out in [1], [2]), like high noise level, limited resolution and artifacts in proximity of edges, that must be taken into account in reconstruction algorithms. On the other hand their ability to acquire data at interactive frame-rates makes possible to use a very large number of views. This represents both a strength point, since the views are much closer than the ones used in typical 3D registration pipelines, but also a critical point for computation and memory requirements.

The reconstruction of static 3D scenes with consumer depth cameras like the Kinect is a novel research field. Among the research projects which have investigated this task, Microsoft's KinectFusion [3] is probably the most relevant. In this project each frame acquired by the Kinect is registered in real-time using the ICP algorithm [4] over the complete 3D scene

description reconstructed by using a variation of the volumetric truncated signed distance function (TSDF). The approach of KinectFusion permits an accurate reconstruction, but the large amount of memory needed by the approach limits its application to small scenes, e.g., the implementation in the PCL library (KinFu [5]) is limited to a $3 \times 3 [m]$ area. Kinect Fusion has also been extended in the Kintuous project [6]. In another recent work [7] both geometric and visual features are used for the reconstruction of indoor environments from the Kinect data. Another research project [8] aims at capturing full 3D human body models using 3 Kinects. This approach is able to register the various body parts under non-rigid deformations. Furthermore the first commercial applications exploiting the Kinect for 3D reconstruction are starting to appear, e.g., *Reconstructme* [9] or *Skaneet* [10]. These applications are typically able to capture a full color 3d model of objects, people or rooms, even if the accuracy of the reconstruction is not always satisfying.

Two approaches for the registration of 3D views are described and evaluated in [11]. The first approach is based on RGB images and estimates a sensor pose using image features, while the second uses only geometrical information. The results show that image-based registration method is particularly suitable for scenes with texture, while the object space-based method is able to work on scenes without texture but requires a sufficient amount of geometric information in the scene. This complementary behavior suggests that geometry and texture should be combined in order to provide a highly reliable method, as proposed in this paper.

This work presents a novel 3D reconstruction pipeline to obtain textured 3D models from the data acquired by the Kinect camera or similar devices (e.g., Asus Xtion or Time-Of-Flight cameras). As [3] and other schemes, the proposed approach uses the ICP algorithm but, with respect to the previous approaches, introduces new elements in order to adapt the reconstruction pipeline to the Kinect data and to exploit the color camera information. Firstly an ad-hoc pre-processing step is used in order to reduce the noise level and to remove the most common artifacts typically affecting in the data acquired by the Kinect. Then the relevant points are extracted and finally a modified version of the ICP algorithm is employed. An important novel element, not present in previous works, is the use of color information acquired by the video-camera not only for texture reconstruction but also to improve the extraction of salient points and the geometry reconstruction.

II. GEOMETRY RECONSTRUCTION PIPELINE

The proposed 3D reconstruction pipeline, shown in Fig. 1, is made by 4 basic steps: in the first step the current depth map is extracted from the Kinect and registered together with color data in order to associate to each pixel a 3D colored point. The depth information acquired by the Kinect is also pre-processed to obtain more accurate results. Then salient points that will be used for the registration process are extracted from the point cloud; in the next step the views are aligned together by a modified ICP algorithm; finally post-processing is applied for surface simplification and polishing.

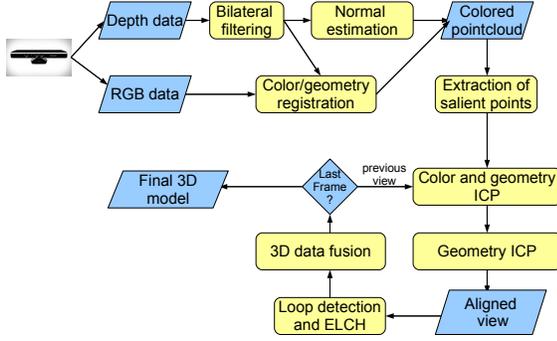


Fig. 1. Architecture of the proposed system

A. Pre-processing of depth information

Before acquisition the system made by the depth and color camera of the Kinect is calibrated by the approach of [12]. The calibration parameters of the IR (depth) and RGB cameras and the relative position between them is used to reproject color data over depth information. Color data is also converted to the CIELAB color space. In this way the acquisition supplies a set of colored 3D points $p_i = (X_{p_i}, Y_{p_i}, Z_{p_i}, L_{p_i}, a_{p_i}, b_{p_i}), i = 1, \dots, N$, where $(X_{p_i}, Y_{p_i}, Z_{p_i})$ are the 3D coordinates and $(L_{p_i}, a_{p_i}, b_{p_i})$ the color components in the CIELAB space.

It is worth noting that, if compared with the data acquired by standard 3D laser scanners, the 3D views acquired by the Kinect are characterized by limited accuracy, high noise level and by the presence of many erroneous depth samples. The proposed pre-processing algorithm is made by 2 basic steps. In the first a bilateral filter [13] is applied to the depth information acquired by the Kinect in order to reduce noise but at the same time preserving the edges. Then we consider a moving window W_{p_i} of size $k \times k$ around each sample p_i of the depth map (the experimental results of Section III were obtained with $k = 3$). We compute the set $S_{p_i} = \{p' \in W_{p_i} \wedge |Z_{p'} - Z_{p_i}| < T_z\}$ of the samples in the window with a depth value similar to the one of the considered point p_i . If the number of samples in S_{p_i} is large enough ($|S_{p_i}| > 0.8|W_{p_i}|$) the point p_i is considered valid, otherwise it is discarded since the point is on a too slanted surface or it is an isolated point. This thresholding is used to filter out unreliable depth values.

Finally the surface normals are estimated [14] for each point (they will play an important role for salient point extraction),

by robust and efficient border- and depth-dependent smoothing.

B. Extraction of salient points

The registration algorithm requires to select a subset of the acquired points to compute the roto-translation matrix between pairs of views. This step is particularly critical in the proposed setup since the data acquired from the Kinect have many unreliable points and furthermore it is not possible to process in real-time too large amounts of samples. In order to obtain an accurate real-time reconstruction it is necessary to extract a small subset of the original points both reliable and meaningful for registration purposes.

To achieve this target we introduce a saliency metric measuring the usefulness of each point for registration purposes. The idea is that the more distinctive points (i.e. the ones either in regions of articulated geometry or high color variance) are the most salient ones.

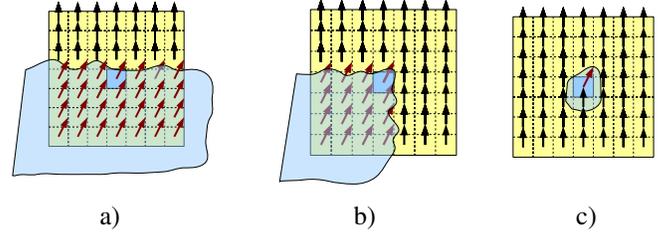


Fig. 2. $|A_{p_i}|$ in different situations: a) On edge samples $|A_{p_i}| \simeq |W_{p_i}|/2$; b) On corner samples $|A_{p_i}| \simeq |W_{p_i}|/4$; c) On isolated samples $|A_{p_i}|$ is very small.

The curvature of the local surface was used as distinctivity measure of geometry (as suggested by [15]). The idea is that points corresponding to high curvature regions can be considered more distinctive since they force tighter bounds on the surface alignment. In particular we compute the normal n_{p_i} to the surface at each point p_i and then we associate to each sample p_i the set

$$A_{p_i} = \{(p' \in W_{p_i}) \wedge (\mathbf{n}_{p'} \cdot \mathbf{n}_{p_i} > T_g)\} \quad (1)$$

of the points for which the surface normals $\mathbf{n}_{p'}$ form an angle smaller than $\arccos(T_g)$ with the normal \mathbf{n}_{p_i} of point p_i . The cardinality of A_{p_i} is therefore inversely proportional to the local curvature of the surface surrounding the selected point. Note that the point p itself is included in the computation in order to ensure that $|A_{p_i}| \geq 1$. Note how a large value of $|A_{p_i}|$ corresponds to samples in flat regions, not very informative for registration purposes. Samples with small $|A_{p_i}|$ are typically associated to edges, corners and high curvature regions. These points represent tighter bounds for the surface alignment. They also have the risk of being less reliable due to the edge artifacts on Kinect data, but notice how edge points have been processed by the bilateral filter of Section II-A. For this reason samples with $|A_{p_i}| > |W_{p_i}|/2 + \sqrt{|W_{p_i}|}$ are excluded from the salient point set and their distinctivity is set to 0. Note how, as shown in Fig. 2 $|A_{p_i}| = |W_{p_i}|/2$ is the typical value for edge

points, the rationale for the threshold value is to keep edge samples or samples with a comparable saliency. On the other side a low value of $|A_{p_i}|$ is usually associated to isolated points typically unreliable or to artifacts due to noise. We decided to exclude points for which $|A_{p_i}| < |W_{p_i}|/4$ (note how a quarter of the window size is the region covered by the considered surface in the case of a typical corner, as shown in Fig.2). The geometric distinctivity measure is therefore computed as the inverse of the cardinality of A_{p_i} , i.e.:

$$D_g(p_i) = \begin{cases} 0 & \text{if } |A_{p_i}| \leq |W_{p_i}|/4 \\ 1/|A_{p_i}| & \text{if } |W_{p_i}|/4 \leq |A_{p_i}| \leq |W_{p_i}|/2 + \sqrt{|W_{p_i}|} \\ 0 & \text{if } |A_{p_i}| \geq |W_{p_i}|/2 + \sqrt{|W_{p_i}|} \end{cases} \quad (2)$$

Fig. 3 shows an example of the computation of geometric distinctivity on a sample 3D view for different values of the threshold T_g .



Fig. 3. Geometric saliency corresponding to different values of T_g . Darker points correspond to larger values of $D_g(p_i)$.

With respect to color information let's recall that a uniform color space, such as CIE Lab, ensures the consistency of the distance measurements between the different color components. Furthermore, since the L component of the CIE Lab color vector (i.e., the luminance), is strongly affected by the viewing direction, specially on reflective surfaces, we decided not to consider it and to use only the a and b components. Similarly to the approach used for geometric information we analyze the window W_{p_i} around point p_i and we look the points with color properties similar as those of p_i , i.e. we compute the set:

$$C_{p_i} = \{(p' \in W_{p_i}) \wedge (\sqrt{(a_{p_i} - a_{p'})^2 + (b_{p_i} - b_{p'})^2} < T_c)\} \quad (3)$$

i.e., the set of the points of W_{p_i} with color components $(a_{p'}, b_{p'})$ similar to those of p_i (also in this case p_i is included in the computation). If it belongs to a uniform color region the cardinality of C_{p_i} will be large. If p_i belongs to regions with a complex texture pattern (more suitable for registration purposes since color data can be used to properly align the surfaces) $|C_{p_i}|$ will assume lower values. As for the case of geometry we threshold the values in order to avoid points in uniform regions or in too noisy areas. The color relevance of the point (an example is shown in Fig. 4) is computed as

$$D_c(p_i) = \begin{cases} 0 & \text{if } |C_{p_i}| \leq |W_{p_i}|/4 \\ 1/|C_{p_i}| & \text{if } |W_{p_i}|/4 \leq |C_{p_i}| \leq |W_{p_i}|/2 + \sqrt{|W_{p_i}|} \\ 0 & \text{if } |C_{p_i}| \geq |W_{p_i}|/2 + \sqrt{|W_{p_i}|} \end{cases} \quad (4)$$

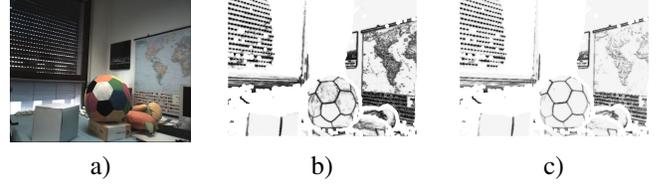


Fig. 4. Color saliency corresponding to different values of T_c . Darker points correspond to larger values of $D_c(p_i)$.

Finally the distinctivity of a point is computed as the maximum of the color and geometric distinctivity:

$$D_p(p_i) = \max(D_g(p_i), D_c(p_i)) \quad (5)$$

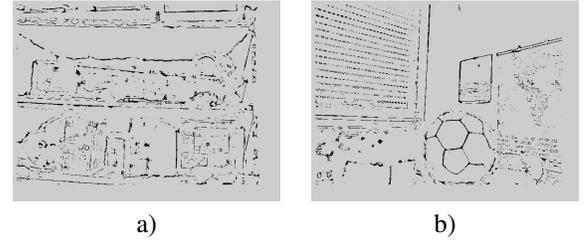


Fig. 5. Maximum of the color and geometric distinctivity $D_p(p_i)$ for two sample scenes. Darker points correspond to larger values of $D_p(p_i)$.

Fig. 5 shows a couple of examples of the computed saliency values on two sample scenes. In order to build the set \mathcal{P}_i of the relevant points of view V_i that will be used for the registration we add a further constraint in order to favor their uniform spatial distribution. Namely the acquired view is divided into quadrants of 40×40 pixels by a regular grid on the Kinect depth map, i.e. the 640×480 depth map of the Kinect is sub-divided into $16 \times 12 = 192$ quadrants. For each quadrant we search for the N_q highest saliency points which will become the salient points for the corresponding regions. If a quadrant contains $N_i < N_q$ salient points (e.g., because it corresponds to a flat and untextured region), the N_i salient points are selected and the missing $N_q - N_i$ points are taken from the other quadrants by increasing their number of salient points (i.e., for the Kinect each quadrant gets $(N_q - N_i)/(192 - 1)$ extra salient points). For the experimental results we used a total of $N_d = 2880$ salient points, i.e. $N_q = 2880/192 = 15$ points for each quadrant (less than 1% of the acquired samples).

C. 3D geometry registration with color-aware ICP

The Kinect sensor is used as an hand-held scanner and is moved around the scene in order to acquire I frames each corresponding to a 3D view $V_i, i = 1, \dots, N$ of the scene. The approach presented in the previous section can be used to extract the set \mathcal{P}_i of the relevant points of view V_i to be used as input for the registration algorithm. The proposed algorithm is based on the Iterative Closest Points (ICP) method [4]. Since the views are very close together the ICP can be applied directly to the acquired data.

The proposed approach is outlined in Algorithm 1. Firstly the relevant points \mathcal{P}_i of view V_i are extracted by the method

of Section II-B. Then a 5-dimensional KD-tree is built where each point has 5 dimensions, the 3 spatial coordinates (x, y, z) and the two color components a and b of the corresponding color value in the CIELAB color space. In order to allow the nearest neighbour search on the 5 dimensional representation that includes two completely different measurement spaces both the geometry and the color are normalized by their standard deviations σ_g and σ_c . The color is then further multiplied by a weighting factor (we experimentally set it to $k_{cg} = 1/7$), i.e.:

$$x' = \frac{x}{\sigma_g} \quad y' = \frac{y}{\sigma_g} \quad z' = \frac{z}{\sigma_g} \quad a' = k_{cg} \frac{a}{\sigma_c} \quad b' = k_{cg} \frac{b}{\sigma_c} \quad (6)$$

A modified version of the ICP algorithm is then used to register the relevant points \mathcal{P}_i over the previously aligned view V_{i-1}^r of the scene. Various extensions and variations of the ICP algorithm have been proposed [16], in this work we used as the distance between corresponding point the distance in the (x', y', z', a', b') space, i.e. the distance depends on both the geometrical distance in the (x, y, z) space and on the difference between the color of the two samples. After the ICP algorithm reach the convergence the set of relevant points \mathcal{P}_i is analyzed and a new set \mathcal{P}'_i is built by removing from \mathcal{P}_i the outliers, i.e. only correspondences with a distance smaller than a threshold T_{icp} in the (x', y', z', a', b') space are preserved.

The remaining set of points \mathcal{P}'_i is used inside a second ICP procedure that performs a final refinement by using geometry information alone. This allows to obtain an accurate geometry alignment and at the same time to avoid common ICP errors on regions or views with limited geometry details where geometry information alone is not sufficient to constrain the registration.

Algorithm 1 Color-aware ICP procedure

- 1: Extract salient points
 - 2: Construct the 5 dimensional KD-tree
 - 3: Run ICP with color and geometry based distances
 - 4: Remove outliers
 - 5: Run refinement ICP with pruned set of salient points and geometry information only.
-

In order to understand how the proposed ICP algorithm with combined color and geometry distance can improve the registration performances lets have a look more in detail to the registration process. If the two point clouds have a large amount of geometry details standard ICP can be applied with the euclidean distance in 3D space as shown in Fig. 6a. Unfortunately in the acquisition of large scenes (e.g. the room or the car in the experimental results) it is quite common to acquire large regions with just a large planar surface and furthermore the low accuracy and high noise level of the Kinect makes difficult to constraint the registration on small objects or surface details as is done with 3D views from laser scanners. In this case, as shown in Fig. 6b, geometry information alone is not able to constraint the registration. In

particular the planar surfaces can “slide” one over the other and the alignment is constrained only in the direction perpendicular to the plane while the alignment in the direction parallel to the plane surface is very unreliable. Fig. 7a shows an example of this problem on a real scene, note how the objects attached to the planar wall surfaces clearly shows the error accumulated in the registration process. In order to avoid the “sliding” effect it is necessary to consider also the color in the computation of the point distances in the registration algorithm. In the proposed algorithm the search for the correspondences in the euclidean 3D space has been modified by adding two further dimensions representing the color of the considered sample (luminance has not been used since it is not very stable across the different views, specially with reflective objects). In this way each point is related to the point on the target view that is close spatially but also have a similar color. Since on planar regions the salient points are typically selected on corner and edges of the texture information this approach allow to precisely align the edges of the objects in the color view and so to constraint the alignment even in the cases where geometry information is very limited or unreliable due to the low accuracy of the depth camera, as shown in Fig. 6c. Fig. 7b shows how by using also color data the scene of 7a can be correctly reconstructed avoiding the “sliding” effect. It is also interesting to notice that the use of salient points only for the new view that is added at each step allows to both drastically reduce the computation time and to improve the registration accuracy.

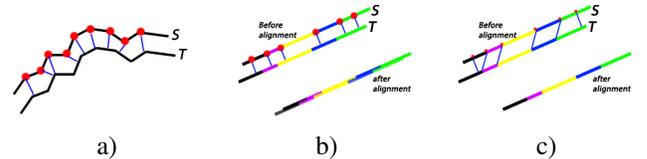


Fig. 6. Alignment of two point clouds: a) alignment of two scenes containing enough geometry information to constraint the registration; b) alignment of two planar surfaces with geometry information alone; c) alignment of two planar surfaces with both color and geometry constraints.

Finally the procedure is then iterated until all the acquired views are processed. During the registration process the algorithm also checks for the presence of loops among the chain of registered views. If a loop is detected the Explicit Loop Closing Heuristic (ELCH) method [17] is used to refine the alignment and avoid the propagation of errors through the registrations of a large amount of views.

D. Fusion of the geometry and color

After registering the new view over the previous acquired data it is necessary to fuse together the two point clouds in order to reduce the number of samples and to produce the final surface. For this task we firstly create a merged point cloud containing all the samples from both V_i and S_{i-1} . Then each point of the set $V_i \cup S_{i-1}$ is analyzed and if another point with a distance smaller than a threshold t_{res} (the threshold depends on the desired final model resolution) is found then a single 3D point is kept. This simple fusion

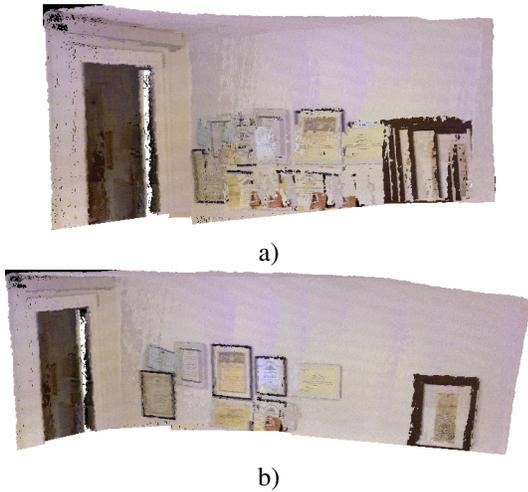


Fig. 7. Example of the reconstruction of a planar scene: a) Reconstruction with geometry-based distance; b) Reconstruction using color and geometry-based distance.

algorithm allows reasonable performances within very limited computation time. Clearly more complex fusion schemes can be used for offline accurate reconstructions. Finally, since the aim of the proposed reconstruction technique is to build colored 3D models, it is also necessary to add color data to the acquired geometry. Each acquired sample has the associated color information, but in the fusion step it is necessary to assign a color value to the samples obtained by merging the points coming from different views. For this task for each 3D sample p_i that is going to be merged we get the normal \mathbf{n}_{p_i} and the viewing direction \mathbf{v}_j corresponding to the view V_j in which point p_i has been acquired. We assign to the merged sample the color data corresponding to the sample for which the normal is better aligned with the viewing direction, i.e., the one that maximizes $|\mathbf{n}_{p_i} \cdot \mathbf{v}_j|$.

III. EXPERIMENTAL RESULTS

In order to evaluate the effectiveness of the proposed approach we acquired several different scenes. We acquired several hundred frames for each considered scene at $10fps$. Each acquired frame is made by a 640×480 depth map and a 1280×1024 color image (bilinear interpolation has been used to assign the color values to the lower resolution depth information). Note that this means that each scene has been acquired in just a couple of minutes, i.e., much faster than any standard 3D scanning techniques.

Fig. 8 shows the reconstruction of a simple office scene acquired by moving the Kinect on a half circle around it. This scene has relevant geometrical features that can bound the registration while texture information is quite limited. The alignment process is robust both for the proposed approach and for other geometry-based approaches. This example together with the one of Fig. 7b shows how the proposed approach is able to exploit the correct clue when only one of the two types of information (color or geometry) has relevant clues.



Fig. 8. Reconstruction of a simple office scene (from 100 frames).

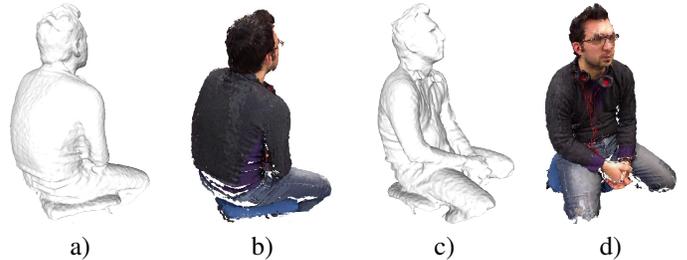


Fig. 9. Reconstruction of a seated person (from 800 frames).

Fig. 9 shows an example relative to the reconstruction of a person. The seated person has been acquired moving the Kinect around him. In this case the corner-less shape can lead to errors in alignment schemes based on geometry only. The proposed approach is able to correctly reconstruct the person shape exploiting both geometry and texture information. A critical point in the texture usage is the management of lightness changing during the acquisition. The scene around the acquisition is not light-controlled so there are reflections and shadows. The choice of ignoring lightness permits to the proposed approach to avoid registration errors and to precisely align the views. In order to avoid blind areas in the final 3D model, a large number of frames is required (800 in this case). In order to avoid error accumulation our approach exploits the on-line loop detection together with the ELCH algorithm and is able to solve this problem keeping in RAM only the last two frames and the computed position of the previously aligned frames, thus keeping the memory usage very low. Computation time requirements are also quite limited, the current un-optimized implementation is able to process around 1 frame per second on a 2,4 Ghz Intel Q6600 processor exploiting a single core. This permits the reconstruction of very long sequences of frames and of large scenes.

Fig. 10 shows our approach applied to a whole colored car (1300 3D views). This scene has several critical issues: the acquisition was done outdoor where sun-light interferes with the IR pattern projected by the Kinect [2]. Furthermore the car's surface is very reflective and reflects the scene all around the car itself. Finally the shape has large regions with very little features. The picture on the left is a view of the back of the car 3D model, while the right picture shows a lateral view of the 3D model with the roof purposely removed in order to show the interior. The latter allows to recognize several element



Fig. 10. Reconstruction of a Chevrolet Aveo (from 1300 frames). a) View from the back; b) side view with the roof purposely removed in order to show the interior.

inside the car like the steering wheel and the seats. To constrain alignments in this scene, the feature extraction approach is fundamental since many frames have poor geometric and color information and only the combined use of the two information sources permits a correct alignment. Note how the geometry is very accurate while texture information is affected by the reflections that are not handled completely by the adopted simple color fusion model (texture creation is in a preliminary stage and is out of the scopes of this paper, further research will be devoted to improve this step).

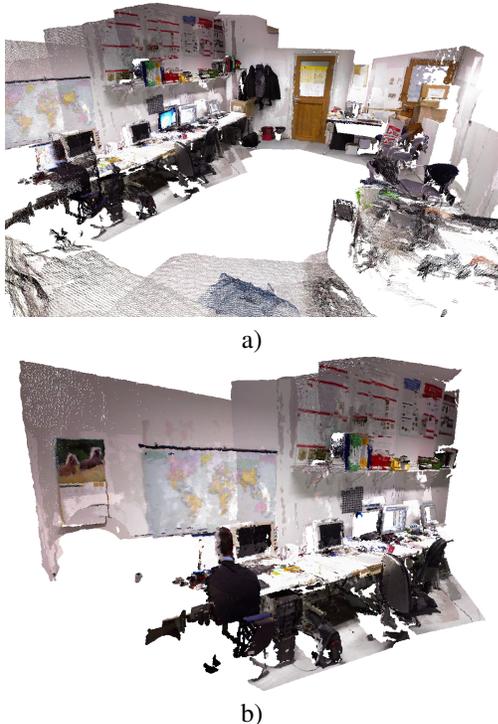


Fig. 11. Reconstruction of the LTTM research lab (from 800 frames).

Fig. 11 shows the reconstruction of a room (800 frames). This scene is particularly challenging since there are large flat surfaces (walls) without relevant geometry information but with texture due to the posted pictures and posters. The proposed approach permits to exploit texture when geometry information is completely lacking and to obtain a complete and accurate 3D model of the scene.

IV. CONCLUSIONS

This paper proposes a novel 3D reconstruction pipeline explicitly targeted to the characteristics of the data acquired by the Kinect exploiting also the side information coming from the associated color camera. The proposed approach demonstrated to be very robust also on challenging scenes thanks to the use of color information in order to assist the 3D reconstruction. Color information has been used both to extract the salient points and to compute of the distances between corresponding points in the ICP algorithm. Furthermore the reliable extraction of salient points has allowed to use a smaller number of points in the registration process and so to speed-up the 3D reconstruction algorithm. Experimental results show how the proposed approach allows to use the Kinect as an handheld scanner for accurate 3D reconstructions. Further research will be devoted to improve the current final fusion step of Section II-D, still in a preliminary development stage for what concerns both geometry and color data. The introduction of a global alignment step able to improve the reconstruction accuracy will also be addressed. The usage of other consumer depth cameras and Time-Of-Flight sensors will also be considered.

REFERENCES

- [1] K. Khoshelham, "Accuracy analysis of kinect depth data," in *Proc. of ISPRS Workshop Laser Scanning*, vol. 38, 2011.
- [2] C. Dal Mutto, P. Zanuttigh, and G. M. Cortelazzo, *Time-of-Flight Cameras and Microsoft Kinect*, ser. SpringerBriefs in Electrical and Computer Engineering. Springer, 2012.
- [3] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *Proc. of IEEE ISMAR*, October 2011.
- [4] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. on PAMI*, vol. 14, no. 2, pp. 239–256, Feb. 1992.
- [5] "Point cloud library (pcl)," <http://pointclouds.org/>.
- [6] T. Whelan, H. Johannsson, M. Kaess, J. Leonard, and J. McDonald, "Robust tracking for real-time dense RGB-D mapping with Kintinuous," MIT, Tech. Rep., Sep 2012.
- [7] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments," *Int. Journal of Robotics Research*, vol. 31, no. 5, pp. 647–663, 2012.
- [8] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan, "Scanning 3d full human bodies using kinects," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 4, pp. 643–650, 2012.
- [9] "Reconstructme," <http://reconstructme.net/>.
- [10] "Skanect," <http://skanect.manct.com/>.
- [11] A. Khosravi, M. Lingenfelder, K. Wenzel, and D. Fritsch, "Co-registration of kinect point clouds based on image and object space observations," in *Proceedings of LC3D Workshop*, 2012.
- [12] "Kinect calibration toolbox," <http://www.ee.oulu.fi/~dherrera/kinect/>.
- [13] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. ICCV*, 1998.
- [14] S. Holzer, R. B. Rusu, M. Dixon, S. Gedikli, and N. Navab, "Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images," in *Proc. of IROS*, 2012, pp. 2684–2689.
- [15] A. Torsello, E. Rodol, and A. Albarelli, "Sampling relevant points for surface registration," in *Proc. of 3DIMPVT 2011*, May 2011.
- [16] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *Proc. of 3-D Digital Imaging and Modeling*, 2001, pp. 145–152.
- [17] J. Sprickerhof, A. Nüchter, K. Lingemann, and J. Hertzberg, "An explicit loop closing technique for 6d slam," in *Proc. of ECOMR*, 2009.