ANDREA BERNARDI

3D RECONSTRUCTION WITH CONSUMER TIME-OF-FLIGHT SENSORS

MASTER'S DEGREE THESIS
IN INGEGNERIA DELL'AUTOMAZIONE

Supervisor: Chiar.mo Prof. Pietro Zanuttigh
Co-supervisor: Dott. Enrico Cappelletto

Università degli Studi di Padova
Facoltà di Ingegneria
Dipartimento di Ingegneria dell'Informazione

9 December 2014
Academic year: 2014-2015

Andrea Bernardi: *3D reconstruction with consumer Time-of-Flight sensors,*
Master's Degree Thesis in Ingegneria dell'Automazione,
© 9 december 2014.

This work has been written in LaTeX using the *ClassicThesis* style by André Miede and the *ArsClassica* package by Lorenzo Pantieri (both available on http://www.ctan.org).

Art thou not, fatal vision, sensible
To feeling as to sight? or art thou but
A dagger of the mind, a false creation,
Proceeding from the heat-oppressed brain?

— William Shakespeare,
Macbeth, Act II, scene I.

To my parents Ginetta and Mario and their endless patience.

ABSTRACT

Simple and fast acquisitions of depth maps can be easily achieved thanks to recent consumer range cameras. These sensors became very popular, but their application to the reconstruction of a 3D scene is a challenging task since they cannot grant the same accuracy level of other devices like laser scanners. This work shows how it is possible to perform reliable reconstructions of 3D scenes captured with such consumer sensors. A reconstruction algorithm that computes the alignment and the fusion of the acquired views into a single final 3D model has been upgraded introducing a more advanced strategy for the fusion task. The entire procedure has been implemented into a software application able to process data acquired form Time-of-Flight cameras such Creative Senz3D and Microsoft Kinect for Xbox One. Finally, a series of experimental measurements has been conducted to evaluate the accuracy of the reconstruction procedure and examine the performances of the sensors.

KEYWORDS

Time-of-Flight sensors, range cameras, 3D reconstruction, Kinect, Senz3D, 3D registration, 3D alignment.

SOMMARIO

Acquisizioni semplici e rapide di mappe di profondità possono essere eseguite facilmente grazie a recenti range camera di tipo consumer. Questi sensori sono divenuti molto popolari, ma il loro utilizzo per la ricostruzione di scene 3D è un'operazione impegnativa dato che non riescono a fornire lo stesso livello di accuratezza di dispositivi come i laser scanner. Questo lavoro mostra come sia possibile realizzare ricostruzioni affidabili di scene 3D ottenute con tali sensori. Un algoritmo di ricostruzione che esegue l'allineamento e la fusione delle viste acquisite in un modello 3D complessivo è stato aggiornato con l'introduzione di una strategia più avanzata per l'operazione di fusione. L'intero processo è stato implementato in un'applicazione software in grado di elaborare i dati acquisiti da telecamere a tempo di volo come il Creative Senz3D e il Microsoft Kinect for Xbox One. Infine, una sessione di misurazioni sperimentali sono state eseguite per valutare l'accuratezza del procedimento di ricostruzione e per analizzare le prestazioni dei sensori.

PAROLE CHIAVE

Sensori a tempo di volo, range camera, ricostruzione 3D, Kinect, Senz3D, registrazione 3D, allineamento 3D.

CONTENTS

1 INTRODUCTION

The reconstruction of the three-dimensional geometry of a scene or the shape of model objects has always been a very fascinating challenge. The most common approach consists in acquiring a set of different 3D views of the scene and merging them together into a single global 3D representation. However, until not too long ago, this possibility was only affordable by research labs or major companies due to the high-cost hardware and instrumentation required.

Recent developments in consumer-grade range sensing technology led to the spread of low-cost sensors which resulted to be both much faster and simpler than their expensive counterparts, maintaining reasonable data accuracy and reliability.

Of course, what has to be intended as "reasonable" varies with respect to the application. A typical consumer device provides limited resolution images with high noise levels, visual artefacts and distortions which cannot be ignored during the reconstruction process. On the other hand, granted enough computational and memory resources, their ability to stream depth data at interactive frame-rates makes the fusion procedure simpler, since more views closer to each other are available. Fortunately, computers able to match, or even exceed, such requirements are quite common nowadays, making possible real-time 3D reconstructions.

A great variety of 3D sensors is currently available, but this work focuses only on a particular subset of passive range sensors which is known as *Time of Flight* cameras. This kind of devices resolves its distance from a scene point, measuring the time-of-flight of a light signal between the camera and the subject for each point of the image.

This work deals with the problem of implementing a 3D reconstruction algorithm and acquiring experimental data with consumer ToF sensors such Creative Senz3D and Microsoft Kinect for Xbox One.

Several research projects covered this topic and Microsoft's KinectFusion is perhaps the most relevant (see [Newcombe et al., 2011]). Its approach exploits the *Iterative Closest Points* method (*ICP*) and a variation of the volumetric *Truncated Signed Distance Function* (*TSDF*) to obtain an accurate reconstruction, but requires too much memory when applied to larger scenes.

Other analoguos works are the Kintinuous project, extension of KinectFusion (see [Whelan et al., 2012]), or human body modeling projects like the one presented by [Tong et al., 2012].

The approach presented by [Cappelletto et al., 2013] has been taken as a starting point. It performs 3D reconstruction via the extraction of *salient points* and the *ICP* registration exploiting both depth and color data. Some variations have been added to the algorithm in order to adapt it to the sensors mentioned above. Moreover the fusion procedure, which merges the aligned views into a single 3D model of the scene, has been upgraded to achieve a good reduction of the number of samples avoiding loss of detail.

Software for the data acquisition has been developed. It allows an easy interaction with the range camera, provides real-time visual output of the captured scene and performs some pre-processing and filtering tasks. The user can hold the range camera with his hand and scan the 3d scene walking around or turning on himself while pointing the sensor toward the scene to acquire.

Finally a series of experimental measurements has been conducted in order to evaluate the accuracy of the reconstruction process and test the behaviour of the ToF sensors mentioned above.

This work is structured as follows:

THE SECOND CHAPTER briefly describes the technology behind Time-of-Flight sensors and their basic mechanics. Several sections have been dedicated to the explanation of the main issues that affects the ToF technology.

THE THIRD CHAPTER examines every step of the core algorithm pipeline for the 3D reconstruction. After a preliminary filtering process, the extraction of salient points is computed in order to allot the alignment and the fusion of the acquired views. As will be explained, the key feature of this approach is the exploitation of both the color and geometry information to achieve the registration.

THE FOURTH CHAPTER introduces an advanced strategy explicitly developed to improve the fusion of the acquired views once they have been aligned by the registration process mentioned above.

THE FIFTH CHAPTER focuses on the consumer ToF sensors Creative Senz3D and Microsoft Kinect for Xbox One used for acquiring the data as well as the software applications implemented to interact with both the range cameras and to compute the 3D reconstruction from the acquired data.

THE SIXTH CHAPTER shows the experimental results obtained after a series of measuring tests. Besides the outcomes of various reconstructions performed from the data acquired by the two devices, a comparison with a reliable ground truth model is presented in order to evaluate the accuracy of the entire process.

2 TIME-OF-FLIGHT CAMERAS

A point-wise ToF sensor estimates its distance from a scene point by emitting radiation waves that travel straight towards the scene for a distance ρ, are then reflected back by the surface and travel back again for the same distance ρ reaching the sensor after a time τ. Thus the measured distance is obtained by the following relationship:

$$\rho = \frac{c\tau}{2} \tag{2.1}$$

where c is the speed of light.

To acquire whole scene surfaces rather than single points, matricial ToF cameras are preferred, since they can measure the scene geometry in a single shot. Such cameras carry a grid of point-wise sensors which estimate their distance from a scene point independently and provide a depth map as output. A depth map is conceptually similar to an ordinary digital picture, but every pixel is associated with the measured distance between the pixel itself and the corresponding scene point instead of a color value.

Most of the commercial products currently available such as Creative Senz3D and Microsoft Kinect for Xbox One (which are going to be treated in detail later on) are implemented following the continuous wave intensity modulation approach, thus a review of basic its operation principles will be presented in the next section.

2.1 AMPLITUDE MODULATED TOF SENSORS

In the model presented, the emitter[1] sends towards the scene an infra-red optical signal $s_e(t)$ of amplitude A_e modulated by a sinusoid of frequency f:

$$s_e(t) = A_e\left[1 + \sin(2\pi f t)\right] \tag{2.2}$$

After hitting a surface, the signal is reflected back to the receiver:

$$s_r(t) = A_r\left[1 + \sin(2\pi f t + \Delta\phi)\right] + B_r \tag{2.3}$$

where A_r is the amplitude of the received signal attenuated by the energy absorption due to the reflection and the free-path propagation, $\Delta\phi = 2\pi f\tau$ is a phase delay representing the non-instantaneous propagation of the signals and B_r takes into account the interfering radiation of the background illumination reaching the receiver.

Recalling equation (2.1), the estimated distance $\hat{\rho}$ can be obtained from a corresponding estimate of the phase delay $\widehat{\Delta\phi}$:

$$\hat{\rho} = \frac{c\tau}{2} = \frac{c\widehat{\Delta\phi}}{4\pi f} \tag{2.4}$$

[1] For the sake of simplicity the emitter will be considered co-positioned with the receiver, although this occurrence never happens. Some devices present multiple emitters distributed around the matrix of sensors simulating a co-axial configuration of the system, but in the most common configuration the sensors grid and the infra-red projector are in two different positions.

which can be inferred from the receiver samples:

$$\widehat{\Delta\phi} = \operatorname{arctan2}\left(s_r(0) - s_r\left(\frac{2}{F_s}\right), s_r\left(\frac{1}{F_s}\right) - s_r\left(\frac{3}{F_s}\right)\right) \tag{2.5}$$

where F_s is the sampling frequency[2].

2.2 NON-IDEALITIES

2.2.1 Phase wrapping

Since $\widehat{\Delta\phi}$ is obtained from $\operatorname{arctan2}(\cdot,\cdot)$, which has the interval $[-\pi, \pi]$ as codomain, only a limited range of distances can be properly calculated. Thanks to the fact that the phase delay can only be positive[3], it is possible to shift the codomain to $[0, 2\pi]$ in order to extend the available range. From equation (2.4) it is clear that $\hat{\rho}$ can assume values within the interval $\left[0, \frac{c}{2f}\right]$, but nothing can be done to estimate distances greater than $\frac{c}{2f}$ because the corresponding higher phase delay values will result periodically wrapped around 2π.

This issue represent a severe drawback for a depth camera since it reduces its operative range down to a few meters, whereas ordinary video cameras commonly adopted by stereo vision systems are not affected by such limitation.

Of course varying the modulation frequency f allows to choose a wider range and as a matter of fact various commercial devices have different maximum ranges. For example the CREATIVE SENZ3D sensor is suited for short ranges up to 1 meters, on the other hand the MICROSOFT KINECT FOR XBOX ONE is able to reach 8 meters of maximum range. In addition more refined technique are usually adopted. For example depth map can be produced by combining multiple images that are captured at different modulation frequency (see section 5.2.1).

2.2.2 Harmonic distortion

Harmonic distortions in the estimated phase delay could be introduced by different causes such as non-idealities of the sinusoids emitted by projectors (usually low-pass filtered squared wave-forms) or the sampling of the received signal that takes a small but finite amount of time. As a consequence a systematic offset component is induced in the estimated distance $\hat{\rho}$ requiring a compensation commonly achievable with a look-up table.

2.2.3 Photon-shot noise

The received signal is affected by photon-shot noise due to the quantized nature of the light and the dark current that flows through the sensors even when no photons are entering the device.

The resultant noise can be described by a Gaussian probability density function with the following standard deviation:

$$\sigma_\rho = \frac{c}{4\pi f\sqrt{2}} \frac{\sqrt{B_r}}{A_r} \tag{2.6}$$

[2] That has to be chosen at least at $F_s = 4f$.

[3] The round trip time τ is obviously a positive value.

It is notable that the precision improves as the signal amplitude A_r increases or the interference intensity B_r decreases. Therefore higher precision data are obtained when measuring points at short distances, with high reflectivity surfaces or when the scene background illumination is low. B_r depends also on A_r, hence raising the signal amplitude would raise B_r too, but the standard deviation σ_ρ would be reduced since the square root dependence of B_r.

Another parameter that affects precision is the modulation frequency f, consequently the possibility of reaching higher ranges by varying f, as stated in section 2.2.1, comes at the cost of altering the measurements precision.

2.2.4 Other noise, saturation and motion blur

Other common noise sources are *thermal noise* components with Gaussian distribution and *quantization* errors. The effects introduced by such random errors can be reduced by averaging the samples over multiple periods, but *saturation* and *motion blur* may arise as side effects depending on the *integration time* length.

When the quantity of photons collected by the sensor exceeds the maximum amount that it can receive (e.g. in the presence of highly reflective surfaces or external infra-red sources), saturation occurs. Of course longer integration times lead to higher number of photons hitting the receiver.

If some scene objects or the range camera itself are moving during the averaging interval, the samples taken at different instants don't relate to the same scene point any more, causing motion blur artefacts.

2.2.5 Flying pixels and multi-path

Since sensor pixels have small but finite dimensions, they cannot be associated to a single point of the scene. They are associated to a finite area, instead. If the area crosses a depth discontinuity (e.g. in the case of an edge between an object and the background), the resulting estimated depth is a value between the minimum assumed by scene points of the closer portion of that area and the maximum assumed by the points of the further region. Pixel associated to such intermediate values are commonly known as *flying pixels*.

Usually, when an optical ray hits a non-specular surface, the *scattering* effect occurs. It consists in reflections along multiple directions instead of a single one. The ray reflected back in the direction of the incident one, is used to estimate the distance, but all the other reflections may hit more scene objects and be reflected back to the sensor altering the measurements. This phenomenon is called *multi-path propagation* and it leads to over-estimation of the scene points distances[4].

[4] More in detail, radial distances are based on the time taken to follow the shortest path between a point and the sensor, but in the presence of multi-path this time length is influenced by longer paths due to other reflections.

3 MAIN RECONSTRUCTION ALGORITHM

The core of the whole 3D reconstruction process presented in this work is based on the scheme proposed by [Cappelletto et al., 2013] which describes an approach to handheld scanning with a Microsoft Kinect for Xbox 360 sensor[1]. Although various modifications have been introduced in order to allow data acquisition with recent ToF cameras (Creative Senz3D and Microsof Kinect for Xbox One) and to improve the fusion process, the main algorithm relies on the same principles. So it comes in handy an oversight of the complete procedure.

3.1 RECONSTRUCTION PIPELINE

The 3D reconstruction algorithm follows a fast and simple pipeline made of four key steps:

PRE-PROCESSING The current depth and color maps are extracted from the ToF camera and registered into a coloured and filtered point cloud.

SALIENT POINTS For the purpose of geometry registration, the extraction of a reduced number of salient points[2] is performed.

GEOMETRY REGISTRATION The point cloud associated to the current view is aligned to the previous ones by a customized ICP[3] algorithm.

FUSION Finally the current point cloud and the preceding ones are fused together to reduce the global amount of points and allow surface generation.

3.2 PRE-PROCESSING

The camera calibration is the first task to be executed before the acquisition process. Then, for each capture, the raw depth map is filtered to reduce noise levels and remove invalid samples and the color information is reprojected over depth data to obtain a coloured point cloud.

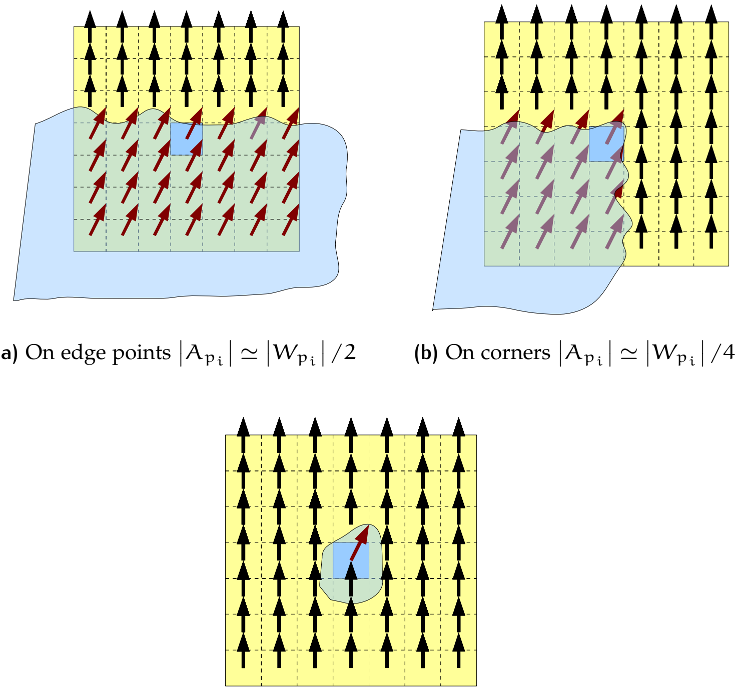
3.2.1 Filtering

The depth map received from the camera is fed to a bilateral filter which operates a good noise reduction preserving sharp edges. Another important step is applying a filter based on the points density. For each sample of the depth map, if the number of his neighbours under a certain threshold

[1] The Microsoft Kinect for Xbox 360 is not a time-of-flight sensor, but a light-coded range camera that project on the scene an infra-red pattern in order to achieve matricial active triangulation and obtain an estimate of the scene depth.

[2] Compared to the whole number of point in the point cloud.

[3] ICP stands for the Iterative Closest Points method.



(a) On edge points $|A_{p_i}| \simeq |W_{p_i}|/2$

(b) On corners $|A_{p_i}| \simeq |W_{p_i}|/4$

(c) On isolated points $|A_{p_i}| \simeq 1$

Figure 1: Cardinality of the set A_{p_i} in different situations.

distance is not high enough, the sample is discarded. This kind of filter prevents unreliable points to be taken into account.

3.2.2 Changing the color space

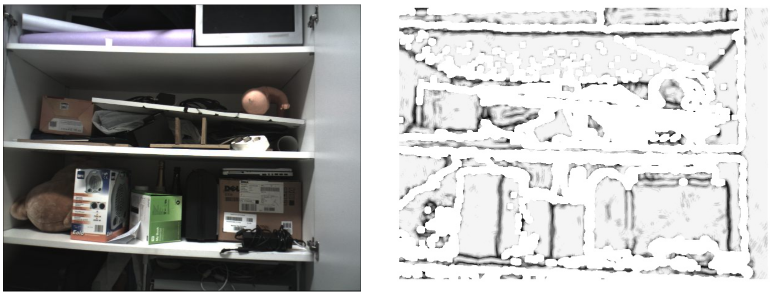
The color map acquired from the camera is usually coded into the RGB format. Since the registration process performed by the ICP algorithm described in section 3.4 includes informations on color distance between points, it is worth it to choose a uniform color space which provides consistency of distance measurements between different color components such the CIELAB color space.

The CIELAB space has three coordinates (L, a, b) where L stands for lightness and a and b for the chromatic components. Therefore, juxtaposing these components to the 3D position coordinates (X, Y, Z), for each view V_j it is possible to identify every point p_i, $i = 1, \ldots, N$ by the sextuple:

$$p_i = \left(X_{p_i}, Y_{p_i}, Z_{p_i}, L_{p_i}, a_{p_i}, b_{p_i}\right). \tag{3.1}$$

3.2.3 Normal estimation

The last step of the preliminary computation is the surface normals estimation for each point in the cloud. They become useful to select valid salient points since regions with high curvature (i.e. wide angles between normals) provide tighter bounds on the views alignment.



(a) Color map

(b) Distinctivity map

Figure 2: Geometric distinctivity example. Darker points correspond to higher saliency.

3.3 SALIENT POINTS

Since the amount of samples acquired at each capture is quite large (usually hundreds of thousands of points), is not possible to compute the alignment process in real-time taking into account every single point. Therefore is essential to select a meaningful subset of the current point cloud and resolve the rototranslation matrix between the previous view and this limited subgroup.

It is obvious to point out that the selection criteria to adopt are crucial to obtain a good registration. Points marked as salient should have peculiarities that can help the alignment process and be as mush reliable as possible at the same time.

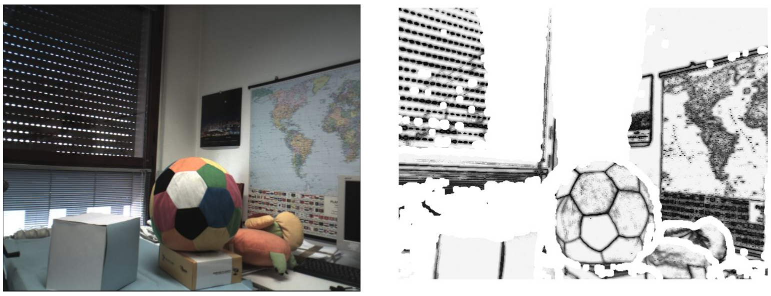
The idea behind the work of [Cappelletto et al., 2013] is to consider both the local surface curvature and the color variance as parameters to compute the distinctivity measure. This approach allows to achieve reasonable good registrations when the scene lack one of these two components, for example a flat wall with posters on it (poor geometry, rich texture).

3.3.1 Geometric distinctivity

Around each point p_i of the surface, the algorithm considers a window W_{p_i} of size $k \times k$. Given the normal n_{p_i} to the surface at each sample p_i, the following set is computed:

$$A_{p_i} = \left\{ \left(p \in W_{p_i}\right) \wedge \left(\mathbf{n}_p \cdot \mathbf{n}_{p_i} > T_g\right) \right\} \tag{3.2}$$

that gathers the points for which the angle between the normals $\mathbf{n}_{p_i}$ and $\mathbf{n}_p$ is smaller than $\arccos\left(T_g\right)$. If the point p_i is located on a high curvature region (such as corners and edges), the cardinality of the set A_{p_i} will be low. On the other side when p_i is on a flat region, nearly all of the other surrounding points will have almost the same normal, implying large cardinality of A_{p_i} (see Figure 1). In order to exclude unreliable points due to noise and artefacts, values of $\left|A_{p_i}\right|$ outside a certain range must be discarded since they typically represent either irrelevant $\left(\left|A_{p_i}\right| \leqslant \left|W_{p_i}\right|/4\right)$ or isolated points $\left(\left|A_{p_i}\right| \geqslant \left|W_{p_i}\right|/2 + \sqrt{\left|W_{p_i}\right|}\right)$.



(a) Color map

(b) Distinctivity map

Figure 3: Color distinctivity example. Darker points correspond to higher saliency.

In conclusion the geometric distinctivity measure is given by[4]:

$$D_g(p_i) = \begin{cases} 0, & \text{if } |A_{p_i}| \leqslant |W_{p_i}|/4 \\ 1/|A_{p_i}|, & \text{if } |W_{p_i}|/4 \leqslant |A_{p_i}| \leqslant |W_{p_i}|/2 + \sqrt{|W_{p_i}|} \\ 0, & \text{if } |A_{p_i}| \geqslant |W_{p_i}|/2 + \sqrt{|W_{p_i}|} \end{cases} \tag{3.3}$$

3.3.2 Color distinctivity

With the purpose of considering color information besides geometry, the color components presented in section 3.2.2 must be included into the distinctivity measure, but the lightness L high dependence on viewing direction and reflectiveness of the surface makes it worthless, so only the a and b components will be considered.

Repeating the way followed for the geometry, given the set of surrounding points similar to p_i with respect to the color attributes:

$$C_{p_i} = \left\{ (p \in W_{p_i}) \wedge \left(\sqrt{(a_{p_i} - a_p)^2 + (b_{p_i} - b_p)^2} < T_c \right) \right\} \tag{3.4}$$

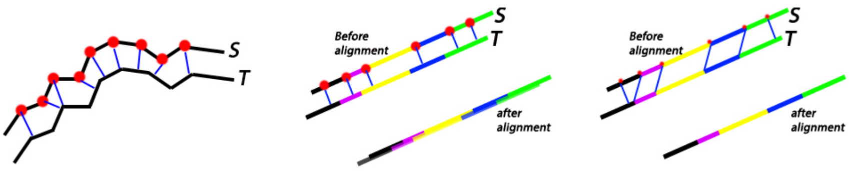
it is easy to see how points located on a region with rich color texture will lead to low cardinality values of the set C_{p_i}, meanwhile when p_i stays on uniformly coloured surfaces, $|C_{p_i}|$ will assume larger values. Obviously the first case $\left(\text{when } |A_{p_i}| \geqslant |W_{p_i}|/2 + \sqrt{|W_{p_i}|}\right)$ is the most desirable since color information can facilitate the registration process, but it is also better not to pick points from too noisy regions $(|C_{p_i}| \leqslant |W_{p_i}|/4)$.

Thus the color distinctivity measure is given by[5]:

$$D_c(p_i) = \begin{cases} 0, & \text{if } |C_{p_i}| \leqslant |W_{p_i}|/4 \\ 1/|C_{p_i}|, & \text{if } |W_{p_i}|/4 \leqslant |C_{p_i}| \leqslant |W_{p_i}|/2 + \sqrt{|W_{p_i}|} \\ 0, & \text{if } |C_{p_i}| \geqslant |W_{p_i}|/2 + \sqrt{|W_{p_i}|} \end{cases} \tag{3.5}$$

[4] Note that there is always at least one element in A_{p_i}, since the point p_i itself is also counted.

[5] Also C_{p_i} is never less than 1.



(a) Enough geometry information (b) Low geometry information (c) Exploiting color information

Figure 4: Alignment of two point clouds. When there is not enough geometry information (i.e on flat surfaces) the alignment can be difficult to achieve. Exploiting color information helps constraining the registration.

3.3.3 The distinctivity measure

Finally the relevance of a point is chosen as the maximum between equation (3.3) and (3.5):

$$D(p_i) = \max(D_g(p_i), D_c(p_i)) \tag{3.6}$$

3.3.4 Salient points distribution

Usually the best registrations are obtained when the salient points are not confined in a single small area, thus for the purpose of ensuring a uniform spatial distribution of the relevant points, each view is divided into 16×12 quadrants and for each of them N_q points with the highest relevance, according to equation (3.6), will be selected. If it is not possible to pick enough points in a certain quadrant, the remaining amount required to reach N_q will be chosen from the remaining quadrants.

3.4 ITERATIVE CLOSEST POINTS METHOD

The *Iterative Closest Points* method presented by [Besl and McKay, 1992] is a well-known algorithm used to align two clouds of points. One of them, called the *target*, is kept fixed, while the other one, usually known as the *source*, is iteratively transformed (by translations and rotations). The idea is to increasingly minimize the distance from the source to the target point cloud.

In the cases presented here, the ICP method can be applied directly to the acquired point clouds[6] without any preliminary manual alignment, since the views captured by the range cameras are very close to each other. Moreover the use of salient points for the source view that is aligned at each step grants a remarkable reduction of the computation time, preserving the accuracy at the same time.

As mentioned earlier, an important improvement to the ICP algorithm is to perform the registration using the distance in the 5-dimensional space (x, y, z, a, b) instead of the Euclidean 3D space alone.

The procedure followed is made of four basic steps:

A. construction of a 5-dimensional KD-tree to allow the nearest neighbour search;

[6] After the pre-processing stage described in section 3.2.

B. running of modified ICP with distances based on both color and geometry;

C. removing of outliers;

D. final refinement using ICP based on geometry only.

For each view V_j, after computing the set P_j of the most salient points, the cloud is organized into a 5-dimensional *KD-tree* where each node has three normalized spatial coordinates and two normalized color components (x', y', z', a', b') showed in equation (3.7). The normalization is required to merge two different measurement spaces and is done dividing each component by the standard deviations σ_g and σ_c and scaling the color ones by the weighting factor k_{cg} which balances the importance given to geometry with respect to color relevance.

$$x' = \frac{x}{\sigma_g}, \quad y' = \frac{y}{\sigma_g}, \quad z' = \frac{z}{\sigma_g}, \quad a' = k_{cg}\frac{a}{\sigma_c}, \quad b' = k_{cg}\frac{b}{\sigma_c} \tag{3.7}$$

The use of such data structure simplify the nearest neighbour search needed to register the relevant points P_i over the previously aligned view V_{j-1}^r which is done running the ICP (based on distances in the 5-dimensional space mentioned above) until it reaches the convergence.

After that, some of the salient points may be far away from all the points of the target view[7]. Such outliers are then removed from the set P_j and a new alignment is performed to refine the registration; only the geometry information is considered this time.

These operations are iterated until all the captured views are registered. The fact that both geometry and color are taken into account during the alignment grants better results in avoiding any *sliding-effect* that may occur when the scene contains wide planar regions.

3.4.1 Loop detection

Registering the current view over the previously aligned ones can lead to errors propagation, especially when the captures are made with noisy consumer devices.

To limit this unwelcome effect, the presence of loops inside the sequence of registered views is constantly checked at every iteration. When a loop is encountered, the registration is refined using the *Explicit Loop Closing Heuristic* method by [Sprickerhof et al., 2009]. Therefore errors accumulations is avoided even when the number of acquired views is high.

Note that to achieve a good reconstruction, the way users acquire the scene becomes of great importance. Closing loops when using range cameras as handheld scanners it is an easy operation. For example one can just go around an object to acquire it completely or turn around pivoting on his feet to capture a room.

3.5 FUSION

Once the current view has been registered over the previous capture, it is necessary to merge them together to obtain a single 3D model of the scene.

[7] For example they may be part of a portion of the seen captured in the current view, but not in the past ones.

Since a straightforward composition of the point clouds would become unnecessarily cumbersome, a simple downsampling procedure based on the points distance is run. When this distance is under a certain threshold, the points are combined into a single point.

3.5.1 Color of the merged points

Two points of different clouds, even if they are very close spatially, may have very different color components since they are particularly affected by the viewing direction (e.g. in case of reflections). Averaging the color components of the samples to be merged is not a good solution because the result would turn out too blurry.

The solution can be found exploiting the fact that samples captured form a viewing direction aligned with the normal of the surface they lie on are less affected by reflections. The color components assigned to the merged sample will be those corresponding to the point with the higher dot product $\left|\mathbf{n}_{p_i} \cdot \mathbf{v}_j\right|$ of its surface normal with the viewing direction $\mathbf{v}_j$ of the camera (when that particular sample was captured).

4 IMPROVEMENTS TO THE ALGORITHM

The algorithm described in chapter 3 provides reliable reconstructions, but its fusion step follows a basic approach and must be improved adopting a more advanced strategy which better exploits the geometry of the scene.

When a registered view is fused together with the previous one, many points of the former will be placed near other points of the latter but not necessarily in the same exact positions. On the contrary, most of the points will be in a slightly different place with respect to their counterparts. This may be due to non-perfect registration and noisy data (see Figure 5).

How to deal with such situation? Keeping every samples, in order to prevent loss of geometry information, leads to clouds of high size. Reducing the points amount by downsampling could be a reasonable thing to do, but usually the points to be merged together are chosen by evaluating their distance only. This approach allows the possibility to adjust the size of the final point cloud by setting the threshold on the merging distance, but does not distinguish between adjacent points due to the alignment process and samples which are close to each other because they belong to a high-detail region.

The idea adopted in this work suggests to perform a selective reduction of the samples by comparing their reliability based on the local geometry of the scene and the viewing direction.

For each point p_i of the current (aligned) view V_j^r, a properly shaped 3D *hull* H_{p_i} is built around it and every point of the target cloud V_{j-1}^r that lies inside the hull is collected to form the set:

$$M_{p_i} = \left\{ \left(p \in V_{j-1}\right) \wedge \left(p \in H_{p_i}\right) \right\}. \tag{4.1}$$

Every point in M_{p_i} (including p_i) is then checked and kept or discarded depending upon their reliability which is based on the surface normal and the viewing direction.

4.1 CLOSE POINTS REDUCTION

As described in section 2.2, faulty samples are often associated with high curvature or slanted surfaces and depth discontinuities. Most of the times a point p_i in such situations shows a surface normal $\mathbf{n}_{p_i}$ which makes a wide angle[1] with the viewing direction $\mathbf{v}_j$ of the camera. On the contrary samples taken on flat surfaces, perpendicular to the optical axis, usually display modest or none distortions. Moreover all the noise affecting the distance measurements adds a disturbance component along the optical ray connecting the camera with the point itself.

These considerations lead to two conclusions. First, the normal alignment with the viewing direction could be reasonably used to gauge the samples accuracy. Therefore selecting the point corresponding to the better alignment and discarding the others during the merging operation would

[1] Up to $\pi/2$, since what does actually matter is the dot product $\left|\mathbf{n}_{p_i} \cdot \mathbf{v}_j\right|$.

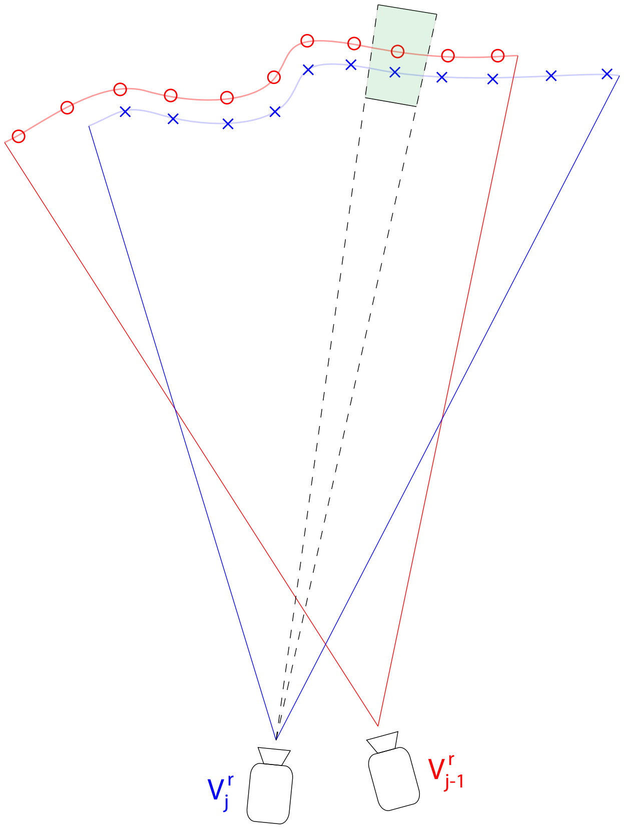


Figure 5: This picture shows how corresponding points are "cloned" during the fusion process. The red circles represent the samples of the previous view V_{j-1}^r, while the blue crosses are the points of the current one V_j^r.

decrease the total amount of points preserving their reliability. Second, the points collected to form the set M_{p_i} should be chosen along the optical ray since the erroneous samples due to the noise lie along that direction. The latter suggestion gives a hint on the shape that the hull H_{p_i} should have to correctly enclose the corresponding points of different views.

4.2 MERGING HULL

In accordance with the reasoning explained in section 4.1, the scheme actually adopted in this work to build the hull H_{p_i} is the following.

The hull has the shape of a frustum of pyramid with a square base and the axis corresponding to the optical ray. It is centred on the point p_i of the current view and both its height and base side length are proportional to the local mean minimum distance of the points of the current cloud (see Figure 6).

In this way the hull is narrow enough to exclude points of the previous view V_{j-1}^r which are unrelated to p_i and must not be merged in order to preserve the geometric integrity of the scene. At the same time is adequately wide to cover every "clone" of the point p_i along the optical ray.

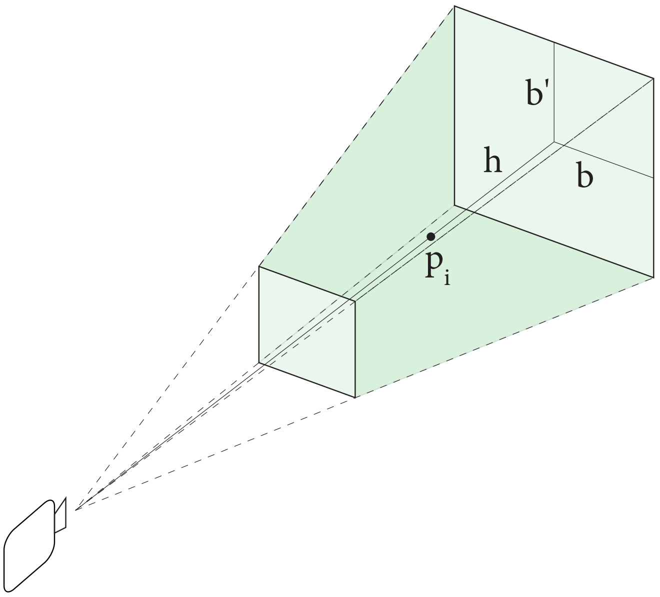


Figure 6: Scheme of the merging hull H_{p_i} built around the point p_i of the aligned view V_j^r.

More in detail, for each point $p_i \in V_j^r$ the mean minimum distance $d_m(p_i)$ between two points of the same view V_j^r in a neighbourhood of p_i is computed. Then the length $\bar{h}$ of the line segment h connecting the point p_i and the base of the frustum is computed:

$$\bar{h} = k_h d_m(p_i) \tag{4.2}$$

where the factor k_h is a parameter that allows to customize the length of the hull. The lower base of the frustum is given by the plane orthogonal to the optical ray that is as distant from p_i as $\bar{h}$ and the line segments b and b' which length is:

$$\bar{b} = \bar{b}' = k_b d_m(p_i) \tag{4.3}$$

where the factor k_b is a parameter for the setting of the width of the hull. The upper base and the other faces of the frustum are computed by the intersection of the plane parallel to the lower base (and opposed with respect to p_i) and the optical rays through the lower base vertices.

Once the hull H_{p_i} has been outlined, checking which point in inside it is an easy task. At this point the computation of the set (4.1) is completed.

4.3 FUSION AND COLOR

During the fusion operation, the dot product $N(p_i) = |\mathbf{n}_{p_i} \cdot \mathbf{v}_j|$ is compared to $N(p) = |\mathbf{n}_p \cdot \mathbf{v}_j|$ for each $p \in M_{p_i}$. Every point p which verifies the relation:

$$N(p) > N(p_i) \tag{4.4}$$

is stored, while the others are deleted. In this way the overall number of points in the cloud is kept moderate.

If there is at least one point p for which (4.4) is true, then there is no need to add p_i to the point cloud since it is less reliable than the point that has already been stored. On the other hand, if M_{p_i} is empty or every p has been discarded, the point p_i is added to the merged cloud because it increases the geometric detail or the general accuracy respectively.

Note that the approach that chooses the point with the highest absolute value of the dot product above, gives also the best color in the sense explained in section 3.5.1.

4.4 STATISTICAL OUTLIERS REMOVAL

The implementation of the algorithms described in this work exploits many tools provided by the open-source Point Cloud Library. It contains various filtering functions that are very useful for processing data with the purpose of 3D reconstruction.

As a matter of fact the *statistical outliers removal* method turned out to be a good way to reduce the amount of flying pixels mentioned in section 2.2.5. Therefore has been inserted as an additional pre-processing step before the salient point extraction in the reconstruction pipeline outlined in section 3.1 and as a final refinement step after the fusion operation since grants better results than the straightforward downsampling of section 3.5.

The removal algorithm presented by [Rusu and Cousins, 2011], compute the local statistical filtering of a given cloud selecting the points which have a distance from the mean distance shorter than a certain factor of their standard deviation. This ensures that high-detail regions are not decimated by an indiscriminate downsampling or low density areas (i.e. background walls) are not excessively eroded, while flying-pixels and noisy samples are cleared away.

5 HARDWARE AND SOFTWARE DESCRIPTION

The 3D reconstruction algorithm presented in this work has been implemented and tested with two consumer Time of Flight sensors. The entire process has been divided into two separate tasks in order to handle the different cameras easily[1].

The first one covers the acquisition process and controls the connection to the device, the data capture, the pre-processing and generates a colored point cloud for each view. Two versions of this software have been implemented, one for the Creative Senz3D and another one for the Microsoft Kinect for Xbox One.

The second task handles the actual 3D reconstruction. It takes the point clouds as input, proceeds with the registration and fusion process and returns a single point cloud representing the reconstructed 3D model of the acquired scene.

The devices tested share the same working principles described in chapter 2, but differ in various aspects. For example they have distinct working ranges, data reliability and physical dimensions. This chapter will examine these elements[2] and the specific developed software as well.

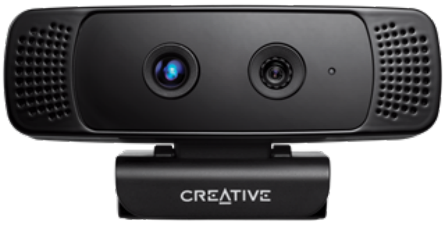


Figure 7: The Creative Senz3D range camera.

5.1 CREATIVE SENZ 3D

The Creative Senz3D is a short range ToF camera based on the DepthSense 325 by SoftKinetic. It a low cost and compact device suitable for being used as an handheld scanner while acquiring a scene.

It produces three different bitmaps as output: a color map, a depth map and an intensity (or confidence) map.

[1] This separation has also proved useful for debugging purposes, since it allows the creation and the test of numerous datasets

[2] The two commercial devices here described, feature many interesting developing tools for the human-machine interaction like body tracking, gesture and voice recognition. Since this work focuses on 3D reconstruction, though, it won't treat any of these functions.

5.1.1 Technical characteristics

Here is a list of some of the technical characteristics featured by the Senz3D sensor available at [Creative website] and SoftKinetic website.

DEPTH MAP RESOLUTION of 320 × 240 pixels.

COLOR MAP RESOLUTION of 1280 × 720 pixels.

FRAME RATE of 30 fps for the color and 60 for the depth frames.

FIELD OF VIEW 74 degrees horizontal by 58 degrees vertical.

DEPTH NOISE less then 1.4 cm at 1 meter (50% reflectivity).

OPERATING RANGE from 15 cm to 1 meter from the camera.

LIGHTING INDEPENDENCE possibility to operate with the usual artificial indoor light or without any external light at all[3].

In addition the sensor is powered via the USB 2.0 connector and an external AC adapter is not needed.

5.1.2 Device limitations

The short operating range is useful when the acquisition of objects very close to the camera is requested, but it is inappropriate to capture big scenes (i.e., rooms or large objects).

The low power requirements of this camera makes its acquisitions less reliable when dealing with black objects. They are not captured well and they are affected by heavy distortions and artefacts.

Another issue is the fact that planes (i.e. the walls of a room or the side of a box) are not rarely altered by noise and distortions that prevents them to appear flat. This happens especially in proximity of concave geometries and is a well known problem, common to many ToF cameras, due to the multi-path propagation described in section 2.2.5.

5.1.3 Acquisition application

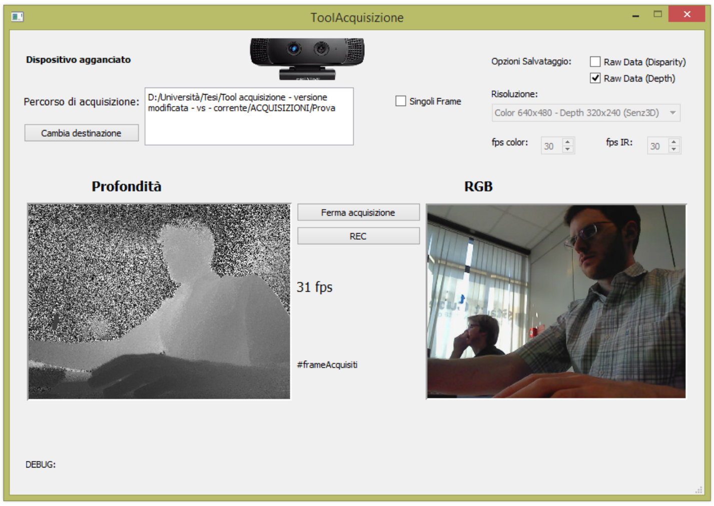
In order to acquire data from the device, an application created by Enrico Cappelletto has been upgraded and adapted to the Creative Senz3D. Originally, the program was able to establish a connection with the Microsoft Kinect for Xbox 360 and the Asus Xtion PRO LIVE sensors, with the purpose to acquire and record data from them.

It has been written in C++ using functions and tools provided by the Qt framework for the design of the graphical user interface and the threads handling. It relies upon the OpenNI framework and the Intel Perceptual Computing SDK to interact with each device.

The application is able to automatically detect which device has been plugged in and, after setting up the connection, is ready to acquire the data. Both the color and depth streams are showed in real time and can be recorded[4] at different frame rates and resolutions.

[3] Differently from the Microsoft Kinect for Xbox One, this camera cannot operate outdoor in the presence of the sunlight.

[4] There is ,also, the possibility of saving a single frame at a time. It turns out useful for debugging and testing purpose.



ToolAcquisizione

Dispositivo agganciato

Percorso di acquisizione: D:/Università/Tesi/Tool acquisizione - versione modificata - vs - corrente/ACQUISIZIONI/Prova

Opzioni Salvataggio: □ Raw Data (Disparity) ☑ Raw Data (Depth)

□ Singoli Frame

Risoluzione: Color 640x480 - Depth 320x240 (Senz3D)

fps color: 30

fps IR: 30

31 fps

#frameAcquisiti

DEBUG:

Figure 8: Screenshot of the acquisition application for the Creative Senz3D range camera.

Since the projection of the color map over the depth data to build a colored point cloud is achieved exploiting the Intel Perceptual Computing SDK, this operation, as well as the filtering process, is done by the acquisition program rather then the reconstruction application that will be described in section 5.3. Therefore the following filters are implemented:

- median filter, provides fast noise removal (especially outliers);
- bilateral filter, ensures edge preserving noise reduction;
- density filter, suited for flying-pixel elimination;

A threshold on the intensity[5] of the received signal establishes which pixels of the depth map are believed to be enough confident. The reliable ones are kept and the others are discarded as well as the samples outside a certain interval of distances.

Figure 9: The Kinect for Xbox One range camera.

[5] Provided by the confidence map generated by the sensor.

5.2 MICROSOFT KINECT FOR XBOX ONE

The Kinect for Xbox One is the second-generation motion sensor created by Microsoft for the video game console market. Nevertheless, like its predecessor Kinect for Xbox 360, it has been instantly used for countless non-gaming related applications. Like the previous sensors, it produces three different bitmaps as output: a color map, a depth map and an intensity map.

5.2.1 Technical characteristics

Here is a list of some of the technical characteristics featured by the Kinect for Xbox One sensor disclosed in the article by [Sell and O'Connor, 2014].

DEPTH MAP RESOLUTION of 512 × 424 pixels.

COLOR MAP RESOLUTION of 1920 × 1080 pixels.

FRAME RATE of 30 fps both for color and depth frames.

FIELD OF VIEW 70 degrees horizontal by 60 degrees vertical.

DEPTH RESOLUTION within 1 percent of distance.

OPERATING RANGE from 0.84 to 4.2 meters from the camera[6].

EXPOSURE TIME 14 ms maximum.

LATENCY TIME 20 ms for acquired data delivery to the main system software.

DEPTH ACCURACY within 2 percent.

LIGHTING INDEPENDENCE possibility to operate with or without any external light source and even outdoor[7].

With respect to what explained in chapter 2, the Kinect for Xbox One sensors implementation presents some differences. The emitted signal is modulated with a square wave instead of the common sinusoidal modulation. Moreover each pixel of the receiver generates two outputs and the incoming photons (i.e. the reflected signal or the ambient light) contribute to one or the other according to the state of a clock signal. This solution limits the effects related to harmonic distortions.

Other featured expedients are the use of *multiple modulation frequencies* to prevent the phase-wrapping effect of section 2.2.1 and two different *shutter times* (of 100 and 1000µs) to achieve optimal exposure even with a fixed aperture.

Note that, in opposition to the Creative Senz3D, this device shows better results while trying to acquire darker colors like black clothes or plastic objects (especially on opaque surfaces)[8].

The scene planes are, also, less affected by deformations and usually appear like reasonably flat surfaces.

[6] These specifications are referred to the device ability to handle gestures and body tracking. The actual is wider: experimental results shows a range form 0.5 to 8 or 10 meters.

[7] The ability to operate outdoor even under the direct sunlight is a great advance with respect to the Kinect for Xbox 360 and the majority of other consumer ToF cameras.

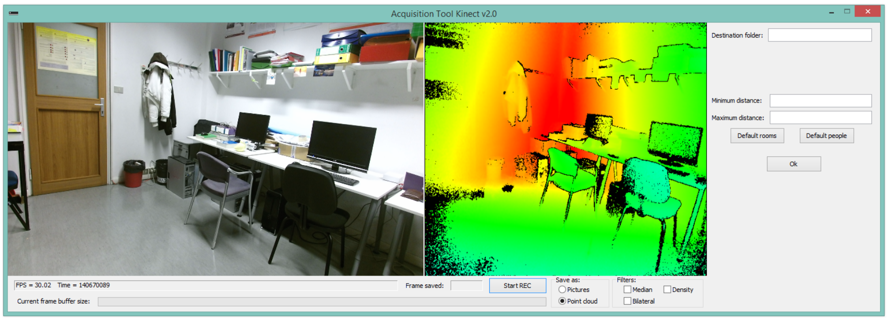
[8] Probably this behaviour is due to the higher power emission of this device.

5.2.2 Device limitations

Indubitably the Kinect for Xbox One presents very good performances at a limited cost. It is one step ahead the Kinect for Xbox 360 as well as many other consumer products of the same market share. Nevertheless there are some minor drawbacks.

The device has higher power requirements compared to the previous version or the Senz3D, so it cannot rely on the USB port any more, but it needs an AC adapter. Moreover the connection to the computer is possible via another specific adapter. As a consequence handling the cables could not be always easy, especially while moving around an object to perform a 360 degrees acquisition.

Transferring high definition data at high frame rates is a challenging task, therefore an USB 3.0 port is required and computers that lack this kind of port cannot connect to the device.



FPS = 30.02 Time = 140670089

Save as:
○ Pictures
● Point cloud

Filters:
□ Median □ Density
□ Bilateral

Figure 10: Screenshot of the acquisition application for the Kinect for Xbox One range camera.

5.2.3 Acquisition application

The software for acquiring data form the Kinect for Xbox One has been written in C++. It is a *multithreading* application that makes use of the Kinect for Windows SDK v2.0 and some functions provided by the OpenCV library[9].

The program establishes the connection with the range camera and shows on the screen the real-time color and depth data streams. The latter exploits a false color scale to represent the distance of the objects inside the scene.

The user can start and stop the recording of the data streams which are buffered to prevent any losses of frames during the saving operation. There is the possibility to choose the output format between digital pictures and colored point clouds[10].

In order to process the acquired data before the registration procedure,the same filters presented in section 5.1.3 have been implemented and can be applied.

Finally, the minimum and maximum allowed depth values can be specified: any sample outside this interval will be discarded.

[9] OpenCV is a an *open source* library suited for computer vision developing and image processing.

[10] In this case the color map is reprojected over the depth data.

5.3 RECONSTRUCTION APPLICATION

The application that computes the actual 3D reconstruction is standalone and can receive as input sets of views acquired by different devices. It has been written[11] in C++ and uses the OpenCV and PCL libraries.

The point clouds generated by the acquisition applications are taken as input then registered and fused together following the algorithms described in chapters 3 and 4. The output is a single point cloud with the complete 3D model of the scene.

There is no graphical interface since very little user interaction is required. The setting of some customizable parameters can be done via a configuration text file. The sate of the reconstruction process and any possible error that may occur are showed on a command line interface.

[11] Originally created by Enrico Cappelletto and then modified following what has been previously explained.

6 EXPERIMENTAL RESULTS

In this chapter some experimental results will be showed. Both the devices mentioned above has been tested using them as a handheld scanner. The users walked around the object to be acquired while keeping the camera pointed at it or turned around himself when acquiring a 360 degrees view of the room which is showed by Figure 11.

6.1 EXPERIMENTAL RESULTS WITH SENZ3D

A set of boxes of various dimensions and colors has been reconstructed with the Creative Senz3D. The process involved the fusion of 351 frames and generated a final point cloud of 1520294 samples.

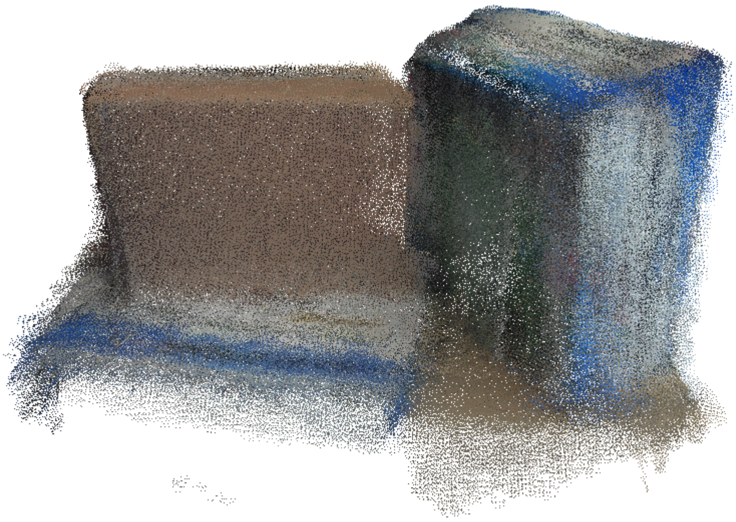
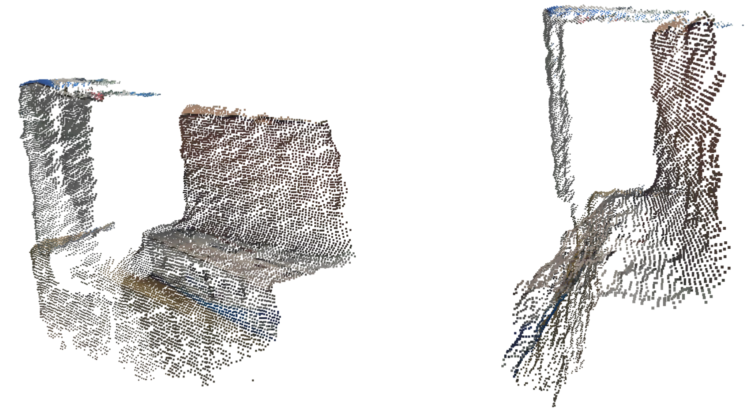


Figure 11: Final point cloud of a set of boxes reconstructed with the Creative Senz3D range camera.

The result shows that the reconstruction process worked properly, but is noisy and not well defined. The reasons of this poor result are due to the low accuracy and resolution of the device. The Creative Senz3D has not been thought as a measuring device, but as a interactive camera more suitable to detect hand gestures than the exact distances of the scene points.

Watching a single view closer, some issues of this sensor are clearly visible (see Figure 12). The shape of the boxes is not captured properly. Especially along the corners there are many deformations that bend the surfaces. The side of the brow box, also, is not flat at all. It has many bumps that make



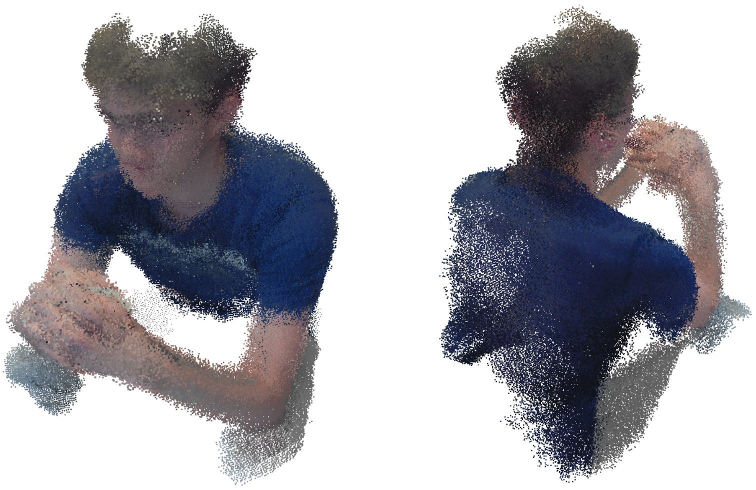
(a) Front-view point cloud detail (b) Side-view point cloud detail

Figure 12: Details of a single view of the acquired set of boxes. The deformation issues are clearly visible.

both the alignment process very hard to accomplish and the final result noisy.

Figure 13 shows the 3D reconstruction of a sitting person. An amount of 301 frames has been captured and the reconstruction generated a final point cloud of 1482002 samples.

The result appears to be better, compared to the previous example, because the curved shape of the body makes the deformations less noticeable. Actually, the absence of sharp concave edges reduces the amount of disturbance that affect the shape of the point clouds since the multi-path propagation effect is weaker. The posture and the body parts are clearly distinguishable, but if some details of the face and fingers are not recognizable.



(a) Front-view point cloud detail (b) Back-view point cloud detail

Figure 13: Reconstruction of a person made with the CREATIVE SENZ3D range camera.

6.2 EXPERIMENTAL RESULTS WITH KINECT FOR XBOX ONE

Compared to the Senz3D, the Microsoft Kinect for Xbox One is way more powerful and accurate. Thanks to his broader operative range, it has been possible to acquire the whole LTTM[1] laboratory room that is shown in Figure 14. The acquisition input is made of 421 frames that have been fused together to form a final point cloud made of 2001385 points.

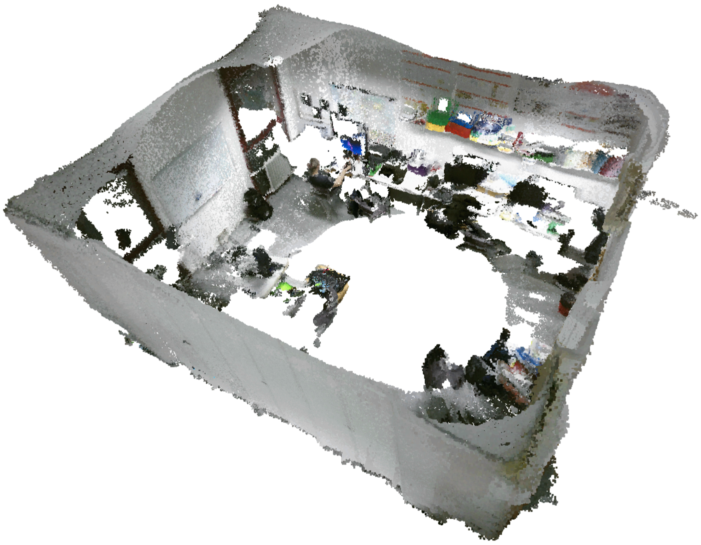


Figure 14: Final point cloud result of the 3D reconstruction of a room.

The outcome is indubitably superior compared to the reconstructions made with the Creative Senz3D. Looking at Figure 15, it is possible to recognize the objects in the scene. Both the shapes and the colors are well defined. Even many black objects have been accurately acquired. The noise level is quite low and the room structure has been properly rendered.

The Kinect for Xbox One is less affected by the issue of the Senz3D. In particular the flat surfaces acquired with the former appear to be smooth and even (see Figure 16). This fact, the higher resolution of the color and infra-red cameras, the overall accuracy and precision of the device make the alignment process less cumbersome. Therefore good reconstructions are more easily achieved.

[1] LTTM stands for *Multimedia Technology and Telecommunications Lab.*

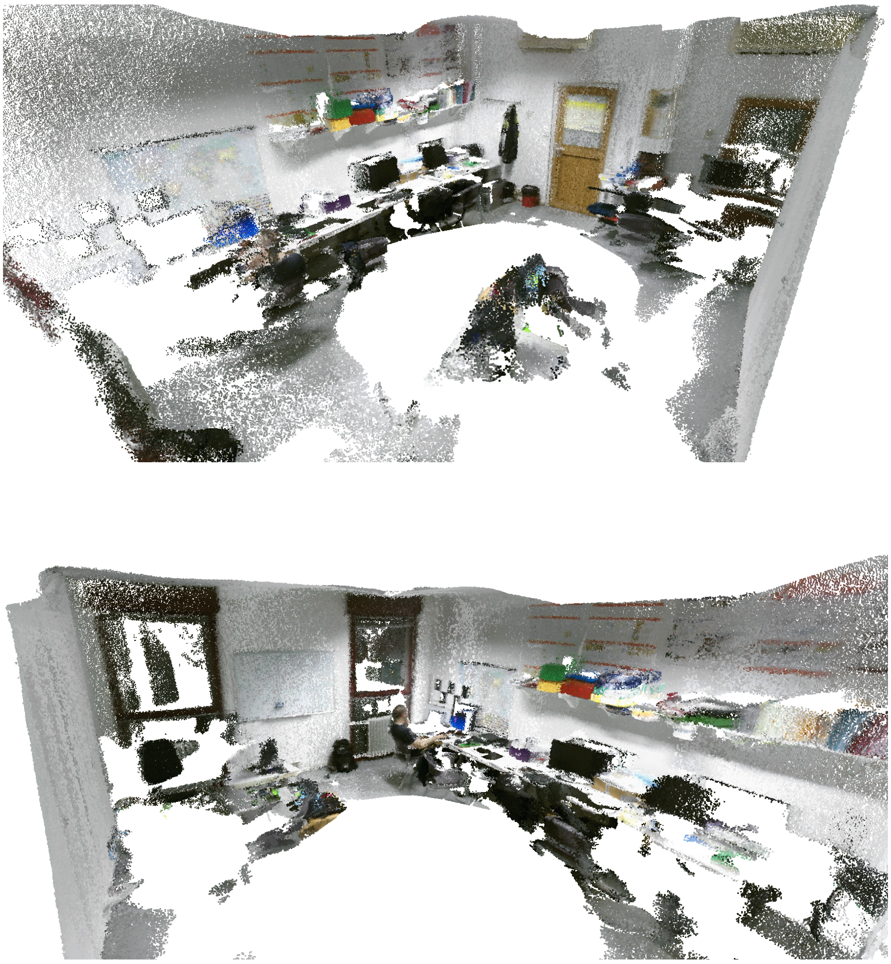
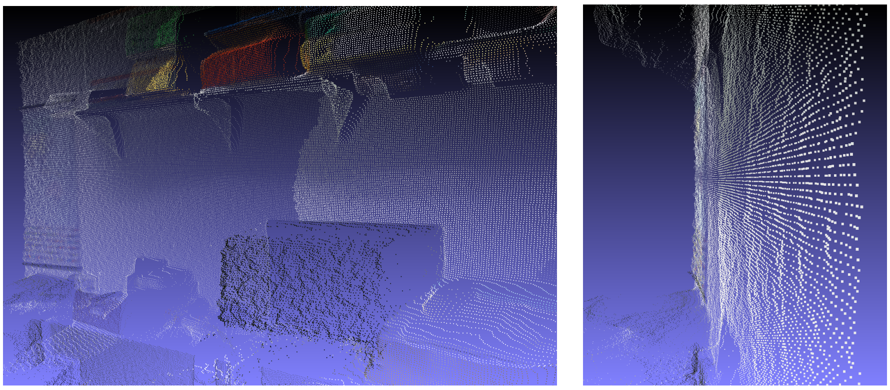


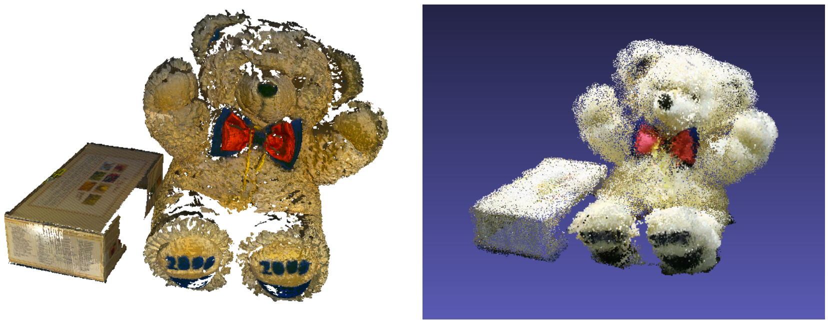
Figure 15: Reconstruction of the LTTM laboratory room made with the Microsoft Kinect for Xbox One range camera. The two pictures show the final point cloud from different points of view.



(a) Front-view point cloud detail

(b) Side-view point cloud detail

Figure 16: Details of a single view of the acquired room. The flat surface of the wall is not affected by the deformations visible in Figure 12. Only the dark computer screen on the bottom appears roughly irregular, but this is mostly due to the translucency of the material.



(a) Ground truth model mesh

(b) Reconstructed model point cloud

Figure 17: Ground truth model compared to a reconstructed one from the data acquired with the Kinect for Xbox One range camera.

Table 1: Accuracy measures through ground truth comparison.

Measurement No.	Mean distance [mm]	Standard deviation [mm]	Point cloud size
1	11.3071	9.0541	169814
2	12.9166	10.5491	182427
3	11.8071	9.4014	163091
4	11.1993	9.1719	138538
5	12.1339	9.6379	191159

6.2.1 Accuracy evaluation

In order to evaluate the accuracy of the reconstruction process combined with the Kinect for Xbox One sensors, another 3D scene has been acquired. A *ground truth* 3D model was generated using a NextEngine 2020i Desktop Laser Scan (see Figure 17).

Using CloudCompare[2], the reconstructed point cloud has been aligned to the ground truth model (exploiting some common points and an ICP refinement). Then the aligned models have been compared and the average distance and standard deviation of their samples have been computed.

The acquisition with the Kinect for Xbox One has been repeated five times comparing each reconstructed point cloud to the ground truth model. The results are presented in Table 1.

As expected the accuracy of the Kinect for Xbox One is smaller than the one of a laser scanner, but it allows to obtain reasonable results. Since while capturing the scene, the average distance between the sensor and the objects was about 1 meter, the difference between the two models stays inside the error ranges of the sensors presented in section 5.2.1, therefore the reconstruction algorithm of chapters 3 and 4 works properly.

Note that a teddy bear is not the easiest object to acquire with a Tof camera since his plush fabric forms a very noisy and irregular surface. In fact, as showed by Figure18, the samples on the box at the side of the stuffed toy present an higher accuracy than the points on the teddy bear.

[2] CloudCompare is an open source software designed for point cloud processing.

Table 2: Comparison between the ground truth and the point cloud generated with the original fusion algorithm of chapter 3.

Algorithm	Mean distance [mm]	Standard deviation [mm]	Point cloud size
original	15.0315	10.9532	857122
proposed	11.3071	9.0541	169814

Finally a reconstruction with the unmodified version of the fusion algorithm (i.e. the one described in chapter 3 without the upgrades of chapter 4) has been done and the comparison between the generated point cloud and the ground truth led to the results showed in Table 2.

The mean distance and the standard deviation are slightly worse compared to the results obtained with the fusion procedure described in chapter 4 and the number of samples of the final point cloud is clearly larger. As a consequence the proposed fusion algorithm generates cleaner point clouds.

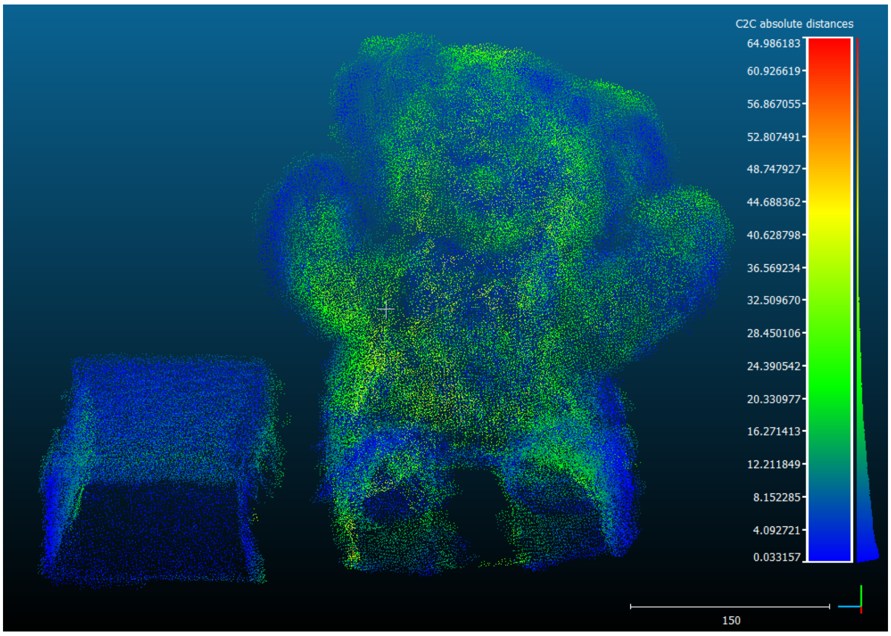


Figure 18: Distance between the ground truth and the first reconstructed point cloud of Table 1. The color scale represent the distance that every point has from the ground truth model. The mean distance is 11.3 mm and the standard deviation is 9.05 mm.

7 CONCLUSIONS

The work presented here consisted in an improvement of the algorithm described by [Cappelletto et al., 2013] suitable for use with the customer Time-of-Flight sensors Creative Senz3D and Microsoft Kinect for Xbox One and an implementation of a software system able to interact with the range cameras and generate a final reconstructed 3D model followed by a series of experimental measurements with the purpose of evaluating the accuracy of the reconstruction process.

A simple, but more refined, fusion strategy has been adopted in order to merge together the different point clouds aligned during the registration procedure and reduce the global amount of samples preventing the loss of geometric detail at the same time.

The experimental results show that the adopted approach has been successful. Different 3D scenes have been acquired and reconstructed obtaining satisfactory results. The comparison with a ground truth model captured with a 3D laser scanner showed the reasonable accuracy of the measurements and the observed procedure.

The conducted tests affirmed the superior performances of the Kinect for Xbox One with respect to the Senz3D sensor. Data acquired by the former sensor are more detailed both in colors and geometry, moreover are less affected by deformations. This fact prevents issues during the alignment process and leads to better reconstructed models.

Further research could be focused on more sophisticated solutions for what concerns the handling of the color data during the fusion procedure. Another improvement to be considered is the development of a better global alignment procedure in order to further enhance the reconstruction accuracy.

This work confirmed that using consumer range cameras as a handheld scanner is an easy task for the user and allows fast and accurate 3D reconstructions.

BIBLIOGRAPHY

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

T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard, and J. McDonald, "Robust Tracking for Real-Time Dense RGB-D Mapping with Kintinuous," *Computer Science and Artificial Intelligence Laboratory Technical Report MIT-CSAIL-TR-2012-031*, September 2012.

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, April 2012.

E. Cappelletto, P. Zanuttigh, and G. Cortelazzo, "Handheld scanning with 3D cameras," in *Multimedia Signal Processing (MMSP), 2013 IEEE 15th International Workshop on*, Sept 2013, pp. 367–372.

P. Besl and N. D. McKay, "A method for registration of 3-D shapes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 14, no. 2, pp. 239–256, Feb 1992.

J. Sprickerhof, A. NÃijchter, K. Lingemann, and J. Hertzberg, "An Explicit Loop Closing Technique for 6D SLAM," in *ECMR'09*, 2009, pp. 229–234.

R. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 1–4.

Creative website, Creative Senz3D specifications. [Online]. Available: http://us.creative.com/p/web-cameras/creative-senz3d

SoftKinetic website, SoftKinetic DepthSense cameras specifications. [Online]. Available: http://www.softkinetic.com/Products/DepthSense-Cameras

J. Sell and P. O'Connor, "The Xbox One System on a Chip and Kinect Sensor," *Micro, IEEE*, vol. 34, no. 2, pp. 44–53, Mar 2014.

C. Dal Mutto, P. Zanuttigh, and G. Cortelazzo, *Time-of-Flight Cameras and Microsoft Kinect*, ser. SpringerBriefs in Electrical and Computer Engineering. Springer, 2012.

A. Payne *et al.*, "7.6 A 512×424 CMOS 3D Time-of-Flight image sensor with multi-frequency photo-demodulation up to 130MHz and 2GS/s ADC," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, Feb 2014, pp. 134–135.

K. Khoshelham, "Accuracy analysis of Kinect depth data," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVIII-5/W12, pp. 133–138, 2011. [Online]. Available: http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XXXVIII-5-W12/133/2011/

ACKNOWLEDGMENTS

I really want to thank the best supervisor ever: Prof. Pietro Zanuttigh, who has always been extremely helpful and friendly, and my co-supervisor Enrico Cappelletto who gave me lots of advices and helped me to solve every problem I encountered working at this thesis, even when he was busy.

I thank as well the other *colleagues* at 3DEverywhere who made me have a great time: Prof. Guido Maria Cortelazzo, Giovanna, Davide (*the good one*), Mauro e Matteo, besides the foam rubber ball, the unlucky plant and the Roadhouse. It has been a wonderful and interesting experience.

Of course I want to thank my parents, I would have never reached this goal without their and my family's tremendous support and encouragement. A special thanks goes to my grandparents Lucia, Anna e Giovanni, I wish you were here with me today.

I would like to thank two of the best friends someone can have: Carlo and Lorenzo. Thank you for the awesome things we have done together since high school. Let's keep going!

Ψ *Il Triumvirato*

A big and strong thanks goes to the members [*eheh*] of the *Appartamentissimo* Carlo, Dario, Olivi, Raga, Pado, Toni, Raghiño, Niccolò, Faster, Tarta and Patrick. Thank you for all the time we spent studying and for the rare moments of fun we had. Living with you has been worth all the years I spent at university!

I would also like to thank Giulia, for giving me the strength to finish my exams and get my master's degree.

Finally, I would like to thank all my friends, the old ones from my hometown Vittorio Veneto and the new ones I made in Padova, like my mates at Collegio Leopardi.

Ci è voluto un po',
ma alla fine ce l'ho fatta!
gg
Andrea Bernardi

Padova, 9 december 2014