Flight Autonomy of Micro-Drone in Indoor Environments using LiDAR Flash Camera

*Aarón Montoya Salado , **Bertrand Vandeportaele , ***Simon Lacroix ****Gautier Hattenberger

***Abstract*— *Autonomy* starts with awareness of the environment. Robots are given autonomy using sensors that endow them with perceptual capabilities, such as *cameras*. Recently, a new type of camera working under the *Time-of-Flight* principle has been developed, capable of acquiring dense depth maps at high frame rates. Its small size and weight make it suitable for its use on-board a flying vehicle for indoor *localization* and *mapping*. This document outlines the first approaches taken in the use of a ToF camera for such tasks constrained by real-time requirements. The camera has been mounted on a flying vehicle that uses the open source Paparazzi autopilot system developed by the ENAC (*Ecole National de l'Aviation Civile*) french team. Since indoor environments are predominantly planar, *planar patches* have been favoured to model the environment and detect motion of the UAV. A *Region Growing* segmentation algorithm identifies and extracts planes from the scene in real-time. Planes are tracked and registered across a sequence of frames to estimate the camera's *ego-motion*. Initial results of plane-based *visual odometry* are presented and confirm the device suitability.**

I. INTRODUCTION

Mobile robots rely on sensors for tasks such as *localization*, *mapping*, *obstacle avoidance*, *path planning*... In general, the most popular sensor is the *laser rangefinder*, which normally provides a 180° view on the plane of measurement. However, non-simulataneous acquisition of points, added to their high costs, limit their widespread use. Alternatively a *stereo-vision* system can be considered, which requires the matching of corresponding points from two images to obtain depth information. However, stereo-vision is computationally costly and is limited to textured scenes [2]. Recently, a new type of camera has appeared, with the ability to produce dense three-dimensional data in real-time. It employs an array of Infrared LEDs that illuminate the scene with modulated light. Light returns to the camera, and distance is recovered by measuring the phase shift of the reflection. This is known as the *Time-of-Flight* principle. ToF cameras are advantageous over stereo-vision because of their ability to reconstruct in 3D an observed scene regardless of the textures on it. ToF cameras have been succesfully evaluated for its use in mobile ground robotics [3], [4], and adapted to scenarios where a mobile robot uses acquired data to perform different tasks [5], [6]. Its small size and weight make it a promising alternative to other sensors for its use on-board a flying vehicle. To our knowledge, this type of camera has never been used before on micro *Unmanned Aerial Vehicles*. This document synthesizes the main outcomes of our initial investigations using a ToF camera. The remainder of this paper is structured as follows: Section II summarizes relevant related work. Section III shortly introduces the UAV with which some initial experiments have been performed, as well as the ToF camera SwissRanger SR-3000 that has been used for our investigations. Section IV describes the process for *ego-motion* estimation, based on the extraction, tracking and registration of planes on the scene. Section V shows initial results in the estimation of the camera orientation. Finally Section VI concludes this document.

*Student in the Erasmus Mundus Master in Computer Vision and Robotics (VIBOT).
amontoya@laas.fr
**Associate Professor at Université de Toulouse: UPS, INSA, INP, ISAE; Researcher at LAAS-CNRS.
bvandepo@laas.fr
***Senior Research Scientist in Mobile Robotics.
Simon.Lacroix@laas.fr
Laboratory for Analysis and Architecture of Systems (LAAS-CNRS). 7 Avenue du Colonel Roche F-31077, TOULOUSE, France.

***Research Scientist in Aerial Robotics.
gautier.hattenberger@enac.fr
Ecole National de l'Aviation Civile (ENAC). 7 Avenue Edouard Belin, 31400, TOULOUSE, France.

II. RELATED WORK

One of the first applications in robotics considering ToF cameras as an alternative to laser scanning, stereo or monocular vision, was presented in 2004. Weingarten, Grüner and Siegwart [3] evaluated a SwissRanger SR-2 device in terms of basic obstacle avoidance and local path planning capabilities. Navigation and path planning was performed robustly based on the provided data.

In 2006, Ohno et al. used a SwissRanger SR-2 to estimate a robot's trajectory and reconstruct the surface of the environment [5]. The calculated trajectory was compared with precise reference data in order to demonstrate the algorithm's precision.

Pose Estimation and *Mapping* has recently been addressed by May et al. [7], [8]. They evaluate the pose estimation accuracy of four different approaches that extract features suitable for registration: scanline approximation, Karhunen-Loève Transform (KLT), Scale-Invariant Feature Transform (SIFT), and no data reduction. To map the scene, first invalid data points are discarded by filtering. Next the map is generated incrementally by registering consecutive 3D captures. The accumulated error during motion is determined and distributed by loop closing, and finally the map is refined. Accurate maps are built using these strategies.

In order to overcome the low measurement accuracy of the camera, error modelling and calibration of these devices has been addressed by some researchers. Because ToF cameras provide monochromatic reflectance images and are based

on the *pinhole camera model*, photogrammetric calibration is feasible, which is easily done using *Bouguet's Camera Calibration Toolbox* for MATLAB® [9]. Additionally, a calibration of the provided range data is performed to collect accurate depth measurements. Lindner and Kolb [10], Kahlmann et al. [11], and Fuchs et al. [12], [13] have already addressed these issues improving the sensor's accuracy.

Time-of-Flight cameras present a number of important errors. While some of them cannot be corrected, the most critical ones can. *Light Scattering* occurs in the camera's internal optical path due to secondary light reflections that are not absorbed by the imager. The effect is the superposition of measurements from the background, making them appear closer. *Wrap-Around* is due to the phase-shifting principle of measurement of these devices. Since phase-shift is assumed in the range from 0° to 360°, a phase-shift of 370° is measured as 10°. This obviously also corrupts the distance measurements. This last problem has been addressed by Dubois and Hügli [14], [15], who proposed an algorithm that reduces scattering by solving an inverse deconvolution problem. Poppinga and Birk showed an approach that corrects several errors of ToF cameras at once [16]. They correlate the amplitude and depth images to identify and remove erroneous pixels. *Wrap-Around*, *light scattering* and other effects are effectively fixed using this approach. Another important problem is the so-called *Jump Edges*. These occur in the transition between two imaged objects. Objects physically disconnected will be seen by the ToF camera smoothly bridged by sparsely distributed points. A detailed overview of these and other ToF cameras noise and errors sources can be found in [17].

The use of *planar features* for localization and mapping is a relatively new approach that has only recently been considered. Weingarten, Grüner and Siegwart proposed an algorithm to recover planar surfaces of physical environments without considering topologically ordered points [18]. The same authors presented in [19] an improved method for plane fitting that considers probabilistic information to propagate uncertainty. Vaskevicius, Birk, Pathak and Poppinga proposed in [20], [21] a fast and accurate method to extract planes from clouds of 3D points. An optimized *Region Growing* algorithm is implemented thanks to the reformulation of the mathematical principles behind the least-squares plane fitting.

Being ToF cameras relatively new technologies, only a few approaches related to *ego-motion* estimation and *visual odometry* using such devices have seen the light. Dröschel et al. perform ego-motion estimation using a SwissRanger SR-3000 ToF camera [22]. SIFT features are extracted from the reflectance image and succesively matched across frames. For each matched pixel, a 3D coordinate is known. A set of known correspondences makes it possible to register the clouds of 3D points. Results are refined with the information provided by an *Inertial Measurement Unit* (IMU). A Kalman Filter predicts the system velocity estimate using the IMU data, and the motion estimates are fused. Pose estimation based on registration of planar features has been addressed first by Weingarten et al. [23]. His contributions include a probabilistic model of planes that contains their associated uncertainty, evaluation of different approaches to plane segmentation, and finally a novel SLAM algorithm based on the Extended Kalman Filter (EKF) adapted for using planar segments. More recently, Pathak et al. also considered planar segments as useful features for a robot to use in localization and mapping tasks [24], [25]. They describe a strategy for performing plane matching and decoupling the rotations and translations from the corresponding normals between two frames. They develop a complex mathematical formulation that allows to decouple the full pose of the robot, while considering the plane's associated uncertainties. Results are presented using different sensors, including ToF cameras and laser rangefinders.

III. DEVICES DESCRIPTION

A. The Flying Micro-UAV

The flying vehicle with which experiments have been started is a micro-UAV property of ENAC. It is a quadrotor powered by four brushless Roxxy motors, each mounted in the corners of a cross-shaped MK-40 frame made of aluminium shaped profiles by Mikrokopter. The central part of the UAV is custom made. The total length and width of the UAV is 70 cm. including propellers. Distance between motors is 45 cm. The datalink is a 2.40 GHz X-Bee Pro modem. The vehicle is controlled by a Paparazzi-Booz autopilot developed at ENAC [1]. This autopilot consists of 3 boards: the main board that comprises power supply, a LPC2148 32-bit ARM7 microcontroller and a barometer; the IMU board, comprising 3-axis gyroscopes, accelerometers, magnetometers and a 16-bit ADC; finally, the GPS board, using a LEA-5H GPS and Galileo module for positioning. It has a weight of 700 g. with batteries, which goes up to 1 kg. when the ToF camera and an XSens MTi IMU are mounted. The maximum payload is of around 500 g. The vehicle is shown in Fig. 1 with the ToF camera and the IMU mounted.

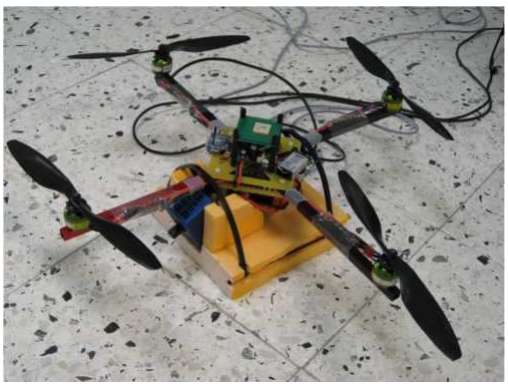


Fig. 1: The ENAC micro-UAV with the SwissRanger SR-3000 and the IMU XSens MTi

B. The SwissRanger SR-3000

The ToF camera that has been investigated is the SwissRanger SR-3000, developed by the *Centre Suisse*

d'Electronique et de Microtechnique (CSEM). It is a *state-of-the-art*, solid-state imaging device working under the Time-of-Flight principle, designed for operation under controlled lightning conditions. It's capable of delivering *depth* as well as gray-level *intensity* images at high frame rates [26].

Fig. 2: The CSEM SwissRanger SR-3000 Camera

The SR-3000 camera is based on a 2-dimensional dedicated image sensor with a spatial resolution of 176×144 pixels, manufactured in 0.6 μm CMOS/CCD technology. The size of each pixel is 40×40 μm^2. It has a set of 55 Near-Infrared (NIR) LEDs used to illuminate the scene. The emitted light is typically modulated at an internal frequency of 20 MHz, yielding a non-ambiguity range of 7.5 m. The lense has a *Field of View* (FOV) of 47.5×39.6 degrees and an aperture diameter of 5.7 mm. It consumes an average power of 12 W, which can go up to 18 W, depending on the integration time. The device contains no moving parts. Its size (50×67×42.3 mm^3) and relatively low weight (160 gr.), make it practical and easy to handle.

The principle of operation of the device is as follows: first the observed scene is illuminated with modulated near-infrared light (NIR) that is reflected by the objects, travelling back to the camera. The light is absorbed by a CCD/CMOS sensor. Every pixel on the sensor samples the amount of modulated light reflected by objects in the scene. This is done four times every period at equal intervals. These measurements, allow to recover the sinusoidal incoming signal, and thus, the distance to the objects on the scene (Fig. 3).

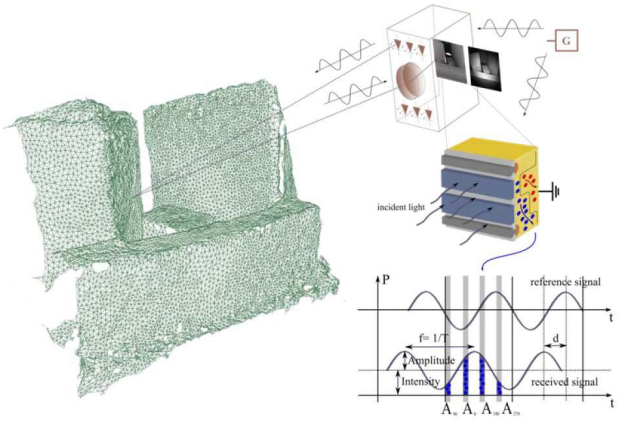


Fig. 3: Time-of-Flight operating principle. Distances are estimated by measuring phase-shift. Figure taken from [8]

IV. METHODOLOGY

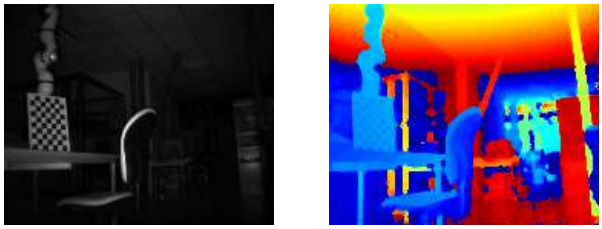
This section briefly outlines the algorithms, techniques and strategies followed in our first investigations. Starting from the *calibration* of the SwissRanger SR-3000, and the evaluation of *filtering* techniques, we continue describing the algorithms that allow to extract planes from the scene observed by the SwissRanger SR-3000 in real-time, followed by a description of the *tracking* and *registration* strategy we have adopted. Finally, *ego-motion* is described, which allows to estimate the pose of the camera between consecutive frames.

A. Camera Calibration

Photogrammetric calibration of the SwissRanger SR-3000 is feasible, given the *intensity* images it provides. Calibration is done in the same way a regular camera is calibrated. Several shots of a *checkerboard* are taken at different positions and orientations. These are then processed using **Bouguet's Camera Calibration Toolbox** for MATLAB®. The result is the camera's main characteristics: the *principal point* and the *focal lengths*. Since the images provided by the ToF camera are low in contrast, images are pre-processed by simple *histogram equalization* that enhances image quality. Knowledge of the camera's *Intrinsic Parameters* allows to convert the depth readings provided by the camera to *Cartesian Coordinates*.

B. Filtering

Some of the most critical errors of ToF cameras have been already described above. This section proposes approaches to effectively remove them from the made measurements. The effect of *Wrap-Around* can be observed in Fig. 4. Distance measurements are color coded in Fig. 4b; while blue color means *closer*, red color means *distant*. Notice that after a certain distance, a sudden change in depth measurement is observed, which means that the non-ambiguity range of 7.5 m. has been exceeded.



(a) Intensity image (b) Depth image

Fig. 4: Typical Indoor Scene with evident Wrap-Around

Although this problem can be dealed with by simple *amplitude thesholding*, this tends to discard not only wrong pixels, but also a significant number of valid pixels that correspond to low reflectivity surfaces on the scene. A better solution has been proposed by Poppinga et al. [16], that relates both *depth* and *intensity* images to identify invalid pixels. The basic principle is that correct depth measurements correspond to brightness values that decrease as the distance

to the imaged object increases, due to the fact that distant points in the scene reflect only a little amount of the emitted light. In contrast, wrapped-around pixels present low distance values and significantly low brightness since the object is farther away than what measurements suggest. Fig. 5 shows the result of using this filtering scheme in the same indoor scene.

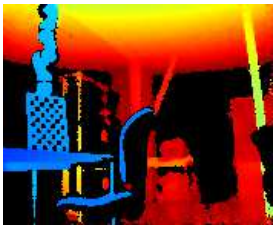
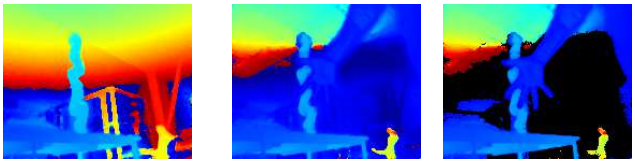


Fig. 5: Filtering with the approach proposed in [16]

Not only the *wrap-around* effect has been corrected, but *black objects*, whose depth measurements tend to be noisy, have also been nicely removed. Also, *bright light*, which introduces large amounts of random noise is removed. Another important error this approach corrects is the *light scattering* phenomenon. Fig. 6 shows the filtering of a scene that has been corrupted by the introduction of a foreground object. Wrong pixels are correctly removed using the adopted filtering scheme. The removal of this effect is beneficial for tasks such as *obstacle avoidance*, where it is important to correctly observe objects the robot could collide with. By filtering, the shape of the obstacle is precisely known, and a proper strategy to avoid it can be implemented.



(a) Depth image of indoor scene

(b) Introduced foreground distorts most of the pixels around

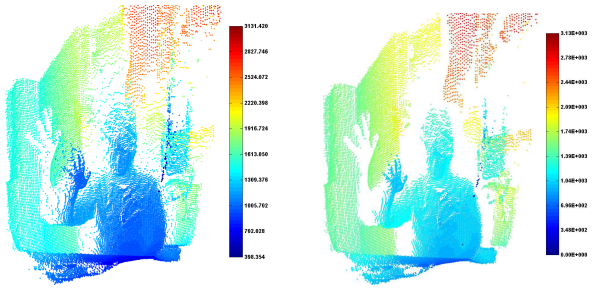
(c) Adaptative filtering discards distorted pixels

Fig. 6: Effect of the light scattering phenomenon and the proposed algorithm results

Jump Edges are filtered as proposed by Khongsab in [27]. Each point is compared with its eight neighbours. If the distance between a point and one of its neighbours exceeds a threshold, a counter is incremented. If the counter increases a certain number of times, the point is classified as a Jump Edge and filtered. Fig. 7a shows a cloud of 3D points where the effect of Jump Edges is evident. The result of filtering them is shown in Fig. 7b.

C. Feature Extraction

As it has been said before, the use of *planar features* has been motivated by the fact that the camera will be mainly used on indoor scenarios. Here the approach proposed in [20], [21] is adopted. It aims at the fast extraction of planes



(a) Before Jump Edges removal

(b) After Jump Edges removal

Fig. 7: The Jump Edges problem and proposed algorithm results. Erroneous points at the interface of the human and the wall have been removed

from the depth images provided by a ToF camera. In general, planes are described by the following expression:

$$Ax + By + Cz + D = 0 \tag{1}$$

where *A*, *B* and *C* form a *normal vector* of unit magnitude, which describes the vector normal to the plane, and *D* is the perpendicular distance from the plane to the origin. *x*, *y* and *z*) on the other hand are the coordinates of a 3D point. Given N number of points $\mathbf{p_i} = [\ x_i\ y_i\ z_i\ ]^{\mathrm{T}}$, the mass center $\mathbf{m} = [\ \bar{x}\ \bar{y}\ \bar{z}\ ]^{\mathrm{T}}$ of this set of points is:

$$\mathbf{m} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{p_i} \tag{2}$$

which is used to calculate the following *Covariance Matrix*:

$$C = \sum_{i=1}^{N}\begin{pmatrix} c_n(x_i,x_i) & c_n(x_i,y_i) & c_n(x_i,z_i) \\ c_n(y_i,x_i) & c_n(y_i,y_i) & c_n(y_i,z_i) \\ c_n(z_i,x_i) & c_n(z_i,y_i) & c_n(z_i,z_i) \end{pmatrix} \tag{3}$$

where each element is given as:

$$c_n(a,b) = \sum_{i=1}^{n}(a_i - \bar{a})(b_i - \bar{b}) \tag{4}$$

From 3, three *eigenvalues* and three associated *eigenvectors* can be calculated. The smallest eigenvalue points to the eigenvector that defines a vector normal to the plane fitted to the N points. The vectors components are the parameters *A*, *B* and *C*. Element *D* is given as $D = -A\bar{x} - B\bar{y} - C\bar{z}$. The *Mean Square Error* can be used as a quality measure of the fitted plane:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(Ax_i + By_i + Cz_i + D)^2 \tag{5}$$

Given these mathematical formulations, a *Region Growing* segmentation algorithm is implemented, which aims at extracting planar regions from the scene. The algorithm starts selecting a point $\mathbf{p_1}$ and its nearest neighbor $\mathbf{p_2}$, and labels them as a new region. Then the region is extended evaluating neighbouring candidate points, according to their distance

to the formed plane, and the new MSE value that would result from adding the point to the plane. The region is grown until no more points can be added, and a new point is selected to grow a new plane. The algorithm stops when no more planes can be growed. An efficient implementation is achieved by reformulating expressions 3 and 5. This mathematical reformulation can be found in [20], [21]. Fig. 8 shows illustrative results of the segmentation algorithm.

(a) Intensity image (b) Depth image (c) Segmented image

Fig. 8: Results of the adopted segmentation algorithm

D. Feature Tracking

Once planes have been extracted, we propose to track them across a sequence of frames. This is achieved following a simple approach based on *morphological operations*. The first frame in the sequence is fully segmented. Then, a "prediction" is created by eroding the borders of the extracted planes. Thanks to the high frame rates the camera achieves, the eroded image can be superimposed on the following frame correctly. This prediction guides the new segmentation and planes are extracted again. Not only does this approach avoids extracting planes from scratch at each new frame, but also automatically solves the correspondence problem. The method is illustrated in Fig. 9.

(a) Frame " t " (b) Prediction (c) Frame " t+1 "

Fig. 9: Planes Tracking and Matching across a sequence of frames. By eroding frame " t ", planes are predicted in frame " t+1 "

E. Polygonization

Once planes are extracted, a *polygonization* step follows, which aims at converting surfaces into polygons for proper 3D visualization. Two algorithms have been implemented: the first one, known as *Graham Scan* [28], creates a *Convex Hull*, which is the envelope that contains the minimal set of points required to represent the polygon; the second algorithm creates a *Concave Hull*, which has a higher level of representation detail. Fig. 10 shows respectively the results of both algorithms on the planes shown in Fig. 8c.

The use of polygons will be further investigated as another alternative to the proposed plane-prediction algorithm.

(a) Convex Hull (b) Concave Hull

Fig. 10: Polygonization results

F. Ego-Motion & Visual Odometry

Ego-motion consists in the movement the camera experiments between two succesive frames. The composition of estimated ego-motions gives us the *visual odometry*: the accumulated motion the camera experiments in time.

Let us define two observations of the same normal vector in consecutive frames $\mathcal{F}_i$ and $\mathcal{F}_j$ as $\mathbf{n_i} = [\ A_i\ B_i\ C_i\ ]^{\mathrm{T}}$ and $\mathbf{n_j} = [\ A_j\ B_j\ C_j\ ]^{\mathrm{T}}$ respectively. The *Homogeneous* transformation that relates corresponding normal vectors is given as:

$$\nu_i = \mathbf{H}^{-\mathrm{T}}\nu_j \tag{6}$$

where $\nu_i = [\ \mathbf{n_i}^{\mathrm{T}}\ D_i\ ]^{\mathrm{T}}$ and $\nu_j = [\ \mathbf{n_j}^{\mathrm{T}}\ D_j\ ]^{\mathrm{T}}$. Rotational and Translational terms in 6 are contained within as:

$$\nu_i = \begin{bmatrix} \mathbf{R}^{\mathrm{T}} & 0_{3\times1} \\ -\mathbf{T}^{\mathrm{T}}\mathbf{R} & 1 \end{bmatrix} \nu_j \tag{7}$$

Rotations and Translations are decoupled from 6 such that:

$$\mathbf{n_i} = \mathbf{R}^{\mathrm{T}}\mathbf{n_j} \tag{8}$$

$$-\mathbf{T}^{\mathrm{T}}\mathbf{R}\mathbf{n_j} = D_i - D_j \tag{9}$$

While 8 requires to be solved via non-linear optimization, using methods such as *Levenberg-Marquardt*, 9 is a linear expression that can be solved using *Linear Least Squares* or *Singular Value Decomposition* (SVD). The solution of both expressions gives the amount of movement the camera has experienced between two frames. In our actual application, we are interested only in estimating *rotations*. Given a total of K frames, the camera's pose is compounded as follows:

$$\mathbf{H}_1^{\mathrm{K}} = \mathbf{H}_1^2\mathbf{H}_2^3\ldots\mathbf{H}_{\mathrm{K}-2}^{\mathrm{K}-1}\mathbf{H}_{\mathrm{K}-1}^{\mathrm{K}} \tag{10}$$

V. RESULTS

Work has been done in its entirety using C++ as the programming language of choice. To achieve the best results, the *GNU Scientific Library* has been used to optimize the computation of crucial elements that have to be recomputed several times during the plane extraction process, such as *eigenvalues*, *eigenvectors*, as well as the *Levenberg-Marquardt* non-Linear Optimization.

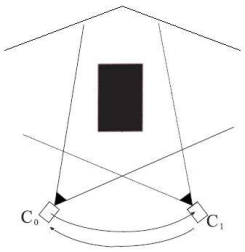
Table I shows some results illustrating the computational costs of each of the stages involved in the complete plane extraction process. Times are the average of the time taken

per frame. The dataset used to benchmark the results was obtained from the experimental scenario shown in Fig. 11a, consisting of a sequence of 670 frames (more details ahead). In the sequence, between five to six main planes are normally seen. Some snapshots of the performed movement (as segmented images), are shown in Fig. 12. Notice in the table listed individually the two segmentation modalities: the one consisting of a *full plane segmentation*, and of an *incremental segmentation*.

TABLE I: Average Computational Times for the Plane Extraction Process considering the sequence of the Fig. 12

COMPUTATIONAL SPEEDS	
Segmentation Stages	**Average Time [ms.]**
Cloud of 3D Points	8.6
Seeds Distribution	1.2
Full Segmentation	61.3
Incremental Segmentation	47.6
Pose Estimation	0.3

Before embedding the camera into the flying vehicle, the experimental scenario shown in Fig. 11a has been considered. The camera was rotated back-and-forth approximately 90° while tilted 45° around a square-shaped object positioned in front of two meeting walls. The ground-truth was provided by a highly accurate *Inertial Measurement Unit* Xsens MTi attached to the back of the camera and measuring its 3D orientation in parallel (Fig. 11b).



(a) The camera rotates through positions C_0–C_1–C_0

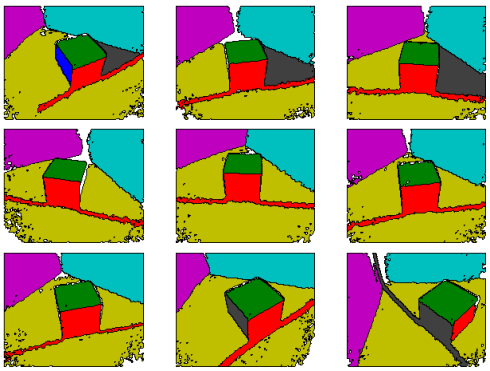
(b) XSens MTi

Fig. 11: Experimental Scenario and Ground-truth

Fig. 13 shows the results of this experiment. In Fig. 13a the Euler Angles provided by the MTi are directly compared with those calculated by plane-based Visual Odometry. The error in the three angles remains always below 10°. A better way to compare rotations is by measuring a single angle between ground-truth and calculated orientations at each frame. This is done converting rotations to *quaternions* (Fig. 13b). In this representation is noted that the maximum reached angle between both orientations is of 12.2°.

A. Experiments with the UAV

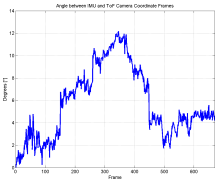
Two datasets were acquired with the SwissRanger SR-3000 used on-board a micro-UAV recently developed at



(a)

Fig. 12: A few snapshots illustrating the movement of the ToF camera during image acquisition. The IMU was attached in the back of the camera simultaneoously acquiring data. Images shown are the output of the segmentation and tracking algorithm.

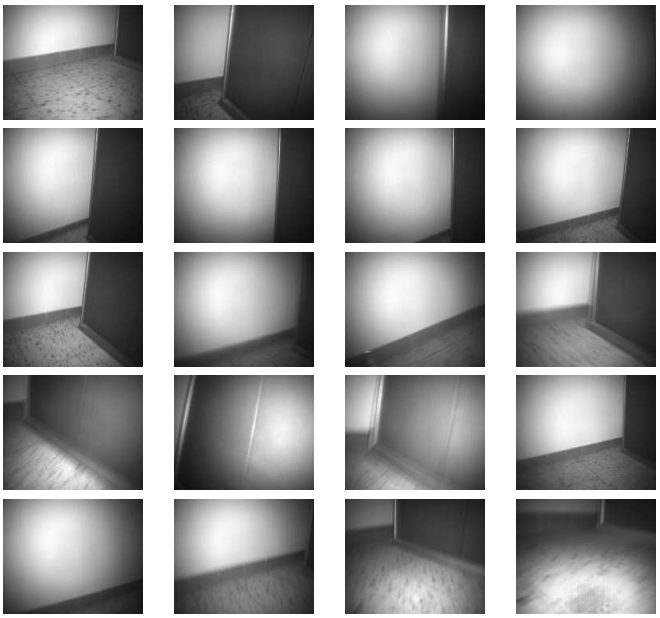
(a) Direct Euler Angles comparison



(b) Angle between Rotations in Quaternion Representation

Fig. 13: Results of the ToF camera orientation estimates

ENAC (as shown in Fig. 1). These datasets however did not turn out to be very usable, as the flight conditions were not the most appropriate during tests. First, the dynamics of the UAV have to be adapted in order to reduce the blurring in the images, and hence increase their quality. Second, as we use only the geometry in the TOF camera data to localize the UAV at the moment, the observed scene has to contain at least three non-parallel planes to estimate the odometry from the ToF camera only. This requirement was easy to fulfill with the previous sequence shown in Fig. 12, but it is not always satisfied in the sequence acquired with the UAV. For this reason, the odometry fails from time to time, when not enough planes are visible. Due to time constraints, no more datasets could be obtained to perform more experiments. Nevertheless, the results obtained so far are encouraging, and have demonstrated the feasibility of the camera for localization purposes. In Fig. 14 a few snapshots of a dataset captured with the camera mounted in the UAV are shown for reference.



(a)

Fig. 14: Some frames belonging to a dataset created using the ToF camera on-board the flying micro-UAV.

These first experiences have given us valuable knowledge we lacked before beginning the experiments, that will allow us to better perform experiments in future, by considering aspects such as correct camera positioning, orientation, vibrations induced by the UAV, among others.

VI. FUTURE WORK

While future work obvioulsy includes deriving the full odometry (meaning the computation of *Translation* estimates) it will also try to complement the investigations made so far in order to improve our results, and avoid already identified problems. Three approaches will be investigated in future. First, textures in the environment that can be seen in the intensity images grabbed by the ToF camera will be used to aid the visual odometry; secondly, such results will be fused with information obtained from the Inertial Measurement System; finally, localization will be made using a previously known 3D model of the environment in which the UAV has to move. These investigations will allow us to achieve the autonomous navigation and localization of the flying vehicle in a better and more reliable way.

VII. CONCLUSIONS

This document has outlined our first experiments towards the development of a fully autonomous indoor flying vehicle. We believe that such system has interesting applications, such as in the fields of *rescue* and *service* robotics, reaching areas where sensors such as *Global Positioning Systems* would fail. In principle we have focused our investigations in the use of *planar patches*, given the characteristics of indoor environments. A real-time algorithm has been implemented, that segments a *depth image* into the planes found within it. *Tracking* and *registration* have been made possible thanks to the matricial nature of the ToF images, using an approach based on the morphological operation *erosion*. Using the *normal vectors* of corresponding planes in two consecutive frames, *ego-motion* is estimated. *Rotation* estimates have proven to be accurately estimated. These investigations have proven the ToF camera suitable for its use on-board a flying vehicle, and to autonomously perform tasks such as *localization* and *mapping*.

REFERENCES

[1] P. Brisset, A. Drouin, M. Gorraz, P.S. Huard and J. Tyler, "The Paparazzi Solution", http://paparazzi.enac.fr

[2] J. Poppinga, A. Birk and K. Pathak. "A Characterization of 3D Sensors for Mobile Robots". In *RoboCup 2009: Robot Soccer World Cup XIII*, volume 5949/2010 of *Lecture Notes in Computer Science*, pages 264–275. Springer Berlin / Heidelberg, February 2010.

[3] J. W. Weingarten, G. Grüner, & R. Siegwart, "A state-of-the-art 3D sensor for robot navigation", In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, September 2004.

[4] S. May, W. Bjorn, S. Hartmut and P. Kai. "3D Time-of-Flight Cameras for Mobile Robotics". In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 790–795, October 2006.

[5] K. Ohno, T. Nomura, and S. Tadokoro. "Real-Time Robot Trajectory Estimation and 3D Map Construction using 3D Camera". In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5279–5285, October 2006.

[6] R. Sheh, M. W. Kadous, and C. Sammut. "On Building 3D Maps Using a Range Camera: Applications to Rescue Robotics". Technical Report UNSW-CSE-TR0609, University of New South Wales, 2006.

[7] S. May, D. Dröschel, D. Holz, C. Wiesen, and S. Fuchs. "3D Pose Estimation and Mapping with Time-of-Flight Cameras". In *IEEE/RSJ International Conference on Intelligent Robots and Systems, Workshop on 3D Mapping*, Nice, France, October 2008.

[8] S. May, S. Fuchs, D. Dröschel, D. Holz, and A. Nüchter. "Robust 3D-Mapping with Time-of-Flight Cameras". In *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1673–1678, Piscataway, NJ, USA, October 2009. IEEE Press.

[9] S. Robbins, B. Murawski, and B. Schroeder. "Photogrammetric Calibration and Colorization of the SwissRanger SR-3100 3-D Range Imaging Sensor". *Optical Engineering*, 48(5), 2009.

[10] M. Lindner and A. Kolb. "Lateral and Depth Calibration of PMD-Distance Sensors". In *Advances in Visual Computing*, volume 2, pages 524–533. Springer, 2006.

[11] T. Kahlmann, F. Remondino and H. Ingensand. "Calibration for Increased Accuracy of the Range Imaging Camera SwissRanger". In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 5, pages 136–141, September 2006.

[12] S. Fuchs and S. May. "Calibration and Registration for Precise Surface Reconstruction with ToF Cameras". *International Journal of Intelligent Systems Technologies and Applications*, 5(3/4):274–284, November 2008.

[13] S. Fuchs and G. Hirzinger. "Extrinsic and Depth Calibration of ToF-Cameras". In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2008.

[14] J. Mure-Dubois and H. Hügli. "Optimized Scattering Compensation for Time-of-Flight Camera". In *Two- and Three-Dimensional Methods for Inspection and Metrology V (Proceedings of SPIE)*, volume 6762-0H, September 2007.

[15] J. Mure-Dubois and H. Hügli. "Real-Time Scattering Compensation for Time-of-Flight Camera". In *Proceedings of the ICVS Workshop on Camera Calibration Methods for Computer Vision Systems*, March 2007.

[16] J. Poppinga and A. Birk. "A Novel Approach to Efficient Error Correction for the Swiss-Ranger Time-of-Flight 3D Camera". In *RoboCup 2008: Robot Soccer World Cup XII*, volume 5399/2009 of *Lecture Notes in Computer Science*, pages 247–258, Berlin, Heidelberg, June 2009. Springer Berlin / Heidelberg.

[17] R. Lange. “3D Time-of-Flight Distance Measurement with Custom Solid-State Image Sensors in CMOS/CCD-Technology”. PhD thesis, University of Siegen, 2000.

[18] J. Weingarten and G. Grüner. “A Fast and Robust 3D Feature Extraction Algorithm for Structured Environment Reconstruction”. In *Proceedings of the 11th International Conference on Advanced Robotics*, 2003.

[19] J. Weingarten, G. Grüner, and R. Siegwart. “Probabilistic Plane Fitting in 3D and an Application to Robotic Mapping”. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 927–932 Vol.1, 26 2004.

[20] N. Vaskevicius, A. Birk, K. Pathak, and J. Poppinga. “Fast Detection of Polygons in 3D Point Clouds from Noise-Prone Range Sensors”. In *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics*, pages 1–6, 27-29 2007.

[21] J. Poppinga, N. Vaskevicius, A. Birk, and K. Pathak. “Fast Plane Detection and Polygonization in Noisy 3D Range Images”. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3378–3383, 22-26 2008.

[22] D. Dróschel, S. May, D. Holz, P. Plóger, and S. Behnke. “Robust Ego-Motion Estimation with ToF Cameras”. In *Proceedings of the 4th European Conference on Mobile Robots (ECMR 2009)*, pages 187–192, September 2009.

[23] J. Weingarten. “Feature-Based 3D SLAM”. PhD thesis, École Polytechnique Fédérale de Lausanne, September 2006.

[24] K. Pathak, N. Vaskevicius, J. Poppinga, M. Pfingsthorn, S. Schwertfeger, and A. Birk. “Fast 3D Mapping by Matching Planes Extracted from Range Sensor Point-Clouds”. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* , pages 1150–1155, 10-15 2009.

[25] K. Pathak, A. Birk, N. Vaskevicius, and J. Poppinga. “Fast Registration Based on Noisy Planes With Unknown Correspondences for 3-D Mapping”. *IEEE Transactions on Robotics*, PP(99):1–18, 2010.

[26] R. Lange and P. Seitz. “Solid-State Time-of-Flight Range Camera”, *IEEE Journal of Quantum Electronics*, volume 37, number 3, pages 390–397, March 2001.

[27] P. Khongsab. “Signal Processing and Performance Evaluation of a PMD Camera for Space Docking”, *Master’s Thesis, LuleåUniversity of Technology*, September 2009.

[28] R. L. Graham. “An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set”. *Information Processing Letters*, 1(4):132–133, 1972.