

# JOINT VISUAL AND TIME-OF-FLIGHT CAMERA CALIBRATION FOR AN AUTOMATIC PROCEDURE IN SPACE

Teresa Conceição<sup>1</sup>, Brian Coltin<sup>2</sup>, Andrew Symington<sup>2</sup>, Lorenzo Fluckiger<sup>2</sup>, and Trey Smith<sup>3</sup>

<sup>1</sup> *Faculty of Engineering of University of Porto, Porto, Portugal*

<sup>2</sup> *SGT Inc., NASA Ames Research Center, Moffett Field, CA*

<sup>3</sup> *NASA, NASA Ames Research Center, Moffett Field, CA*

Emails: [teresa.psconc@gmail.com](mailto:teresa.psconc@gmail.com), [brian.j.coltin@nasa.gov](mailto:brian.j.coltin@nasa.gov), [andrew.c.symington@nasa.gov](mailto:andrew.c.symington@nasa.gov),  
[lorenzo.fluckiger@nasa.gov](mailto:lorenzo.fluckiger@nasa.gov), [trey.smith@nasa.gov](mailto:trey.smith@nasa.gov)

## ABSTRACT

Astrobee is a freeflyer robot that will soon board the International Space Station (ISS). It senses with two pairs of color and Time-of-Flight (ToF) depth cameras, whose calibration is critical for mapping, localization, obstacle detection, and more. Even with an initial on the ground calibration, on-orbit recalibration may be required due to hardware changes from shipping, launch, environmental changes, and for future hardware replacements. The astronauts time is limited and expensive, therefore, calibration should not only be accurate and robust but also run in space with minimal human intervention. Hence, we present an automatic intrinsics and extrinsics calibration for the depth and visual cameras of Astrobee. We thoroughly evaluate our procedure and demonstrate it is ready for use on the ISS.

## 1. INTRODUCTION

Depth and visual multi-camera systems have increased in popularity over the last few decades. Manipulation tasks [1, 2], mapping [3], navigation and path planning [4] are some of the tasks that can benefit from a multi-camera system. Monocular systems can address a wide variety of tasks such as recognition, while depth cameras can provide real-time absolute metric information; however, their resolution is insufficient to detect fine details of the scene. Hence, combining both cameras is useful for many robotic applications. Alternatively, stereo visual cameras can also provide metric information, but introduces some ambiguity to the measurements due to challenges in resolving the depth of low-texture scenes. Integration and data fusion of visual and depth cameras inherently requires an accurate intrinsics and extrinsics calibration, which presents some challenges.

Although camera calibration has been a widely explored problem, the fusion of depth and color cameras is still an ongoing research topic. Most traditional methods are not well suited to deal with different types of sensors nor can they cope with the lower resolution of the ToF cameras. Different approaches have been proposed in the past, but most require extensive manual intervention (e.g., complex

setups [5] or manual selection of features in the depth image [6]) and assume the relative pose between cameras (the extrinsics calibration) is easily solved by standard procedures. However, due to the two cameras considerably different fields of view (FOV), challenges arise in acquiring suitable calibration data, leading to unstable parameter estimates and undesirable highly correlated intrinsic and extrinsic camera parameters [7].

To overcome these issues, we extend *Kalibr*, an open source calibration tool, to support less constrained calibration targets like the one available to Astrobee on the ISS. This is especially important considering the FOV issues, but also given ISS storage space limitations. Astrobee has a docking station for autonomous battery recharging that was specifically designed to have fiducial markers fully visible at all ranges as Astrobee docks and undocks. By re-using it as a calibration target, no extra physical items need to be sent to space. We apply a standard camera calibration procedure [8] by taking advantage of the amplitude images of a ToF camera and test their ability to provide accurate intrinsics and extrinsics calibration between color and ToF cameras. Furthermore, the vision system is integrated into the robot by calibrating the extrinsics of the cameras relative to the robots IMU. Contrary to other state of art methods, our approach has a good level of autonomy and is compatible with ROS.

## 2. CONTEXT AND MOTIVATION

Astrobee is a freeflyer robot developed by the NASA Ames Intelligent Robotics Group (IRG) that will replace the previous generation of freeflyer robots SPHERES in the International Space Station (ISS). This compact robot's main goals are to help scientists and researchers run a variety of experiments in zero-gravity, as well as freeing up astronauts' time spent on tedious housekeeping jobs. It is also intended to serve as additional eyes for flight controllers. To complete all these tasks autonomously, Astrobee relies on its vision system which has, among others, two pairs of depth Time-of-Flight and color cameras (Fig. 1). The *NavCam* and *DockCam* are monocular RGB cameras with 1280x960 resolution (1.2 megapixel) and fields of view (FOV) of respectively 120° and 90°. The forward-facing *NavCam* is used for map-

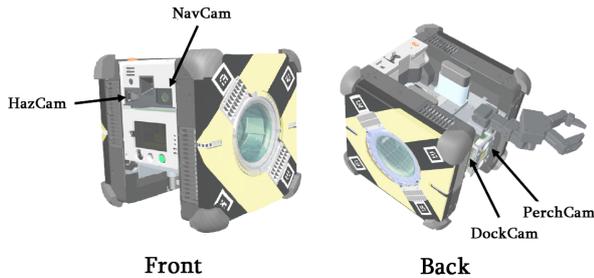


Figure 1. : Astrobee Camera Pairs

ping and navigation whereas the rear-facing DockCam helps to improve pose estimation while docking by detecting fiducials on Astrobee’s dock. As for the depth cameras, both are CamBoard Pico Flexx time-of-flight flash LIDAR depth sensors with lower FOV ( $62^\circ$ ) and resolution ( $224 \times 172$ ), and an effective range of 0.1-4 m at 5 frames per second (fps) (Maximum acquisition is 45 fps but the range is reduced to 1m). The rear-facing *PerchCam* detects and shapes ISS handrails essentially for the perching procedure and the front-facing *HazCam* detects obstacles working alongside the *NavCam* for navigating. Calibrating and integrating all of the cameras is critical to the system functioning correctly and comes as our primary motivation for this work.

The scripts developed as well as all Astrobees flight software are available open source and can be found at [9].

### 3. RELATED WORK

Most of the previous works handle ToF camera calibration individually, focusing on the depth measurements correction. By having calibration data with different integration times and targets with different reflectivities, Kahlmann accounts for distinct sources of errors [10]. Deviations are modeled by comparing the measurements with an external highly accurate ground-truth track and building a look-up table (LUT). Furthermore, the use of a calibration target composed of LEDs is proposed to facilitate its detection in the intensity images. A similar approach is presented in [11] but depth deviations are corrected by linear interpolation of a B-Spline instead. They also proposed the use of an RGB camera with a fixed known relative position to measure the depth ground-truth. Regarding the actual intrinsics and extrinsics parameters estimation, a standard camera calibration method proposed by [8] is generally employed. It consists of moving a planar checkerboard target with different views and orientations in front of the camera/s. After that, homography geometry and point correspondences are used to estimate the desired parameters. Zhang and Zhang [6] simultaneously calibrate a color and a depth camera pair by identifying the checkerboard corners of the calibration target in the color camera images and exploring the planarity of the target in the depth image. In [12], Herrera enhances Zhang’s work by also ac-

counting for depth distortion calibration. Although functional, these works require manual selection of checkerboard corners for initialization of the depth camera parameters, lacking some automatism.

Contrary to the depth image, an amplitude image provides a greyscale visualization of the target, partially resolving the corner identification problem. The idea of using it for calibration purposes has been proposed several times in the literature. Fuchs solves both intrinsics, extrinsics and depth calibration with amplitude images but uses a robotic arm to provide accurate groundtruth [13]. Fusion of ToF cameras with high-resolution cameras is also commonly applied to improve the depth camera calibration, removing the need for complex setups [14, 15, 11]. Additionally, in order to have a good calibration, one should try to cover as many pixel locations as possible by the target but at the same time have an overlapping FOV for both cameras to retrieve the extrinsics. Specifically in the case of a common checkerboard, having to be completely shown in a frame to be recognized, appears as a constraint to the recording of a good calibration dataset. Lindner solves this by having a combined rig of other CCD-cameras [7], which may not always come in hand.

Despite presenting major contributions to the area, all the mentioned works use the HD color camera as an auxiliary way of providing a ground truth to improve the ToF calibration, focusing more on the different depth error corrections. They assume that provided with the visual images, the calibration between them is accurately solved by a standard procedure. However, as noted by Lindner in [7], calibrating the ToF camera alone or even in a pair with just one color camera can lead to unstable pose estimates. The internal and external camera parameters revealed to have higher correlations for the latter cases mainly due to the different field of views.

### 4. TOF DEPTH CAMERAS

Time-Of-Flight depth cameras have been taking the place of more tradition depth sensors such as 2D range sensors or sonars. In particular, the ability to provide real-time 3D visual information, unlike other higher cost and slower processing sensors, presents a great advantage in many applications. Nevertheless, their range measurements are usually less accurate and the lower resolution of their amplitude images can be a considerable constraint. They make observations by illuminating the scene with an Infrared (IR) light signal and calculating the time between its emission and the reception of its reflection. There are two common variations: either directly measuring a pulsed light time of flight on each pixel or modulating continuously light waves through an RF carrier and calculating its phase shift upon reception. More specifically, Astrobee’s ToF cameras are Photonic Mixer Devices (PMD) which implement the latter case.

A ToF camera usually provides a 3D point cloud along with a depth image and an amplitude image (Fig. 2) The

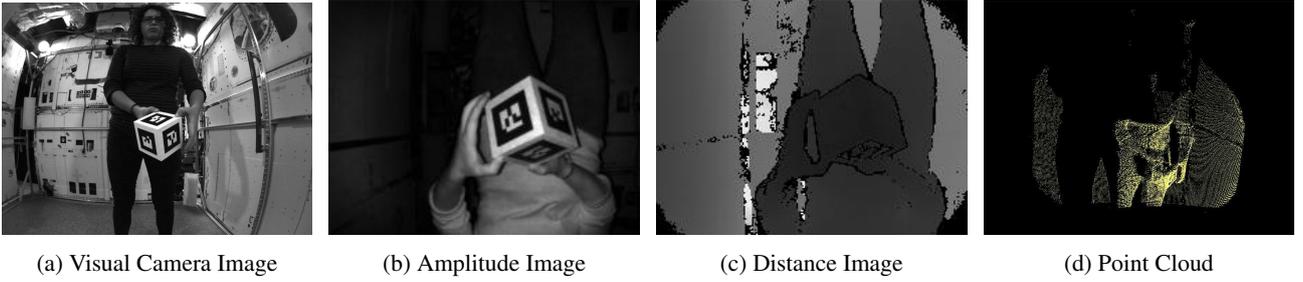


Figure 2. : Different data provided from a camera pair composed of visual camera (a) and a TOF depth camera (b), (c), (d)

range image represents the measured distance from each pixel to its corresponding projected point in the scene, whereas the amplitude image is a quantification of the reflected signal strength. Some cameras, such as Astrobees, may also deliver an intensity image which represents the mean of total light captured by the sensor, including the background light and the reflected IR signal.

## 5. CAMERA MODELS

Camera calibration is essentially the process of estimating the parameters that can transform a real 3D scene into its corresponding 2D camera image. To estimate these parameters we use the most common Pinhole Model to represent the given projection as depicted in Equation 1 and illustrated in Fig. 3.

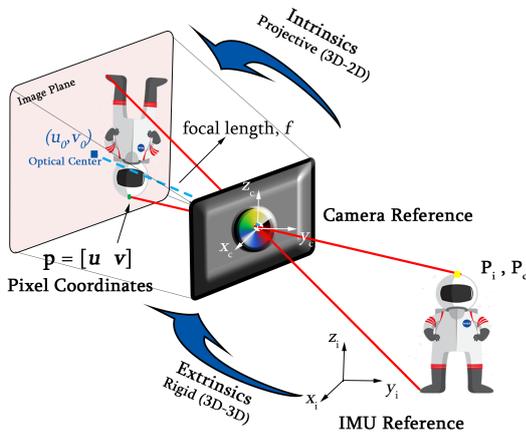


Figure 3. : Projection Pinhole Model

The parameters include the focal Length  $f$  and optical center  $(u_0, v_0)$ , each describing a geometric propriety intrinsic to the camera, the reason for which the estimation of these parameters is called Intrinsics Calibration. The focal length  $f$  represents the distance from the "pinhole", i.e the distance from the center of the camera and the image plane. It is divided in  $f_x$  and  $f_y$  which ideally have the

same value although they may slightly differ in practice. The optical center, on the other hand, is the projection of the camera center (the pinhole) on the image plane. Both are included in  $K$ , the camera matrix (Equation 2), which is used to project camera reference points into their correspondent pixel coordinates in the camera image plane. This transformation is invariant to camera dimensions as only the relative relation between camera coordinates and image coordinates are of interest, not the camera's absolute dimensions. For this reason, both focal length and optical center are represented in pixels. An axis skew coefficient is sometimes also added to the intrinsics parameters but for a true pinhole model, we can consider it as zero.

Furthermore, we still need to relate the camera's location and direction to the world (in this case we consider the world as being the robot, i.e. the IMU frame of reference), the extrinsics, so a coordinates transformation given by rotation  $R$  and translation  $t$  is defined.

$$p_{undistorted} = K \begin{bmatrix} c_w R & c_w t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_w \\ 1 \end{bmatrix} \quad (1)$$

$$K = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

### 5.1. Distortion

The ideal pinhole model does not account for lens distortion, which is not representative of real cameras. Thus, the camera model to be calibrated needs to include distortion coefficients as well.

Different distortion models can be defined depending on the type of lens. The most common and the one used for our ToF depth cameras, is the Radial-Tangential (Eq. 3) proposed by Brown [16], where  $(x_u, y_u)$  and  $(x_d, y_d)$  are the undistorted and distorted points respectively. The model describes radial distortion- an approximately radially symmetric effect caused by different light rays behaviors as incident points get away from the optical center- and tangential distortion, caused by the nonparallelism between the lens and the image plane. The degree of the radial and tangential distortion polynomial is set according to the estimated level of distortion, but usually, a

second-order polynomial with two tangential coefficients ( $t_1, t_2$ ) and two radial coefficients ( $k_1, k_2$ ) is sufficient.

$$x_u = x_d + \underbrace{\tilde{x}(k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots)}_{\text{Radial Distortion}} + \underbrace{t_1(r^2 + 2\tilde{x}^2) + 2t_2\tilde{x}\tilde{y}(1 + t_3 r^2 + \dots)}_{\text{Tangential Distortion}} \quad (3)$$

$$y_u = y_d + \tilde{y}(k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) + t_2(r^2 + 2\tilde{y}^2) + 2t_1\tilde{x}\tilde{y}(1 + t_3 r^2 + \dots)$$

$$\tilde{x} = x_d - u_0 \quad \tilde{y} = y_d - v_0 \quad (4)$$

$$r = \sqrt{\tilde{x}^2 + \tilde{y}^2} \quad (5)$$

For the visual cameras, a different distortion model, the Field-Of-View (FOV) model (Eq. 6) proposed by Devernay and Faugeras [17] was chosen. The FOV model is more appropriate for wide-lens cameras and their nonlinear distortion. It only includes one parameter, the FOV angle  $\omega$  of the ideal fish-eye lens, which in practice may not be the exact camera's FOV.

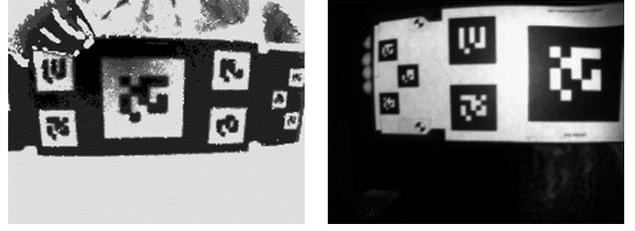
$$x_u = x_d \frac{\tan(rd\omega)}{2 \tan(\frac{\omega}{2})} \frac{1}{rd} \quad y_u = y_d \frac{\tan(rd\omega)}{2 \tan(\frac{\omega}{2})} \frac{1}{rd} \quad (6)$$

$$rd = \sqrt{x_d^2 + y_d^2} \quad (7)$$

## 6. PROCEDURE

As previously stated, several methods use the depth image (Fig. 4a) to calibrate the depth cameras. However, despite providing extremely useful information on the relative distance of scene objects, they are usually very noisy. Hence, unless used along with specifically designed calibration targets with, for instance, features at different depths, it can be challenging to find sufficient correspondences between images to accurately calibrate a depth camera. Furthermore, for the extrinsics calibration between two cameras in a pair (visual and depth camera), this correspondence is even harder given the cameras distinct image characteristics. To overcome this, the amplitude image of the ToF camera (Fig. 4b) is used where features of a normal planar calibration target can be distinguished and therefore we can treat the depth camera as a normal visual one (or as a "bad quality visual camera").

The calibration procedure is based on the one proposed by Zhang [8] and consists of inferring the intrinsics and extrinsics from different perspectives of a planar calibration target whose 3D geometry and specifications are well known. Ideally, the process would go through a single round of data acquisition followed by the calibration computation and optimization, however, an accurate and robust process of camera calibration and its integration with



(a) Depth image

(b) Amplitude Image

Figure 4. : Comparison between ToF camera Amplitude and Depth Images

the IMU, requires in fact, very distinctive datasets. For intrinsics and extrinsics estimation between cameras, images should be recorded with slow movements in order to reduce motion blur. This comes with an increased importance due to the lower resolution of the depth cameras for which the effects of motion blur can lead to worse markers recognition and even calibration divergence. On the other hand, correlation of camera images with IMU movements can only be correctly achieved with the excitement of all IMU axis and with meaningful velocity and acceleration information, therefore faster and shakier movements are necessary. As it is evidenced in images of Fig. 5, taken while rapidly moving the robot, motion blur effects are substantially worse for the depth camera. Because of that, in this case, we only considered the higher resolution camera for the IMU-Cameras extrinsics calibration. The relative pose between the depth cameras and the IMU can be inferred from the previous camera-camera extrinsics estimation.

Overall, our approach divides the procedure into two main steps summarized in Table 1.



(a) Nav Cam

(b) Haz Cam

Figure 5. : Camera Images Comparison for IMU calibration dataset at the same timestamp. Motion Blur effect is definitely clearer on the depth camera amplitude hampering an accurate target detection

Finally, it should also be noted that the calibration accuracy/efficiency trade-off is highly related to the number of dataset images and different target perspectives captured. Despite the importance of this fact, we were more interested in having a very accurate calibration rather than an extremely fast one, so minimizing the number of images used and consequently the runtime was not a main point of focus.

**Table 1.** : Calibration procedures description

Procedure	Objective	Recorded Data	Description
1	Calibrate Intrinsic and Extrinsic Camera pairs	Visual camera grayscale images Depth camera amplitude images	Slow movements (reduce motion blur)
2	Calibrate Extrinsic Cameras - IMU	Visual camera grayscale images Depth camera amplitude images IMU data	Fast and shaking movements (excite all IMU axis)

## 6.1. The Target

Regarding the calibration target, Astrobees dock already has fiducial markers in a specific design pattern to enable autonomously docking/undocking operations. By reusing it as a calibration target, we not only avoid shipment of extra equipment to the current overcrowded ISS but also make use of the already high reflectance dock's material. This is particularly important to easily detect the markers by differences between black and white in the amplitude image. Contrary to checkerboards or circle grids, fiducial based targets can automatically detect orientation given its self-identifying binary codes. Most standardized methods and software tools only support checkerboard detection which imply that the whole target needs to be within the camera image for determining its orientation. Considering the quite distinct cameras FoV and resolutions, it can be extremely difficult to have in both cameras a clear view of the whole target at the same time, therefore the use of these markers (also known as AR markers) is definitely a big advantage in our case.

## 6.2. Software and Main Algorithms

For the actual estimation problem, *Kalibr* [18], an open-source software tool, was chosen due to its several different features:

- Flexibility: supports 3 types of targets, 3 types of distortion models and 2 types of projection models
- Robust feature extraction and optimization algorithms
- Outputs extrinsics related to IMU measurements (which most calibration toolboxes don't)
- Provides error metrics and respective data visualization
- ROS friendly and automatic

Following the data acquisition previously depicted, the ROS Bags and configuration files (with cameras and IMU characteristics and target description) were inputted into the software. Its calibration can be briefly summarized by algorithms 1 and 2.

---

### Algorithm 1 Camera Pair Intrinsic and Extrinsic Calibration Algorithm

---

**For each camera in a camera pair**

- 1: Images pre-processing (gaussian filtering applied for smoother image analysis)
- 2: Detect markers in a image set and compare with expected target.
- 3: Remove images without successful detections
- 4: Initialize camera parameters and solve bundle adjustment for each camera individually

**After each camera initialized:**

- 5: Build correspondence graph between cameras observations based on simple timestamps approximation
  - 6: Initialize extrinsics parameters
  - 7: Solve bundle adjustment for intrinsic and extrinsic
  - 8: Delete outliers
- 

For IMU calibration, only the visual camera was considered given its better resolution and consequently better resistance to motion blur. The transformations from the IMU reference to the depth cameras' were inferred from the previously determined camera pair extrinsics.

---

### Algorithm 2 Camera-IMU Extrinsic Calibration Algorithm

---

**For each camera in a camera pair**

- 1: Lines 1 - 3 from Algorithm 1
  - 2: Splines computation for IMU data points interpolation. This was made considering previous information on IMU characteristics and has the possibility of inputting a gravity estimation, which is useful for space applications.
  - 3: Time calibration between camera and IMU.
  - 4: Initialize extrinsics parameters
  - 5: Solve bundle adjustment for extrinsics. If desired, re-computation of intrinsic and extrinsic between cameras is also possible.
  - 6: Delete outliers
- 

The bundle adjustments were achieved through a Levenberg-Marquardt trust region optimizer solved by a Sparse Cholesky Linear Solver. For more details, we refer to [18].

Finally, it is important to note that, despite being a very robust calibration tool, *Kalibr* is not built to deal with lower resolution cameras neither it supports a more freely disposed target as ours. Thus, part of the work was concen-

trated on increasing the flexibility of the software.

As matter of fact, the algorithm does by default a sub-pixel refinement in which it compares its own fiducial detection with an OpenCV one. If the difference between the two is greater than a specific threshold, the observation is considered as faulty. However, for a lower resolution camera as the ToF one, this presents a big problem: a lot of observations were being removed and the calibration wasn't converging due to lack of correspondences.

Furthermore, for a robust initialization of parameters, only the frames where the target board view was complete were being used. Again, in depth cameras' images, the detection of all the markers at the same time was never possible mostly because of its lower resolution and field of view. The smaller markers could only be detected if they were close to the camera and therefore the whole target would not fit in the image. Hence, not only the sub-pixel refinement feature was removed but parameter initialization was also relaxed.

As far as the target is concern, *Kalibr* only supports AR markers symmetrically organized in grids. However, that's not the case of our dock target so support for a more flexible target was added in order to have a successful calibration (most of the changes were on lines 1 - 4 of 1).

## 7. EXPERIMENTAL TESTS AND RESULTS

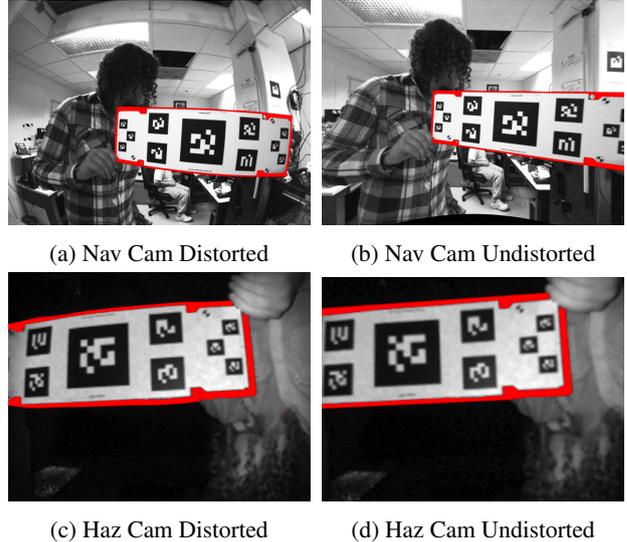
The final procedure was implemented and further authenticated through different tests in a frictionless surface lab (to replicate space dynamics) as well as inputted in Astrobe's mapping and navigation procedures to verify its reliability.

The data acquisition for procedure 1 (explained in Table 1) was done by moving the target in front of the cameras with different poses and different distances trying to cover all pixels of both cameras images and at the same time having overlapping FOV's so we can have a good amount of correspondences. The RGB cameras' images were recorded in grayscale mode with at 15Hz while the depth camera data was acquired at 5Hz (maximum acquisition frequency for best performance).

On the other hand, for procedure 2, since all IMU axis need to be well excited for movement detection, the target was now static and the robot detached from its base. Although we can program the robot to move and record the data autonomously, given the required rapid and shaky movements, we moved the robot by hand for a safer procedure. IMU data was recorded at 250Hz whereas visual cameras and depth cameras had a frame rate of 15Hz and 5Hz respectively. As previously explained in section 6, despite still recorded (for possible recalibration or verification), depth cameras images were not considered for calibrating the extrinsics to the IMU due to its lower resolution and acquisition frame rate.

**Table 2.** : Camera pairs calibration Reprojection Errors

Camera	Reprojection Error [pixel]	
	$\mu$	$\sigma$
Nav Cam	0	0.431
Haz Cam	0	0.639
Dock Cam	0	0.320
Perch Cam	0	0.548



*Figure 6.* : Camera images before and after undistortion given by camera pair calibration. Target lines on original images (6a, 6c) were corrected to straight lines after undistortion (6b , 6d)

The resulting projection errors and respective standard deviation were quite satisfactory as demonstrated by Table 2. The results showed no systematic error (the mean is approximately 0), with its maximum value being no greater than 1.5 pixels (Fig. 7). Moreover, one can see that generally, markers were detected on almost the entire depth cameras' FOV while for the RGB cameras there is a lack of observations on the corners mainly due to its fish-eye type lens. Theoretically, this can have implications on accurate treatment of points in those areas but it shouldn't be a major problem in practice.

In addition to this, visual inspection also served as quality analysis for the first part of the procedure. Rectified images were computed for verification of the extrinsics between cameras (Fig. 9). Distorted and undistorted image comparison measured the quality of distortion coefficients estimation (Fig. 6).

Relatively to the cameras-IMU calibration, as expected due to the less stable dataset, reprojection errors mean and standard deviation increased significantly but still representing very decent metrics (Table 3). Although most of the computed errors remained low, some higher values, up to approximately 12 pixels, were observed as demonstrated by Fig. 8. In this case, it appears that the result may considerably depend on the quality of the data acquired.

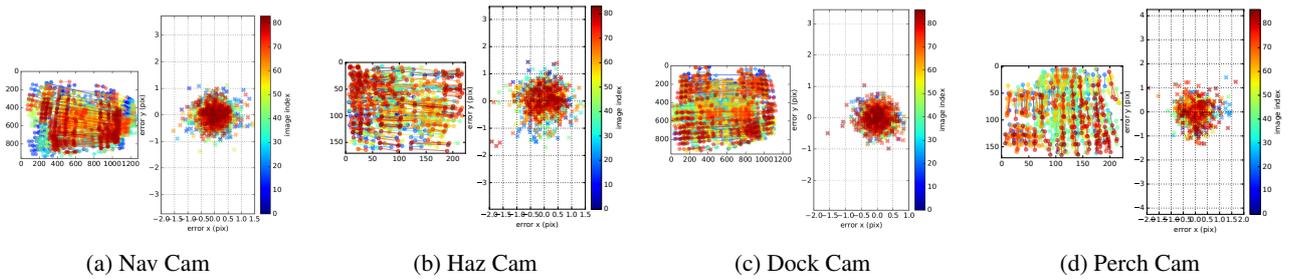


Figure 7. : Reprojection error distribution of camera pairs calibration. The square on the left in each image represents the cameras FOV with all the pixels with successful markers detection. The grid on the right displays the error distribution in pixels. The color bar exposes the errors timing information, with dark blue being the first images on the dataset and dark red the last ones

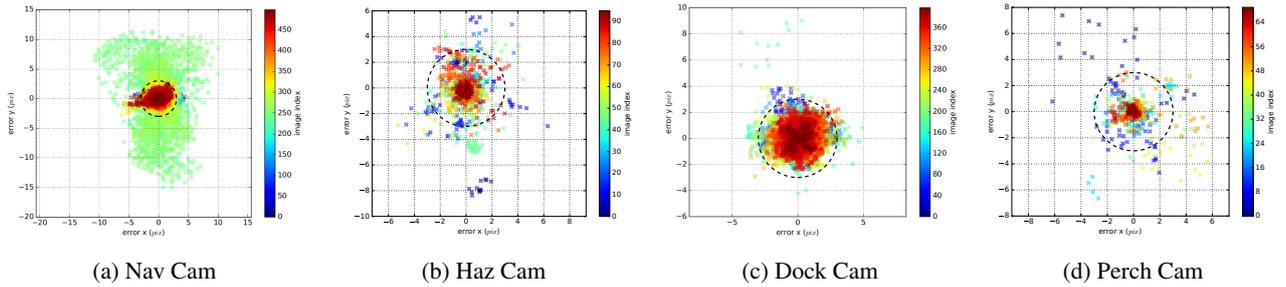


Figure 8. : IMU extrinsics calibration reprojection error distribution.

For instance, the *Nav-Haz* camera pair performed slightly worse, especially at around the middle of the dataset (images indexes 200-250 represented as light green dots). As the camera-IMU calibration uses the previous intrinsics and extrinsics results, this possible has to do with the fact that those frames had markers observation on the image corners pixels, which might not have been observed in the intrinsics calibration.

Moreover, by inspecting the ToF cameras point clouds and resulting frame of references in Fig. 11 and 10 one can confirm the plausibility of that the results.

Besides the camera intrinsics and extrinsics, ToF cameras usually present systematic and non-systematic errors regarding its depth values which have been subject of several studies in the past years. However, in most of the cases, the resulting depth clouds and depth images after camera firmware internal processing (it already does its own depth correction) were quite satisfactory for our specific tasks so we leave a more extensive analysis on the matter for future work.

## 8. CONCLUSION AND FUTURE WORK

With this work, an effective and accurate way of calibrating and integrating a vision system for space applications was successfully developed. Contrary to other methods for joint depth and visual cameras calibration, our approach is flexible enough to cope with different cameras

characteristics and to be integrated as a considerably automatic robotic procedure (although not fully automatic). Through Astrobees teleoperation system, flight controllers or other scientists can supervise the dataset recordings, releasing astronauts from this task. Future work may reside in trying to make the procedure fully autonomous by developing automatic data acquisition as a path planning problem for recording the best views of the target for the calibration. Furthermore, at this moment the two camera pairs are being calibrated separately because they do not share overlapping fields of view. In the future, we wish to find a way of combining all the cameras and the IMU in one single procedure. By spreading fiducials all over Astrobees's workplace and additionally providing depth information, one can compute robust pose estimations and possibly get rid of the current target, creating one big bundle adjustment.

## REFERENCES

- [1] A. Saxena, L. L. S. Wong, and A. Y. Ng. Learning Grasp Strategies with Partial Shape Information.
- [2] U. Klank, D. Pangercic, R. B. Rusu, and M. Beetz. Real-time CAD model matching for mobile manipulation and grasping. In *2009 9th IEEE-RAS International Conference on Humanoid Robots*, pages 290–296. IEEE, 12 2009.
- [3] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments.

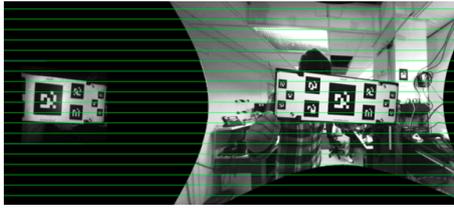


Figure 9. : Depth camera image rectified into visual camera frame of reference. Green lines show matching points on each image.

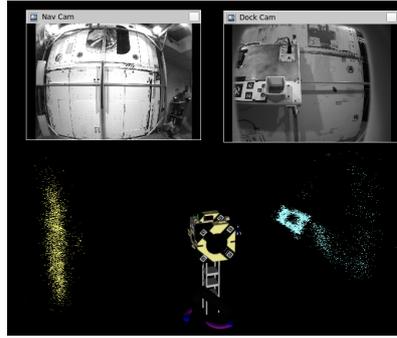


Figure 10. : Depth Point Clouds after applying relative transformation given by extrinsics. Clouds are correctly positioned accordingly to Astrobbee mesh and visual cameras images

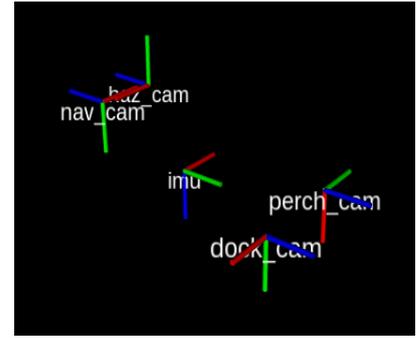


Figure 11. : Resulting frame of references after extrinsics calibration

**Table 3.** : Reprojection Errors and IMU related Errors of Cameras-IMU extrinsics calibration. The last column represents the time calibration results between IMU and Cameras.

Camera	Reprojection Error [pixel]		Gyroscope Error [rad/s]		Accelerometer Error [m/s <sup>2</sup> ]		Timeshift to IMU [s]
	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	
Nav Cam	1.555	1.974					-0.069
Haz Cam	1.439	1.378	0.001	0.001	0.043	0.565	-0.077
Dock Cam	0.946	0.647					-0.070
Perch Cam	0.634	1.513	0.001	0.001	0.0170	0.0165	-0.081

*The International Journal of Robotics Research*, 31(5):647–663, 4 2012.

- [4] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy. Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera. pages 235–252. Springer, Cham, 2017.
- [5] J. Jung, J.-Y. Lee, Y. Jeong, and I. S. Kweon. Time-of-Flight Sensor Calibration for a Color and Depth Camera Pair. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(7):1501–1513, 7 2015.
- [6] C. Zhang and Z. Zhang. Calibration Between Depth and Color Sensors for Commodity Depth Cameras.
- [7] M. Lindner, I. Schiller, A. Kolb, and R. Koch. Time-of-Flight sensor calibration for accurate range sensing. *Computer Vision and Image Understanding*, 114:1318–1328, 2010.
- [8] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.
- [9] NASA Ames Intelligent Robotics Group. NASA Astrobbee Robot Software. <https://github.com/nasa/astrobbee>, 2018.
- [10] T. T. Kahlmann, F. Remondino, and H. Ingensand. Calibration for Increased Accuracy of the Range Imaging Camera Swissranger. *ISPRS Commission V Symposium Image Engineering and Vision Metrology*, 2006.
- [11] M. Lindner and A. Kolb. Calibration of the Intensity-Related Distance Error of the PMD TOF-Camera.
- [12] D. Herrera C., J. Kannala, and J. Heikkila. Joint Depth and Color Camera Calibration with Distortion Correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):2058–2064, 10 2012.
- [13] S. Fuchs and G. Hirzinger. Extrinsic and depth calibration of ToF-cameras. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6. IEEE, 6 2008.
- [14] Young Min Kim, D. Chan, C. Theobalt, and S. Thrun. Design and calibration of a multi-view TOF sensor fusion system. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–7. IEEE, 6 2008.
- [15] A. Kuznetsova and B. Rosenhahn. On Calibration of a Low-Cost Time-of-Flight Camera. pages 415–427. Springer, Cham, 2015.
- [16] D. C. Brown and D. C. Brown. Close-range camera calibration. *PHOTOGRAMMETRIC ENGINEERING*, 37(8):855–866, 1971.
- [17] F. Devernay and O. Faugeras. Straight Lines Have to Be Straight Automatic Calibration and Removal of Distortion from Scenes of Structured Environments. *Machine Vision and Applications*, 13:14–24, 2001.
- [18] P. Furgale, J. Maye, J. Rehder, T. Schneider, and L. Oth. Kalibr calibration toolbox. <https://github.com/ethz-asl/kalibr>, 2014.