IS-1500 NATURAL FEATURE TRACKER
USER GUIDE

700 Technology Park Drive, Suite 102
Billerica, MA 01821 USA
Phone +1 781 541 6330 • Fax +1 781 541 6329
www.intersense.com

MNL-0024 (C)

Contacting Thales Visionix

Please contact us if you need assistance.

Thales Visionix, Inc.
700 Technology Park Drive, Suite 102
Billerica, Massachusetts 01821
USA

Telephone: +1 781 541 6330
Internet: http://www.intersense.com
Fax: +1 781 541 6329

Technical Support: +1 781 541 7624 **email: techsupport@intersense.com**
Sales: +1 781 541 7650 **email: sales@intersense.com**

The label below identifies the protection granted by the Government of the United States to Thales Visionix for InterSense products:

U.S. Patents

5645077, 5807284, 6162191, 6361507, 6786877, 6176837, 6409687, 7395181, 6314055 B1, 6757068 B2, 7301648, 6474159 B1, 6681629 B2, 7000469 B2, 7231063 B2, 6922632 B2, 7725253, 7900524, 8224024, 8473241, 8696458, 8762091, 8972182 and Patents Pending.

Foreign Patents

Europe 1280457 1071369 Israel: 152359 Taiwan: 162248
China: 99807510.8 Hong Kong: 1039884 Japan: 4690546.

Trademarks
InterSense® is a registered trademark of Thales Visionix, Inc. InertiaCam™, InertiaCube™, VETracker™, and SimTracker™ are trademarks of Thales Visionix, Inc. All other trademarks are the property of their respective owners.

Regulatory Statements and Approvals

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

This product meets the applicable Industry Canada technical specification.

"CE" mark indicates that this product complies with the European requirements for safety, health, environment and customer protection.

Classified by UL to 60950-1.

Bureau Veritas NRTL Listing.

Thales Visionix, Inc. End User License Agreement

This License Agreement (“Agreement”) is an agreement between you (“Licensee”) and Thales Visionix, Inc. (“Licensor”). It governs use of the software supplied by Thales Visionix, Inc. ("the Software") and related documentation. By downloading, installing or otherwise using the Software, the user agrees to be legally bound by the terms of this Agreement. Please read the entire contents of this agreement. THIS SOFTWARE IS COPYRIGHT © 2017 by Thales Visionix, Inc. ALL RIGHTS RESERVED WORLD WIDE.

IMPORTANT NOTE

Backups of any important information, valuable data, or Software on the host PC should be made before installing this or any other software. Also, regular backups of the Software and any data files should be made, so that in the event of data loss, information may be restored.

1. DEFINITION

(a) “Documentation” means the standard end user manual for the Product provided by Licensor.

(b) “Product” means all Licensor equipment including, but not limited to, the InertiaCube, IS-900, and IS-1500 product families.

(c) “Software” means all Licensor software code including sample source code, compiled code, and embedded firmware.

2. LICENSE

(a) Grant of Rights. Licensor hereby grants to Licensee a non-exclusive, royalty-free, perpetual license to use the Software solely with Licensor Product and solely for the operation of Product in accordance with its documentation. Licensee may redistribute Licensor libraries, isense.dll, libisense.so, libisense.dylib, sfAccess.dll and libsfaccess.so as required for the operation of Licensee products.

(b) Restrictions. The Software (and any copy thereof) is licensed not sold, and Licensee receives no title to or ownership of the Software and no rights other than those specifically granted in Section 2(a) above. With the exception of Licensor sample source code, Licensee may not modify, reproduce, and create derivative works of the Software and shall not attempt to decompile or otherwise reverse engineer the Software. Licensee shall not remove any proprietary or copyright notices of Licensor in the Software or any copies thereof.

3. WARRANTY

This software and accompanying written materials (including instructions for use) are provided "as is" without warranty of any kind. Further, Licensor does not warrant, guarantee, or make any representations regarding the use, or the results of use, of the software or written materials in terms of correctness, accuracy, reliability, currency, or otherwise. The entire risk as to the results and performance of the software is assumed by Licensee. If the software or written materials are defective, Licensee, and not Licensor or its dealers, distributors, agents, or employees, assume the entire cost of all necessary servicing, repair, or correction.

The above is the only warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose, which is made by licensor on this product. No oral or written information or advice given by licensor, its dealers, distributors, agents or employees shall create a warranty or in any way increase the scope of this warranty and the user may not rely on any such information or advice.

4. LIMITATION OF LIABILITY

Neither licensor nor anyone else who has been involved in the creation, production or delivery of this product shall be liable for any direct, indirect, consequential or incidental damages (including damages for loss of business profits, assets, business interruption, loss of business information, and the like) arising out of the use or inability to use such product even if licensor has been advised of the possibility of such damages.

5. TERMINATION

(a) Termination. This Agreement shall terminate automatically and immediately upon any breach of this agreement by Licensee, including but not limited to any action in violation of the license rights and restrictions set forth in Section 2.

(b) Effects of Termination. Upon termination of this Agreement, the licenses granted in Section 2(a) will terminate and Licensee will cease all use of the Product and delete all copies of any Software (including Upgrades and Updates stored on Licensee computers) in its possession or control. This Section 5(b) shall survive termination of this Agreement.

6. MISCELLANEOUS

(a) Notices. Notices to Licensor pursuant to this Agreement will be sent to 22605 Gateway Center Drive, Clarksburg, Maryland 20972, Attention: Contracts Department. Such notices will be deemed received at such addresses upon the earlier of (i) actual receipt or (ii) delivery in person, or by certified mail return receipt requested.

(b) U.S. Government Use. The Software is a “commercial item” as that term is defined at 48 C.F.R.

2.101, consisting of “commercial computer software” and “commercial computer software documentation” as such terms are used in 48 C.F.R. 12.212.

(c) Export. Licensee understands and recognizes that the Product is export-controlled and agrees to comply with all applicable export regulations. The Product may not be exported to any country outside the country where the Product was acquired without Licensor’s prior written consent.

(d) Assignment & Successors. Licensee may not assign this Agreement or any of its rights or obligations hereunder.

(e) Choice of Law & Jurisdiction. This Agreement will be governed by the laws of the United States of America.

(f) Severability. To the extent permitted by applicable law, the parties hereby waive any provision of law that would render any clause of this Agreement invalid or otherwise unenforceable in any respect. In the event that a provision of this Agreement is held to be invalid or otherwise unenforceable, such provision will be interpreted to fulfill its intended purpose to the maximum extent permitted by applicable law, and the remaining provisions of this Agreement will continue in full force and effect.

(g) Entire Agreement. This Agreement sets forth the entire agreement of the parties and may not be modified except by a written agreement signed by Licensor.

TABLE OF CONTENTS

TABLE OF FIGURES

1. INTRODUCTION

1.1 Overview of the IS-1500

The **IS-1500 Natural Feature Tracker** (**NFT**) is a motion tracking system that uses both visual and inertial input to calculate and output position and orientation data. The system enables GPS-denied tracking and navigation in a free-roaming environment. The IS-1500 kit includes an InertiaCam tracker, a USB-C 2.0 cable, a fiducial poster, and a sensor fusion software suite.

Figure 1 – InertiaCam

The **InertiaCam** is an optical-inertial **6-DOF** (six degrees of freedom) motion tracker. It pairs a monocular camera with a specially developed InterSense inertial measurement unit (**IMU**), known as the **NavChip**. There is a female USB-C connector at the base of the InertiaCam used to connect to the host computer.

InertiaCams have limited internal processing capability. The majority of image and inertial data processing is performed by the sensor fusion software installed on the host computer. The flow chart in **Figure 2** provides an overview of the IS-1500 system architecture. The arrows indicate the flow of the data processing chain, beginning with sensory input from the InertiaCam and ending with the output of tracking data.

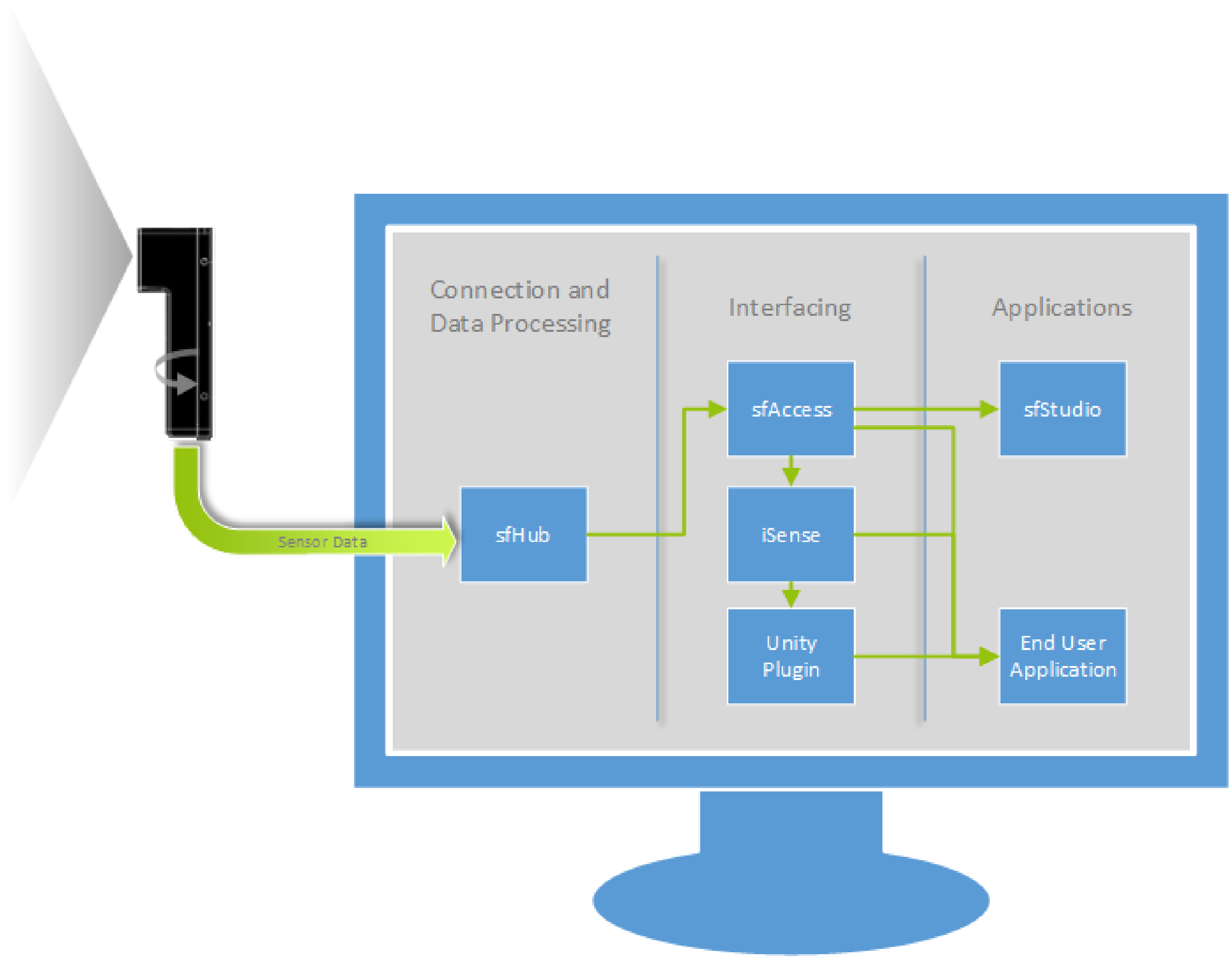


Figure 2 – IS-1500 System Architecture

As the InertiaCam receives optical data from the camera and inertial data from the NavChip, it transmits the information directly to the sfHub program. **sfHub** is responsible for the bulk of the processing of the raw sensor data. It outputs packets of processed tracking data, which are then transmitted to sfAccess.

sfAccess provides a low-level interface to the InertiaCam and sfHub and is only recommended for users requiring advanced features of the IS-1500. The **InterSense library** can be used instead if the host application simply needs tracking output or if compatibility with other InterSense tracker models is required. The suite also comes with an **IS-1500 Unity Plugin**, allowing users to directly integrate the IS-1500 into their Unity applications.

sfStudio is used as the primary program for device setup, display of tracking information, and general diagnostics. sfStudio can also be used to show the tracking status, provide graphical data plots such as tracker position and orientation over time, and display the live camera feed from the InertiaCam.

With regards to terminology throughout the rest of this user guide, the term *IS-1500* will refer to the *system* as a whole. The *system* encompasses the InertiaCam, host computer, and full software suite. The InertiaCam may be referred to as simply the *tracker*. Similarly, the user host computer may be referred to as the *processor*. The software components will typically be identified specifically, although sfAccess, InterSense library, and Unity plugin may be generalized as the *tracking interface* as needed. As will be discussed later in the Understand Tracking section, the IS-1500 is capable of several tracking modes, though these may be collectively known as the *tracking algorithms*.

2 UNDERSTANDING TRACKING

This section is intended for those users who wish to better understand the tracking data produced by the IS-1500 and how it acquires that information. It starts with the basics by briefly reviewing common terms and the Cartesian coordinate reference frame. From there it moves on to discuss how the IS-1500 uses the InertiaCam's NavChip IMU and camera to track its position and orientation.

In explaining how the tracking works, there are many general tips for usage that come up in this section. A brief bullet list of these tips is provided in the Summary subsection.

This section uses many graphs and screenshots from the sfStudio program to illustrate the topics covered. For users who wish to follow along and view live motion tracking data, it may be beneficial to first go through the **IS-1500 Quick Start Guide** or the Installation section to set up the system before proceeding.

2.1 *Tracking Basics*

The **tracking environment** (or **tracking area**) broadly refers to the area the IS-1500 is being used in. The tracking data produced by the IS-1500 uses a Cartesian (rectangular) coordinate reference frame to describe position and orientation within the tracking environment.

The tracking data produced by the IS-1500 reflects the InertiaCam's position and orientation, as defined by the tracker reference frame, relative to the world reference frame. The X, Y, and Z position of the tracker is reported in meters. Orientation is described using Euler angles to indicate roll, pitch, and yaw.

As illustrated in **Figure 3**, **roll**, **pitch**, and **yaw** are the rotations around the X, Y, and Z axes of the world reference frame. As such, roll changes the heading of the YZ plane of the tracker reference frame, pitch changes the XZ plane, and yaw changes the XY plane.

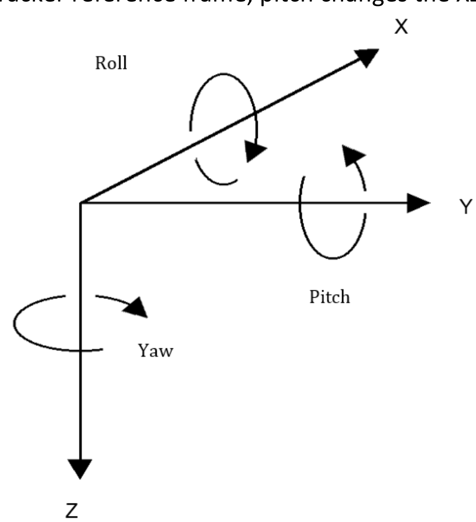


Figure 3 – Rotational Axes

An **origin point** and X, Y, and Z axes for the **world reference frame** are defined by the system when it is initialized. The **origin point** defines the location in the tracking environment at which (X, Y, Z) is (0, 0, 0). When the IS-1500 is first initialized, the only sensory data it can rely on to create a world reference frame is the direction of gravity. Therefore, it initializes using the accelerometers to determine the direction of gravity. The Z axis is always in the direction of gravity, defining pitch and roll, while the direction of the X axis, defining yaw, is effectively arbitrary. Using a right handed coordinate reference frame, the Y direction is 90° clockwise along the X axis. By default, the IS-1500 sets the world origin point as the InertiaCam's position when tracking is first initialized. As will be discussed in the PRA Fiducial Tracking subsection, this initial world reference frame can be changed later by using a fiducial constellation to provide the system with a global point of reference.

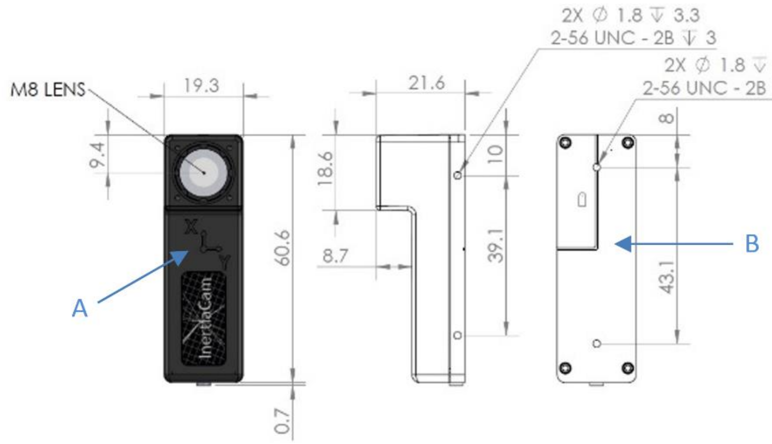


Figure 4 – InertiaCam Dimensions

The **tracker reference frame** (or **body frame**) is a predetermined point and orientation on the InertiaCam and stays fixed to it. It is indicated by markings on the InertiaCam housing, shown in **Figure 4**. The XY marking at point A on the front of the housing indicates the orientation of the positive X and Y axes. The positional origin is located and marked at the back of the housing as the corner of the rectangle, indicated by point B. (Like the markings on the front, the long and short sides of the rectangle on the back also indicate the direction of the positive X and Y axes, respectively).

To help illustrate this, the sfStudio 3D Data Display in **Figure 5** shows a tracker that has just been initialized with the InertiaCam laying on a flat surface so the Z axis of the tracker is in the direction of gravity, the same as the world frame. The world frame in this image has been marked in yellow. Gray gridlines lie across as the XY plane. The axes of the body frame are colored green, blue, and red for X, Y, and Z, respectively. Note that the X and Y axes of the tracker may not line up exactly with those of the world frame.

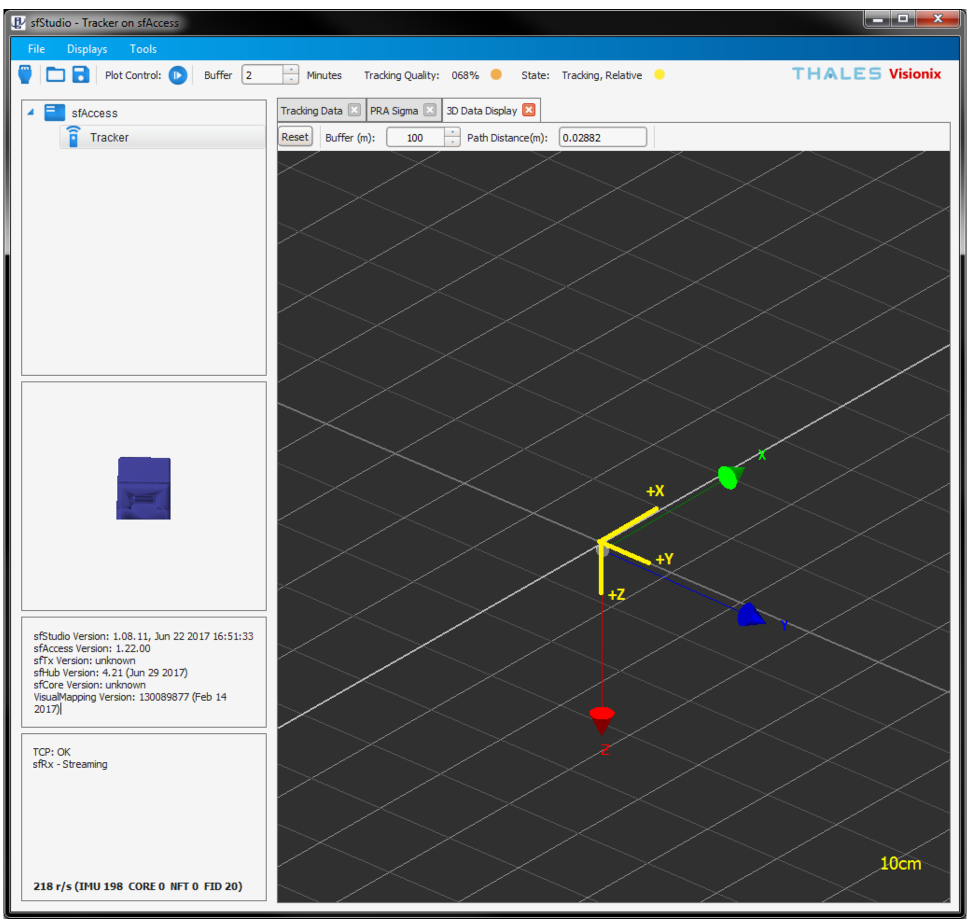


Figure 5 – Initialized Reference Frames

Figure 6 shows another example. Here, the tracker has been moved from its initial position. Instead of lying flat with the camera pointed up, it has been rotated into a more vertical position and moved along an arc towards a position somewhere above the starting point. The thin yellow line shows the path the tracker took. The world frame has remained stationary, but the body frame has moved with the InertiaCam.

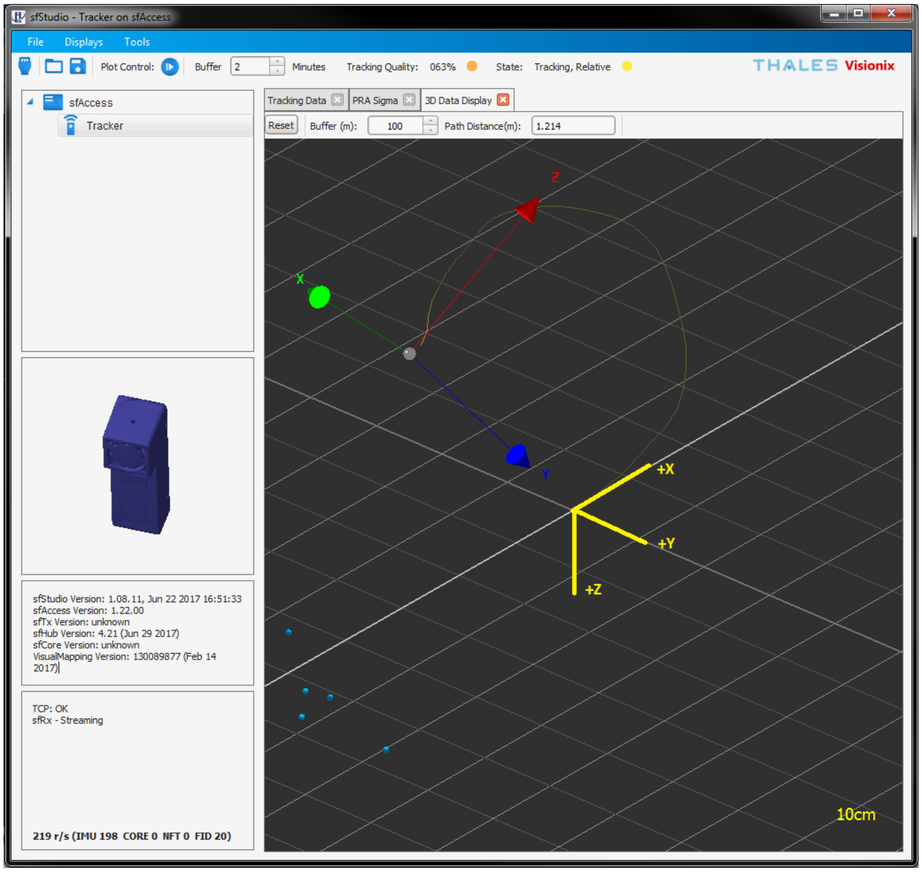


Figure 6 – Tracked Reference Frame

2.2 Achieving Accurate 6-DOF Tracking

The IS-1500 uses optical-inertial tracking, meaning that it uses visual data from the camera along with inertial data from the NavChip IMU to acquire more accurate positional information. As the IS-1500 Natural Feature Tracker name indicates, it uses natural features it identifies in the environment to track off of. **In this context, natural features are those which the system determines to be still or motionless.** However, the IS-1500 can also use fiducial tracking, which was originally developed for earlier optical-inertial trackers like the IS-1200 and HObIT.

2.2.1 3-DOF Tracking from the NavChip

As previously mentioned, the InertiaCam tracker included with the IS-1500 kit is a **6-DOF** (rotation and position) tracker with a monocular camera and a built-in NavChip IMU. The NavChip IMU is the source of high precision **3-DOF** (three degrees of freedom-yaw, pitch, and roll) tracking data. The accuracy of the data produced by the NavChip contributes greatly to the precision performance of the IS-1500 as a whole.

The NavChip measures changes in acceleration and rotation by using an array of individually characterized, semi-redundant accelerometers and gyroscopes. The angular rate and linear acceleration data from the sensors is integrated to produce the standard ΔΘ (angular displacement) and ΔV (change in velocity) IMU outputs. These outputs are integrated at 200Hz to track orientation and position.

This data allows the NavChip to also function as an **attitude and heading reference system** (**AHRS**) to measure yaw, pitch, and roll. The accelerometers provide measurements of pitch and roll by calculating the proportion of gravity along those vectors. However, yaw lies in the XY plane and gravity is along the Z axis, so accelerometers cannot be used to measure yaw. Because of this, gyroscopes are also used, which are capable of measuring rotation about each axis. While gyroscopes are less accurate than accelerometers on the whole, they serve the dual purpose of measuring yaw and providing supplemental pitch and roll data to increase overall accuracy. This is why IMU specifications for pitch and roll accuracy and drift are typically listed separately from yaw, which is prone to drift if not corrected by external data.

While IMUs provide accurate orientation data, even the high precision NavChip cannot be used alone to acquire accurate 6-DOF positional data. Every IMU is prone to a certain degree of drift, especially in yaw, and while the NavChip has been specifically designed to minimize drift, it does not eliminate it. The drift errors are not as problematic when performing a single integration to find orientation. However, acquiring position information requires a double integration of the sensor data, allowing drift to accumulate quadratically with time. An example of this is illustrated towards the end of the VINS Natural Feature Tracking subsection.

In order to obtain accurate 6-DOF tracking, an external source of positional reference is also required.

2.2.2 VINS Natural Feature Tracking

To track off of natural features, the IS-1500 uses a computationally lightweight **Vision-aided Inertial Navigation System (VINS)** algorithm. When an image from the InertiaCam camera is received by the sfHub software, it is processed to isolate particular pixel patches that the system considers unique and identifiable. Frequently the patches will be found in regions of sharp contrast, such as corners or edges. These pixel patches are the **natural features**. **Figure 7** shows the Optical Data display with Image Transfer and VINS Image enabled. It illustrates the features found by VINS while viewing a collection of objects. Each blue box outlines a pixel patch feature with a unique ID in green.

Figure 7 – VINS Features

As the camera continues to provide subsequent frames, they are analyzed to relocate the natural features. If they can't be located or become compromised, they are replaced with new features. Reasons for this might be that the feature has moved out of the frame and is no longer visible, or that it is simply unrecognizable. (Note that many of the features in **Figure 7** are partially identified by a glare, such as 13686 on the tree branch of the ceramic plate. If the light causing the reflection were blocked, that feature would likely become unrecognizable.) The system also has a preference for 'natural' (immobile) features. If a feature previously thought to be motionless is eventually determined to be mobile, it will soon be discarded in favor of a stationary feature.

With VINS, data on each feature is not saved to permanent memory. Data is only maintained for currently tracked features. This means that if the camera is turned away for several seconds and brought back to the same field of view, the previous features will not be identified again. Even if the same pixel patch is identified as a feature, it will have a new feature ID and any data describing it will be generated from scratch.

The VINS algorithm is **robocentric**, meaning that internally the features are described with a distance and bearing vector relative to the tracker reference frame. Humans rely on binocular vision for depth perception and estimation of distances. However, the IS-1500 uses the monocular InertiaCam. To make up for this, the distance of a feature from the InertiaCam is calculated by using **parallax**. Upon first finding features, the system can only roughly estimate their distance from the camera. As the InertiaCam is moved from side to side, the movement of the features from frame to frame is compared to the precise velocity data from the NavChip. With this information, the IS-1500 can accurately determine the distance of each feature. **In practical usage, this means that the more the InertiaCam moves about a feature, the more the system will learn about it, improving the precision of the tracking data.**

Incidentally, this is also how the system determines that a feature is mobile. If the feature is moving, its distance and bearing will not correlate with the IMU data, and it will eventually be rejected. It is simple to demonstrate this by placing a hand in the InertiaCam's field of view. If the hand is kept still, it is likely to be used as a source of natural features, as seen on the left of **Figure 8**. However, when the hand is waved while the InertiaCam is kept still, the features are soon discarded and replaced by alternatives external to the hand. When the system is uncertain about the feature depth, this will be indicated by a red circle or oval, as seen in features 2172 and 2173. The longer the oval, the greater the degree of uncertainty.

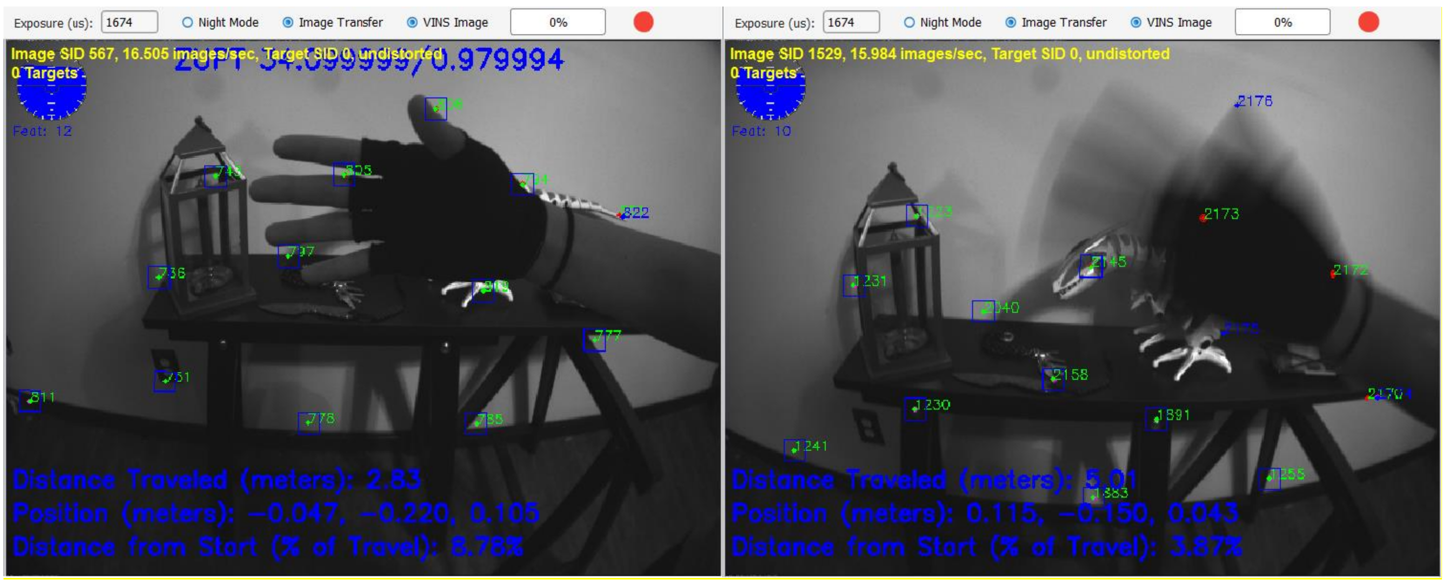


Figure 8 – VINS Hand Waving Test

If both the hand and the InertiaCam are in motion, it is less likely the features will be discarded. Instead the system will continue trying to track off the mobile features. This will feed false pose data to the system and may cause problems with tracking. **For accurate tracking, if there are moving objects in the tracking area, it is generally best to keep them out of the InertiaCam's field of view. If this is not possible, keep the InertiaCam still so the system can discard mobile features and replace them with natural features.**

The depth estimates calculated for each feature carry a degree of possible error. The amount of possible error scales with the distance of a feature from the InertiaCam. This means that while features can be found a long distance away on the horizon, tracking off of them may not yield reliable data. Beyond about two feet, the greater the distance between the detected natural features and the InertiaCam, the less reliable the tracking data will be. However, features that are further away also typically leave the field of view less frequently, providing more stable points of reference. **A general guideline is to keep the InertiaCam field of view pitched about 20° below the perpendicular of the horizon line while tracking.** This will allow the majority of features to be found on the ground and in nearby surroundings with a few still on the horizon. At walking speed, the features on the ground will be in enough frames to provide accurate depth information before leaving the field of view, while features on the horizon provide stability.

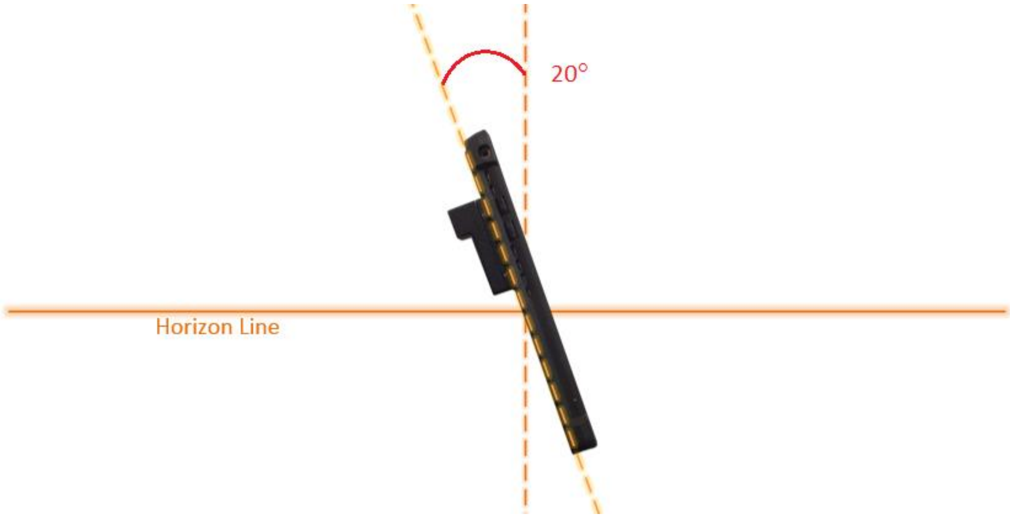


Figure 9 – Optimal 20° Pitch

The VINS algorithm maintains an **extended Kalman filter** (**EKF**) to enable bias correction and a certain level of movement prediction. The EKF is state based, meaning that it is updated with each new tracking calculation. This also means that when tracking is first initialized, the filter does not have much data and can be less reliable. **To ensure the EKF receives more reliable data upon initialization, keep the tracker still while tracking is initializing (before data begins streaming from sfHub).**

To help optimize tracking performance, it is good practice to begin tracking by *exercising* the InertiaCam's NavChip. To do this, begin by placing the InertiaCam on a level surface with the camera facing up. Proceed to rotate the InertiaCam about 90° in each axis, leaving it still for five seconds or more between each rotation. Be sure not to obstruct the camera during this process. Exercising the IMU allows the EKF to begin accumulating IMU bias data and apply accurate corrections to the raw data.

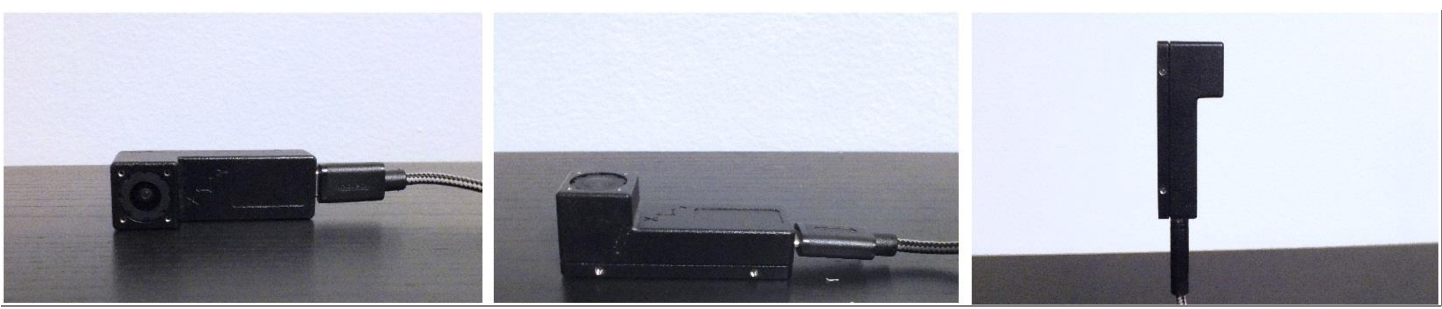


Figure 10 – Exercising an IMU

If optical information in the VINS filter is determined to be unreliable due to a particularly high error estimates in the EKF, data from the IMU will carry more weight in the tracking calculations. For instance, when the InertiaCam is rotated very quickly, or if its vision is temporarily obstructed, VINS tracking will discard all of the features it had previously identified and will need to characterize a completely new set of features. Because the system doesn't have known distances for these features yet, the quality of the tracking data may slightly and temporarily decrease. However, the EKF data will allow the system to rely more heavily on IMU data while the new VINS features are established. This allows the system to stay relatively stable and continue tracking accurately under these conditions.

If the InertiaCam's vision is obstructed for long periods of time, the system will begin to rely almost entirely on the IMU data from the NavChip. As discussed earlier, when the IMU is used exclusively to provide position, there is a high rate of drift. As an example, after connecting to the InertiaCam and beginning tracking, the camera of the InertiaCam is covered. Soon after this, the tracker's position can be seen drifting away at an increasingly rapid rate. This behavior is known as a **divergence**. The software is designed to recognize this divergence and counter it by resetting the filter and returning the tracker position back to the origin. **Figure 11** shows the 3D Data Display of a divergence that has resulted in a filter reset.

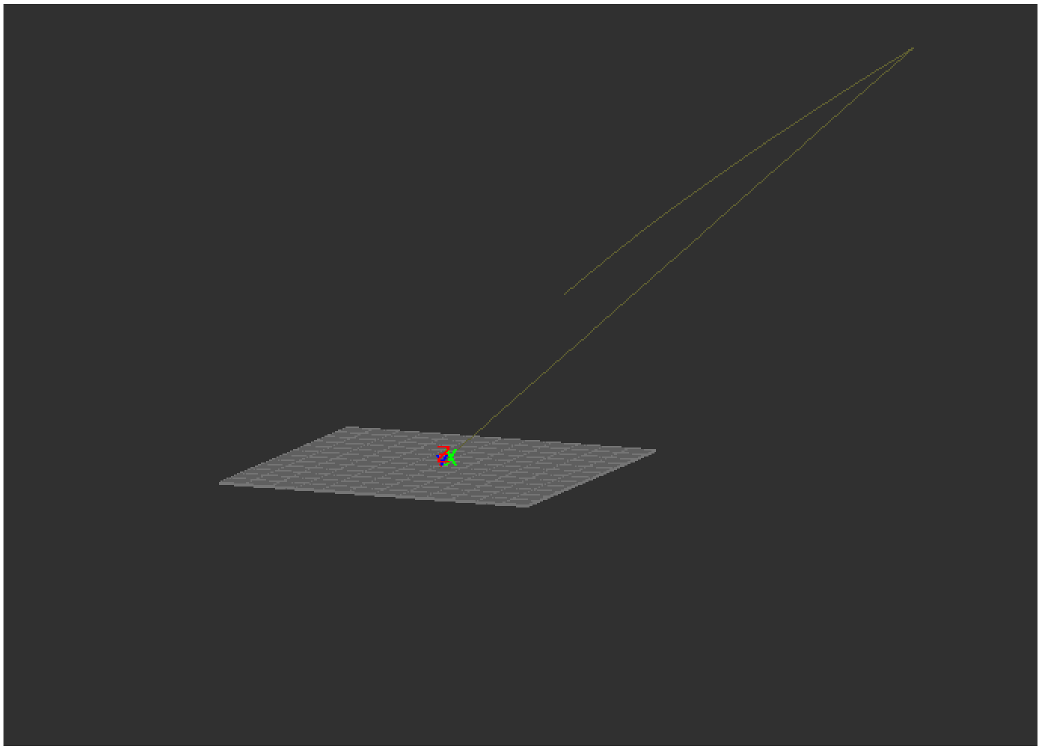


Figure 11 – Divergence

The tracker will continue to diverge and reset until vision is restored and natural features can be found again, as shown in **Figure 12**. When the divergence is detected and the filter reset, the system will place the position back at the last location accurate data was recorded. Orientation will be relatively unaffected. Because the filter is reset, when recovering from a divergence, it is best to treat the tracker as though it was just initialized. **This means while the system recovers from a divergent state after the camera has been obstructed, it is best to keep the tracker still at first, then exercise it or (if using a fiducial constellation to set a world frame) bring fiducials in view.**

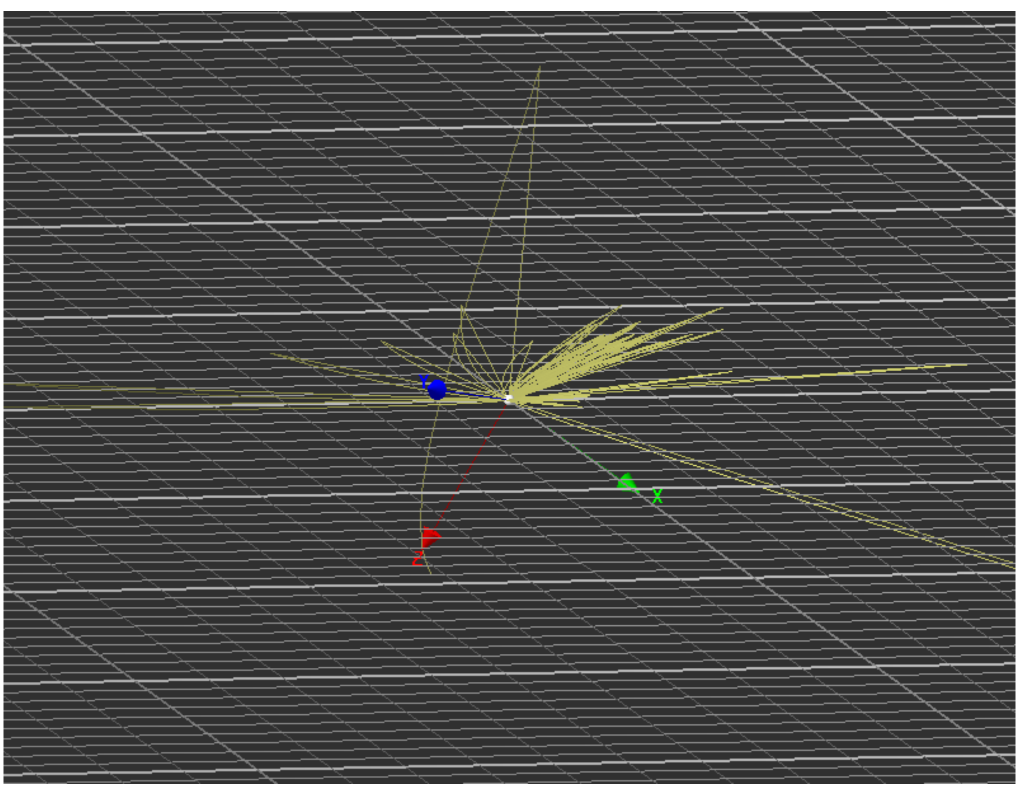


Figure 12 – Continued Divergences

2.2.3 PRA Fiducial Tracking

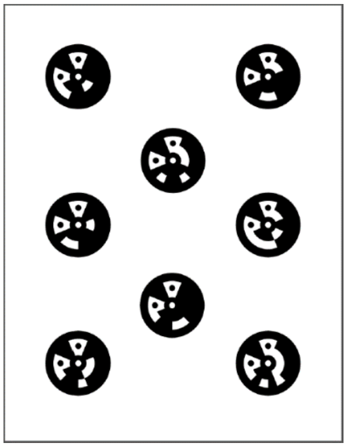


Figure 13 – Default Fiducial Constellation

InterSense optical inertial trackers such as the IS-1500 are capable of tracking off of fiducials like the ones found on the poster included with the hardware kit. InterSense **fiducials** are circular black-and-white patterns that act as known physical landmarks for the system. A fiducial **constellation** refers to the layout of all fiducials in a tracked area. When placed on a flat, level surface, the poster serves as the default fiducial constellation for the IS-1500 system. **Constellations and their fiducials provide predetermined points of reference for the tracker.** Fiducial constellations enable another level of drift correction and set a fixed environment reference frame for the system.

The circular patterns of the fiducials are key to their functionality. The three circles within the pattern (the center eye and two outer eyes) are what the system's image recognition component uses to recognize a pattern as a fiducial. These circles also serve as the tracked points. **Damage to any of these three points, especially the center eye, can render a fiducial unidentifiable to the system. Because the InertiaCam is a visible light tracker reading black-and-white fiducials, uneven lighting across the constellation can also render the fiducials unidentifiable during pattern recognition.**

The rest of the pattern acts like a barcode. The black and white sections are decoded by the software and translated to a numeric value. The numerical value is known as the **fiducial ID**. **In a single constellation, no two fiducials can share a fiducial ID. They must all have a unique ID and therefore a unique pattern. Figure 14** shows the constellation as seen in the Optical Data display. The fiducial overlays shows a square around each recognized fiducial with crosshairs for each eye. The numbers indicate the fiducial IDs.

Figure 14 – Fiducial Overlay

The algorithm used by the IS-1500 for fiducial tracking is called the **Pose Recovery Algorithm** (**PRA**). Both VINS and PRA are run concurrently while the system is tracking, allowing the IS-1500 to use both natural feature tracking and fiducial tracking simultaneously.

The PRA uses the constellation map file (**environmentPSEs.cfg**, located in the ExampleConstellation folder) as a reference. The **constellation map file** acts as a look-up table and includes a list of the IDs of the fiducials in the constellation and their pose relative to a world reference frame. While tracking, the PRA thread cross-references the observed fiducials with their expected layout. The tracker's position and orientation relative to the constellation is then calculated and updated. As far as the system is concerned, the constellation is defined by the map file. If a fiducial is listed in the map file but is never recognized by the system, it will simply never be used for tracking. Additionally, if a fiducial is identified by PRA but the ID is not listed in the map file, it will be disregarded. **However, if a fiducial is detected, the ID is in the map file, and it is in a different position and orientation than what is described in the map file, tracking can become compromised. Also note that the poster and its constellation must be oriented correctly relative to gravity. If the poster is placed on the ceiling or floor (sideways or upside down), the system will not track properly unless the constellation file is modified accordingly.** The constellation file can be rotated in the included sfStudio application's constellation display if such a configuration is desired.

PRA data is only used when the system is able to obtain **optical lock** on the constellation. There are a number of requirements for optical lock to be achieved. **Most importantly, the system must be able to recognize at least four fiducials simultaneously to have enough data points to acquire optical lock.** There are also a number of quality metrics that must be met, but it is possible to obtain optical lock on an incorrect constellation.

In practical usage, when placed in the tracked area, a constellation serves two main purposes. The first is that the constellation defines a fixed world reference frame. Whenever the constellation is first brought into view and optical lock is acquired, the world reference frame changes to be relative to the constellation rather than relative to wherever tracking was initialized. (This is illustrated further in the NFT Status change from 1 to 2 in the <u>Installation</u> section.) If the entire constellation is moved, the world reference frame will move with it. This change happens when the system has acquired.

The second use for fiducial tracking is that it provides another level of drift correction. When using only VINS tracking, there is a certain degree of expected drift- about 1% of the distance travelled for distances under 200 meters. For example, if the user tracks for 50 meters in a circle, the system may report a position that is up to half a meter from the starting position upon return. However, whenever a fiducial or constellation is brought back into view, this drift is corrected since the tracker position is now known relative to a fixed point in the environment. Continuing the example, if there is a fiducial constellation at the starting position, after tracking a 200 meter circle, VINS may have allowed the starting position to drift approximately two meters. When the constellation is recognized, however, the position will snap back to the actual starting position. Also, the Extended Kalman Filter will reset when the PRA achieves optical lock on the constellation, which will correct any aberrant behavior. **Exercising the tracker is not necessary when using a fiducial constellation.**

Using a constellation is recommended for applications that require tracking over large expanses or in cases where the world reference frame needs to be precisely defined. If the provided constellation poster does not meet the needs of an application, a custom constellation and corresponding map file can be created. Instruction for how to create a custom constellation is provided in the **Fiducial Constellation Guide**. When using the IS-1500, constellation density requirements are not as strict as they are for other InterSense optical-inertial trackers. For example, rather than covering a continuous area such as a poster or a room, fiducials can instead be placed in clusters at key landmarks throughout a commonly tracked area.

2.3 Summary

2.1 Tracking Basics

- The tracking data produced by the IS-1500 reflects the InertiaCam's position and orientation, as defined by the tracker reference frame, relative to the world reference frame. The X, Y, and Z position of the tracker is reported in meters. Orientation is described using Euler angles to indicate roll, pitch, and yaw.

2.2 Achieving Accurate 6-DOF Tracking

- The IS-1500 uses optical-inertial tracking, meaning that it uses visual data from the camera as an external reference to acquire accurate positional information.
- The IS-1500 uses two methods of optical-inertial tracking to achieve accurate 6-DOF data - VINS to track off of natural features and PRA to track off of InterSense fiducial constellations.

2.2.1 3-DOF Tracking from the NavChip

- The NavChip IMU is the source of high precision 3-DOF (three degrees of freedom - yaw, pitch, and roll) data.
- While IMUs provide accurate orientation data, even the high precision NavChip cannot be used alone to acquire accurate 6-DOF positional data.

2.2.2 VINS Natural Feature Tracking

- In the context of optical-inertial tracking, natural features are those which the system determines to be still or motionless.
- With VINS, data on each feature is only maintained for currently tracked features.
- The more the InertiaCam moves about a natural feature, the more the system will learn about it, improving the precision of the tracking data.
- A general guideline is to keep the InertiaCam field of view pitched about 20° below the perpendicular of the horizon line while tracking.
- For accurate tracking, if there are moving objects in the tracking area, it is generally best to keep them out of the InertiaCam's field of view. If this is not possible, keep the InertiaCam still so the system can discard mobile features and replace them with immobile, natural features.
- To ensure the EKF receives more reliable data upon initialization, keep the tracker still while tracking is initializing.
- To help optimize tracking performance, it is good practice to begin tracking by exercising the InertiaCam's NavChip.
  - To do this, begin by placing the InertiaCam on a level surface with the camera facing up.
  - Proceed to rotate the InertiaCam about 90° in each axis, leaving it still for approximately five seconds or more between each rotation.
  - Be sure not to obstruct the camera during this process.
- If the optical information within the VINS filter is determined to be unreliable, data from the IMU will carry more weight in the tracking calculations.
- If the InertiaCam's vision is obstructed for long periods of time, the system will begin to rely almost entirely on the IMU data from the NavChip and pose data will diverge.
- It is best to keep the tracker still after the camera has been obstructed while the system recovers from a divergent state.

2.2.3 PRA Fiducial Tracking

- Constellations and their fiducials serve as known, predetermined points of reference for the tracker.
- Damage to any of the three circles of a fiducial can render it unidentifiable to the system. Because the InertiaCam is a visible light tracker reading black-and-white fiducials, uneven lighting across the constellation can also render the fiducials unidentifiable during pattern recognition.
- In a single constellation, no two fiducials can share a fiducial ID. They must all have a unique ID and therefore a unique pattern.
- If the constellation in use is not accurately reflected in a map file, tracking can become compromised.
- For PRA to be used, the system must be able to recognize at least four fiducials simultaneously to have enough data points to acquire optical lock.
- In practical usage, when placed in the tracked area, a constellation serves two main purposes. The first is that the constellation defines a fixed world reference frame. The second is that it provides another level of drift correction.
- Using a constellation is recommended for applications that require tracking over large expanses or in cases where the world reference frame needs to be precisely defined.
- Exercising the tracker is not necessary when using a fiducial constellation.

3 INSTALLATION

The IS-1500 system requires both hardware and software installation prior to use. The IS-1500 Quick Start Guide covers a fairly basic installation of the system. Reading through and following the IS-1500 Quick Start Guide is recommended for a user's first attempt to track with the IS-1500 as it allows the user to gain familiarity with the product in a controlled environment. This guide will take a more detailed approach to the IS-1500 Installation, covering both Windows and Linux installation of the software suite and assuming use of a custom fiducial constellation. Much of the Windows information may appear familiar from the Quick Start Guide.

Installation instructions will be provided in a stepwise fashion. If problems during setup are encountered at any point, refer to the Troubleshooting section or contact Thales Visionix Technical Support using the contact information provided on Page 2.

3.1 Installing the Software Suite

3.1.1 Windows

1. Download the Windows IS-1500 Software Installer.exe from the http://intersense.com/is1500/ website.

2. Run the downloaded installer executable.
   a. Select the components for installation when prompted by the installer. For a first time installation, it is recommended that all checkboxes be checked.
   b. The next page shows an option to install with **Embedded Settings**, which optimizes the tracking settings for devices with lower processing power. It is recommended that this option be checked when using the IS-1500 with a tablet or mobile device.
   c. Finally, select the preferred installation directory.

3. The installer will create an InterSense IS-1500 folder in the Start Menu under All Programs.
   - **Documentation:** Links to the folder containing various IS-1500 user guides and a printable PDF of the default 8.5x11 fiducial constellation
   - **InterSense Unity Plugin:** Links to the InterSense Unity package, which enables the user to interface Unity applications with tracking data from the IS-1500
   - **isUnitySample:** The precompiled Unity demo program
   - **SDK:** Links to the folder containing DLL files and sample code for integrating tracking data into the user software application
   - **sfHub:** Program used by sfStudio and the isUnitySample to interface with the InertiaCam and generate tracking data
   - **sfStudio:** The main GUI application used to map, test and re-configure the tracker as well as view and record both raw and fused data

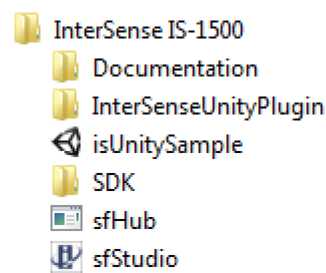


Figure 15 – Start Menu Folders

4. The installer will also create an IS-1500 Software Package folder at the designated installation directory. All of the tools needed to configure and test the IS-1500 as well as map new fiducial constellations are included within this package. The tracker's software directory structure is shown in **Figure 16**.
   - **Configuration Files:** Contains copies of the default constellation and configuration files
   - **Documentation:** Contains various IS-1500 user guides and a printable PDF of the default 8.5x11 fiducial constellation
   - **SDK:** Contains the required DLL files and sample code for integrating tracking data into the user software application
   - **UnityPlugin:** Contains the InterSense Unity package, which enables the user to interface Unity applications with tracking data from the IS-1500
   - **IS1500Driver:** Contains the driver required for the host PC to interface with the InertiaCam
   - **sfHub:** Contains the sfHub software, which receives sensory data from the InertiaCam and processes it to produce raw tracking data
   - **sfStudio:** Contains the main GUI application used to map, test and re-configure the tracker as well as view and record both raw and fused data

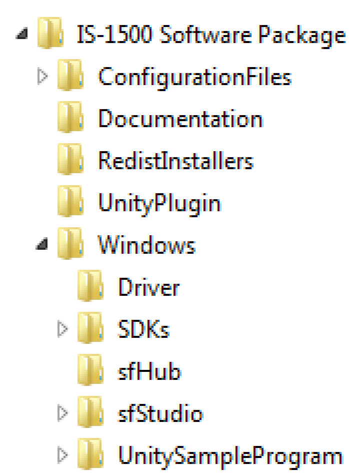


Figure 16 – Software Directory

3.1.2 Linux

1. Download the full software package from the InterSense website at https://intersense.com/is1500/. This is a different download than the installation executable and contains the full software suite for both Windows and Linux. When using the software on a Linux system, it is important to only use software and files from the Linux folder. The file architecture can be found in **Figure 17**. Descriptions for notable folders are as follows.

    - **Configuration Files:** Contains backups of the default constellation and configuration files
    - **Documentation:** Contains various IS-1500 user guides and a printable PDF of the default 8.5x11 fiducial constellation
    - **SDKs:** Contains the required DLL files and sample code for integrating tracking data into the user software application
    - **sfHub:** Contains the sfHub software and required files, which receives sensory data from the InertiaCam and processes it to produce raw tracking data
    - **sfStudio:** Contains the main GUI application used to map, test and re-configure the tracker as well as view and record both raw and fused data
    - **UnitySampleProgram:** Contains the precompiled Unity Sample Program
    - **UnityPlugin:** Contains the InterSense Unity package, which enables the user to interface Unity applications with tracking data from the IS-1500

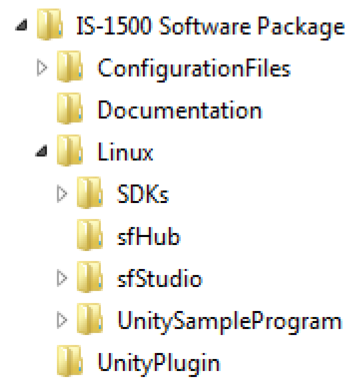


Figure 17 – Linux Software Package

2. Linux systems will require a number of prerequisite library installations before being able to run the sensor fusion software. The following libraries should be installed or updated to the latest version using apt-get commands.
    - libopencv-core2.4v5
    - libopencv-features2d2.4v5
    - libyaml-cpp0.5v5
    - libudev1
    - v4l-utils

3. Once the libraries are installed, a few additional settings need to be established for the software to run properly.
    a. The following two lines need to be added to **/etc/security/limits.conf** to enable setting of thread priorities:

    ```
    hard rtprio 99
    soft rtprio 99
    ```

    b. For the software to access the serial port immediately after the InertiaCam is hot-plugged, **modemmanager** must be removed using the following command:

    ```
    sudo apt-get remove modemmanager
    ```

    c. To allow access to the serial ports, the user must be added to the **dialout** group using the following command:

    ```
    sudo usermod -a -G dialout <username>
    ```

3.2 Connecting using sfStudio

Before using a custom fiducial constellation, it is best to first ensure the tracker has been setup correctly by attempting to track off the default constellation using **sfStudio**. More information on the sfStudio program can be found in the Software section and in the **sfStudio User Guide (MNL-0020)**.

1. If it is not already connected, plug in the InertiaCam to the host computer using the USB C cable included with the IS-1500 kit. When the InertiaCam is plugged in, a blue light will appear for about one second before turning off again to indicate that the InertiaCam has been successfully powered.

2. A fiducial poster is included with the IS-1500 hardware kit. If the poster has been lost or damaged, it can be reprinted from a PDF of this constellation found in the **Documents** folder of the software package, **Fiducial Poster – 8.5 x 11.pdf**. Lay the poster on a flat surface with the fiducials facing the ceiling.

3. Once the default fiducial constellation has been acquired and the InertiaCam connected, run **sfStudio** from the **sfStudio** subfolder. (If the Windows installer has been used as in the Quick Start Guide, sfStudio can instead be run from the Windows Start Menu or Programs list.)

4. When sfStudio first loads, it will not yet be connected to the InertiaCam. This is indicated by the **No Communication Tracking State** and associated grey icon, seen in **Figure 18**. Note that the Tracking Quality is also at 0% with a grey icon. Press the **Connect** button to far left, also highlighted in **Figure 18**, to launch **sfHub** and initate a connection with the tracker.

Figure 18 – Main sfStudio Toolbar

5. A popup will come up that indicates the system is attempting to connect to the tracker. After about fifteen to thirty seconds, the tracker should connect. A number of changes to the sfStudio GUI will take place after the tracker has successfully connected, as seen in **Figure 19**. Primarily, the NFT Status will have changed from **No Communication** with a greyed out icon to **Tracking, Relative** with a yellow icon. The Tracking Quality data will also populate accordingly, showing a 68% quality in the figure below. Additionally, to the right, a blue 3D model of the InertiaCam will be present that responds to and show changes in orientation. Towards the bottom left, blue text will display the data transfer rate in records per second (r/s). **Figure 19** shows the tracker connected and transferring 221 r/s. The Tracking Data tab will also be displayed and begin drawing graphs for orientation, position, and status information.

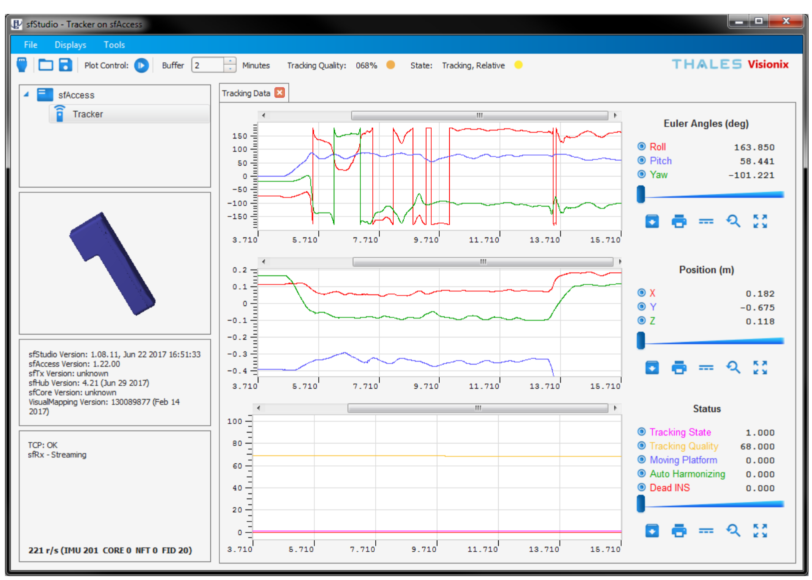


Figure 19 – Tracking Data Display

3.3 Tracking using VINS in Tracking State 1

When the Tracking State is in Mode 1 (Tracking, Relative), the system is producing accurate tracking data using VINS to track off natural features. It is possible to continue tracking this way accurately for about 250 meters (in ideal conditions) without using a fiducial constellation to correct for drift.

1. To view the features InertiaCam is tracking off of, use the **Optical Data Display.**
   a. Click the **Displays** drop down in the top left of the window and select **Optical Data Display** to open it as a tab.
   b. Click the **Image Transfer** button towards the top of the tab to begin streaming live camera feed from the InertiaCam.
   c. Next click the **VINS Image** (not supported with embedded settings) button to show the features recognized by the VINS algorithm. **Figure 20** shows the VINS Image while viewing an assortment of items.

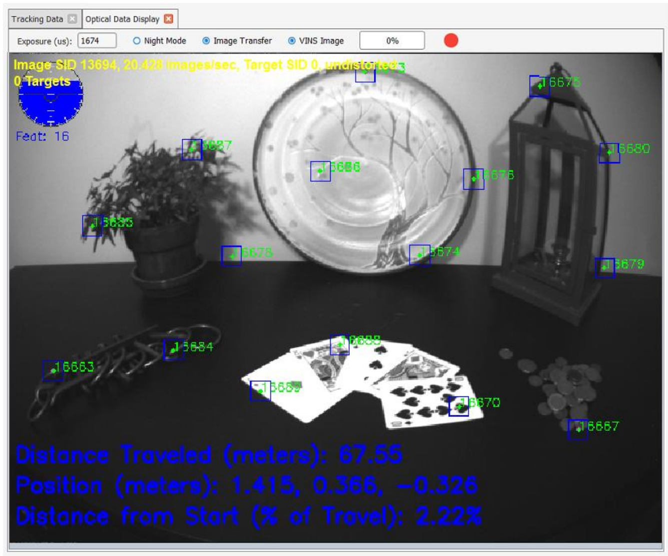


Figure 20 – VINS Image

2. While the NFT is in State 1, position and orientation will be relative to the starting position and orientation. To view the position and orientation of the tracker relative to the starting position, use the **3D Data Display** shown in **Figure 20**.
   a. Go to the **Displays** drop down and select **3D Data Display** to open that up as another tab.
   b. While in State 1, the starting position is located at the center of the grid. The tracker is represented by the point with XYZ axes. As the tracker rotates, the orientation of the axes change. Changes in position leave behind a yellow trail.
   c. Left clicking on the graph will orient the view, right clicking can be used to pan, and the center wheel can be used to zoom in and out.

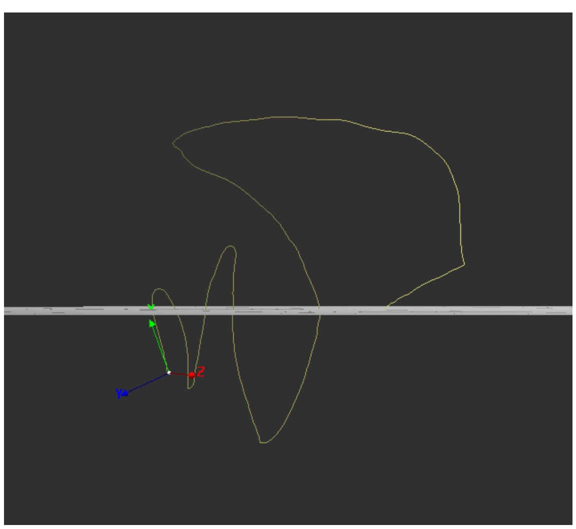


Figure 21 – 3D Data Display

3.4 Tracking using Fiducials in Tracking State 2

1. NFT must view a fiducial constellation to get the Tracking State into **State 2 (Tracking, Referenced)**. To enter State 2 using the default constellation, begin by placing the poster on a level surface.

2. With sfStudio running and the InertiaCam connected and tracking in State 1, position the InertiaCam so the constellation is well within the field of view. This can be checked in the Optical Display tab.
   a. Turn the image transfer toggle to **ON** (filled) and the VINS image toggle to **OFF** (unfilled).
   b. Recognized fiducials will be marked with colored symbology, as seen in **Figure 22**.

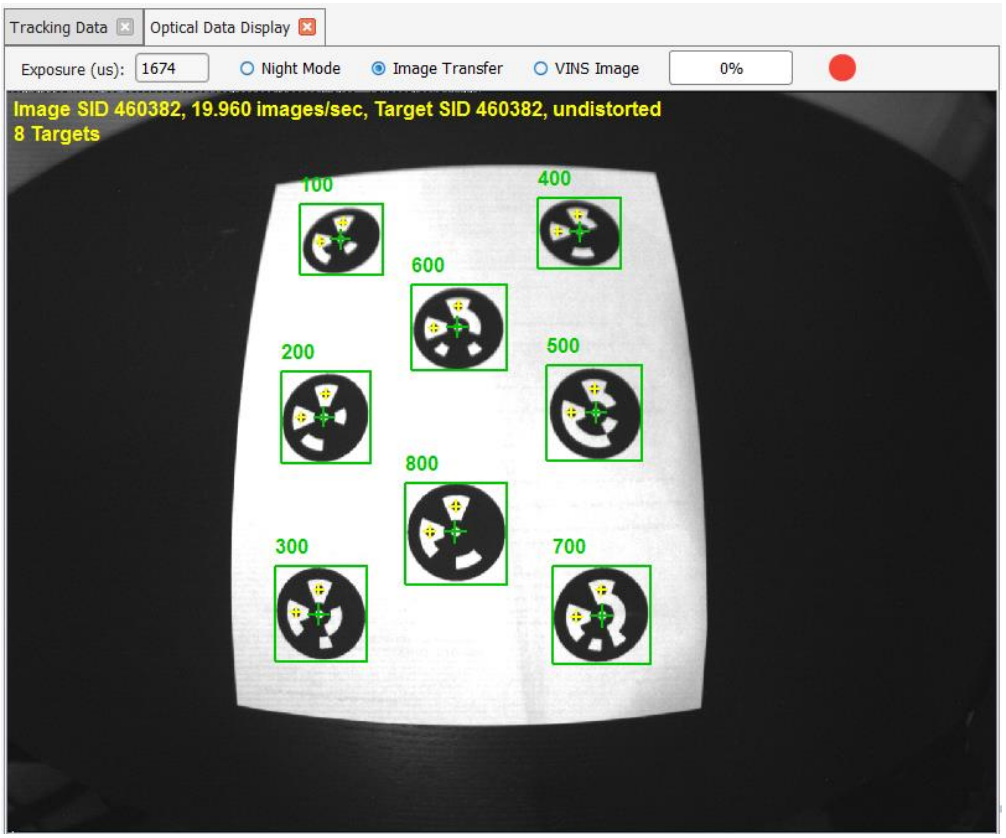


Figure 22 – Fiducial Overlay

3. When the IS-1500 is able to recognize the constellation it will enter State 2, and the icon will change from yellow to green. The Tracking Quality should also increase, as seen in **Figure 23**.

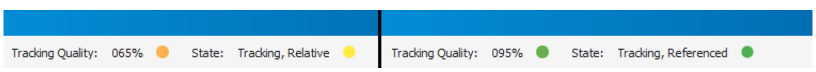


Figure 23 - Tracking State Changes

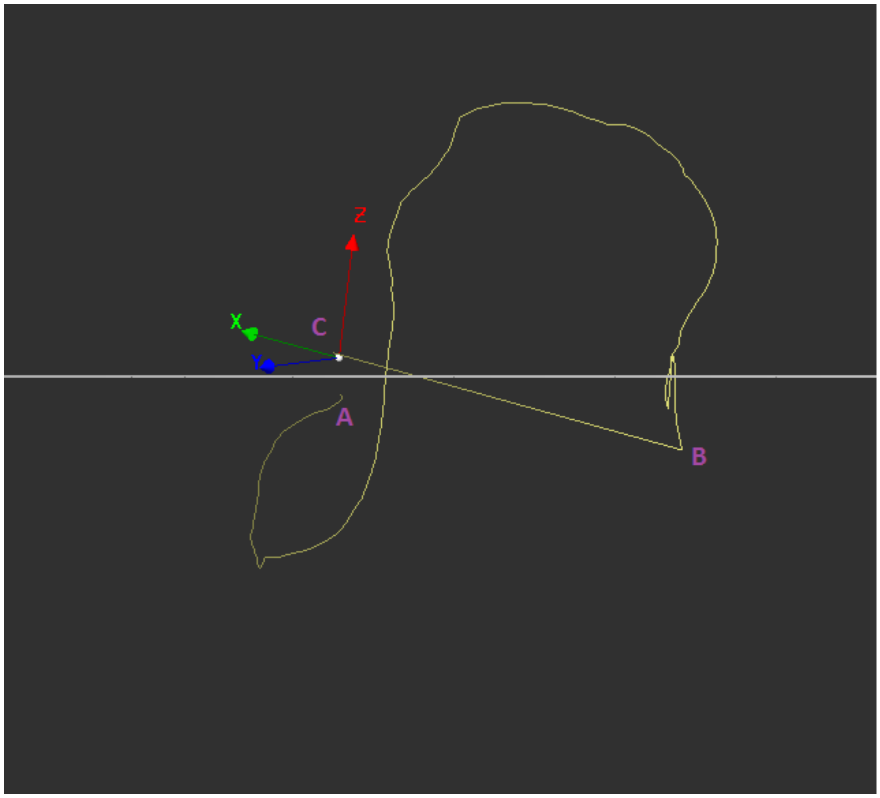


Figure 24 –Tracking State 2 Relocalization

4. The system will shift and relocalize such that the starting point will no longer be the origin. **Instead, position and orientation will now be reported relative to the origin of the fiducial constellation.** The relocalization jump from State 1 to State 2 can be viewed either using the data plots in the Tracking Data tab or the path in 3D Data Display. The latter is illustrated in **Figure 24**.

    a. The path begins relative to the starting position, near Point A.
    b. The InertiaCam is then moved to Point B to hover over and view the constellation.
    c. The straight diagonal line from Point B to the tracker's current position at Point C is the result of progressing from Tracking State 1 (Tracking, Relative) to State 2 (Tracking, Referenced).
        i. The tracker has been kept still since Point B – it is the coordinate reference frame that has shifted (like moving a sheet of paper from under a still pencil).
        ii. The distance from B to C is effectively the distance from the original starting point to the origin of the constellation.
        iii. **To summarize, the IS-1500's output of position and orientation has corrected itself from tracking based on its starting point without a global point of reference (Point B, State 1) to using a landmark it recognizes to determine its actual position and orientation relative to the world (Point C, State 2).**

In State 2, the InertiaCam will continue tracking using VINS as before in State 1. However, now anytime the constellation comes back into view, the position and orientation will perform another relocalization, correcting any accumulated drift from using VINS alone. Moving the constellation itself will result in another change of the coordinate reference frame. Therefore, for the purposes of drift correction to acquire accurate data, it is best to have a constellation in a fixed, stable position.

3.5 Using a Custom Fiducial Constellation

1. The process of creating a fiducial map that will ensure accurate tracking quality is covered in the Fiducial Constellation Setup Guide. If needed, contact Thales Visionix Technical Support for further information on how to create a constellation.

2. Once the custom constellation is in place, it will need to be mapped before the IS-1500 can track off of it. The map informs the IS-1500 system what fiducial IDs to expect and at what placements. One of the fiducials can serve as the origin of the constellation. The origin also can be easily edited to be any point once the map has been created using the sfStudio Constellation Display tools. It is important to note that the Simple Visual Mapping tool used to generate map files can only run on Windows operating systems.

3. The map file is saved as the **environmentPSEs.cfg** in the **sfStudio** subfolder. Start by creating a copy of the current file and renaming it, for example, to **FidPoster.cfg**. The system requires **environmentPSEs.cfg** to be the name of any map in use, and the tool used for mapping will overwrite any file that currently has that name. **It is therefore very important that an existing map file be copied and saved under a different name before another map file is created.** Keeping backups of constellations allows the users to revert to a previous constellation or switch between them without needing to recreate the map files.

4. With sfStudio connected to the InertiaCam, select the **Simple Visual Mapper** in the **Tools** dropdown of **sfStudio**. From here, the **Simple Visual Mapper Tool** section will cover the process of mapping the custom constellation. See also the **sfStudio User Guide (MNL-0020)** for more details.

5. When mapping is complete, the SVM tool automatically saves it on your computer.

6. If the IS-1500 is able to reach State 2 using the new map, it is recommended that a backup of this custom map file also be created.

4 SOFTWARE

4.1 Software Flow Chart

The Introduction section provided a brief overview of the software architecture. Refer to the flowchart provided in **Figure 2** as needed.

4.2 sfHub

The sfHub software is a console executable that is responsible for calculating tracking data from the images and IMU records provided by the InertiaCam. When run, it establishes a connection to the InertiaCam and begins streaming in the optical data from the camera and inertial data from the IMU. This data is run through the VINS and PRA threads described in the Understanding Tracking section to calculate tracking data. The tracking data is then output by sfHub and passed through sfAccess to sfStudio or the end user application.

The communication between sfHub and sfAccess is established over UDP and TCP. In some server-like set-ups, one computer (a **host**, or **server**) can be used to run sfHub while another (the **client**) runs sfStudio or the end user application. This serves to take the computational load of calculating tracking data off of the master computer. It is also useful in scenarios that require tracking to take place in one location and analysis to be performed elsewhere. For more information on sfHub and sfAccess to allow for this kind of set-up, refer to the Changing TCP and UDP Settings in sfAccess.ini and sfHub.ini subsection of the Advanced Settings and Configuration Files section. Be sure to read through the Introduction and Editing Configuration Files sections for warnings and instructions before attempting to change the configurations. It should also be noted that image transfer will not be possible in this type of configuration, as this is only available when sfHub is run locally.

In order for sfStudio or the end user application to receive tracking data, sfHub needs to be running in the background to generate the data. When using sfStudio, sfHub is automatically launched by the **Connect** button, as will be discussed in the sfStudio section. However, end user applications will need to either launch sfHub themselves or else sfHub will need to be launched by the user manually prior to running the application.

4.3 Interfacing with Tracking Data

For the majority of users, the InterSense library provides the most simple and effective approach to integrating tracking data from the IS-1500 into the end user application. The library is distributed as iSense.dll for Windows and as libisense.so for Linux. The InterSense library is functional with most InterSense tracking devices, including the 3-DOF InertiaCubes and 6-DOF IS-900. It is largely backwards compatible with legacy versions, requiring minimal to no software changes to successfully switch between tracking hardware. However, a number of changes have been made to the InterSense library for use with the IS-1500. A list of these changes can be found in the **InterSense Developer Guide** (MNL-0015), along with further information on use of the InterSense library.

The InterSense library is specifically created for ease of use and compatibility with the other InterSense trackers. While this is beneficial for the majority of users, it also limits the degree of IS-1500-specific functionality the InterSense library can reasonably incorporate. Some advanced users may find the InterSense library restrictive and unable to provide the low level data required by their applications. The sfAccess library provides more expansive functionality for the IS-1500, but is only recommended for use in applications that specifically require its low-level access. Further information on the sfAccess can be found in the **IS-1500 Advanced Developer Guide** (MNL-0025).

The third method of interfacing with tracking data is intended for users working with Unity applications. A Unity plug-in is provided with the software package for simple integration. The Unity plug-in uses the InterSense library as a foundation and shares the same level of functionality.

4.4 sfStudio

The sfStudio software is primarily used for the set-up and analysis of IS-1500 tracking. It is an excellent tool for viewing tracking data or troubleshooting the system. There are several tools and displays of particular importance to the IS-1500:

- sfStudio Toolbar
- sfStudio Sidebar
- Tracking Data Display
- Optical Data Display
- 3D Data Display
- NFT Sigmas Display
- Make Fiducials Tool
- Simple Visual Mapper Tool

Each of these will be briefly covered in this section relative to their functionality with the IS-1500. It is important to note that sfStudio is not only used with the IS-1500 and may have some functions or displays that are not intended for use with the IS-1500. The features that are functional with the IS-1500 are not limited to those explained here.

For additional information on sfStudio and its use with the IS-1500, reference the **sfStudio User Guide (MNL-0020)**.

4.4.1 sfStudio Toolbar

The main sfStudio toolbar is located at the top of the window. There are three features here of note for the IS-1500, boxed in red in **Figure 25**.

Figure 25 – Main Toolbar

The button to the far left is the **Connect** button. This will automatically launch sfHub and establish a connection to the InertiaCam. This can also be done by selecting **Connect to Devices** from the **File** dropdown menu. Some tools or displays will not be available or functional until sfStudio is connected to the InertiaCam.

The Tracking State indicates which one of the following four modes the tracker is in.

0. **No Communication (Grey)** – No communication with sfHub and/or the InertiaCam.
1. **Tracking, Relative (Yellow)** – The IS-1500 is connected and tracking, but has not yet acquired optical lock on a fiducial constellation.
2. **Tracking, Referenced (Green)** – This status is only reached after optical lock has been acquired on a constellation and the tracker's position and orientation relative to a predetermined world reference frame are known.
3. **Lost (Red)** – The InertiaCam is connected, but a divergence or other anomaly has been detected. The InertiaCam will not stay in State 3 for long, as this mode is usually associated with a filter reset back to State 1.

The Tracking Quality metric, or TQ, indicates how god tracking is overall. A mid-range TQ does not necessarily indicate poor data, however. TQ is zero for States 0 and 3. In state 1 or 2, TQ is a function of pose uncertainty and feature (or target) count. Uncertainties are typically lower in State 1 than in State 2, so TQ will typically also be lower in State 1.

4.4.2 sfStudio Sidebar

Figure 26 – Sidebar

The sfStudio sidebar typically occupies the left-hand pane and is divided into four panes. Once the software is connected to the InertiaCam, its data will populate as shown in **Figure 26**.

The top pane indicates that data is currently being viewed for the connected tracker. This can change if a logged data file is instead being viewed.

The second pane from the top shows an animated 3D model of the InertiaCam. The orientation of the model reflects the current orientation of the tracker and is useful for quick reference.

The third pane from the top provides information on the software versions currently in use.
The bottom pane provides connection information. When the connection is stable, TCP should read as OK as shown. The data records per second (r/s) are shown in blue. The IS-1500 will normally receive overall about 400r/s, with about 200 IMU r/s. When fiducials are in the field of view, 20 FID r/s is normal.

4.4.3 Tracking Data Display

This is the default data display tab for the IS-1500. When the connection to the InertiaCam is established, it will automatically appear and begin populating with data. This tab provides the current values and historical plots for the InertiaCam orientation, position, and linear velocity.

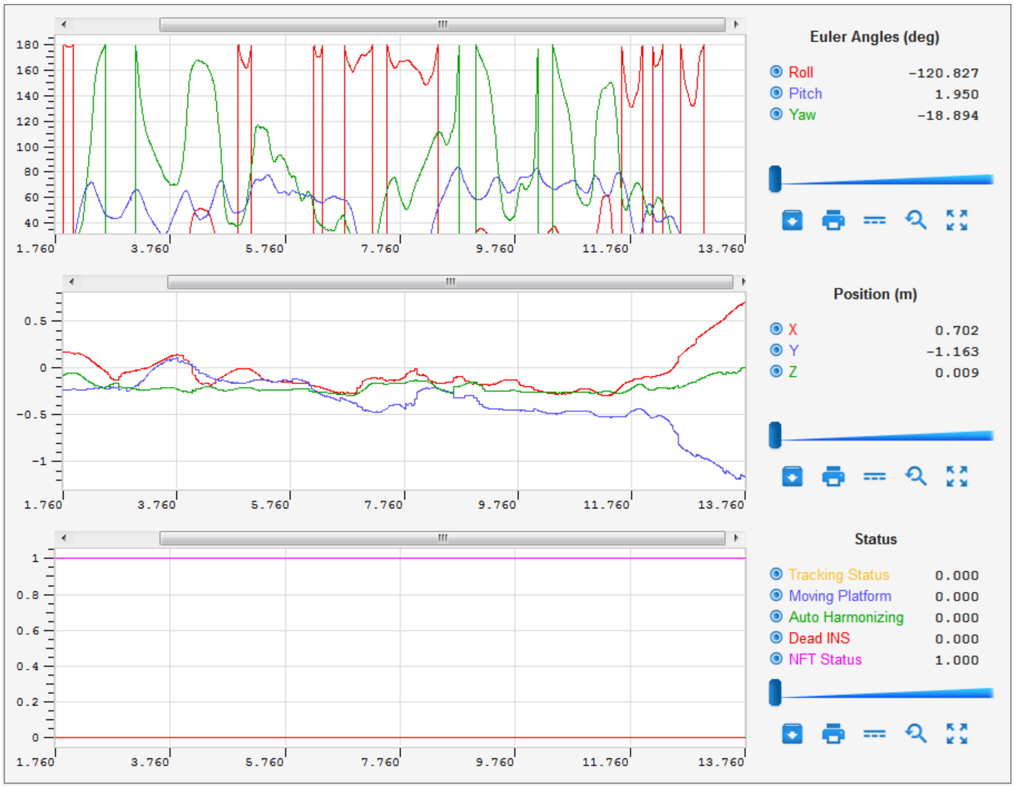


Figure 27 – Tracking Data Display

4.4.4 Optical Data Display

The Optical Data display is used to view the live camera feed from the InertiaCam. As shown in the Understanding Tracking and Installation sections, it can also be used to show overlays for recognized fiducials and natural features.

By default, the InertiaCam is set to use auto-exposure. The current exposure is shown on the left-hand side of the Optical Data toolbar. Auto-Exposure can be temporarily disabled by using the toggle to the right of the exposure. When using manual exposure, the exposure reading can be overwritten to the desired exposure, which will take effect after pressing the Enter key. To make manual exposure the default or to change the auto-exposure settings, see the InertiaCam Exposure Settings subsection of the Advanced Settings and Configuration Files section.

Image Transfer toggles the streaming of camera footage from the InertiaCam. VINS Image toggles the overlay of natural features found by the VINS algorithm. When VINS Image is enabled, fiducial overlays will be disabled. In **Figure 28**, the top image shows Image Transfer while the bottom shows the VINS Image of the same scene. Note that when using embedded settings, VINS image generation is disabled to conserve processing power, so this toggle will only display a black image.

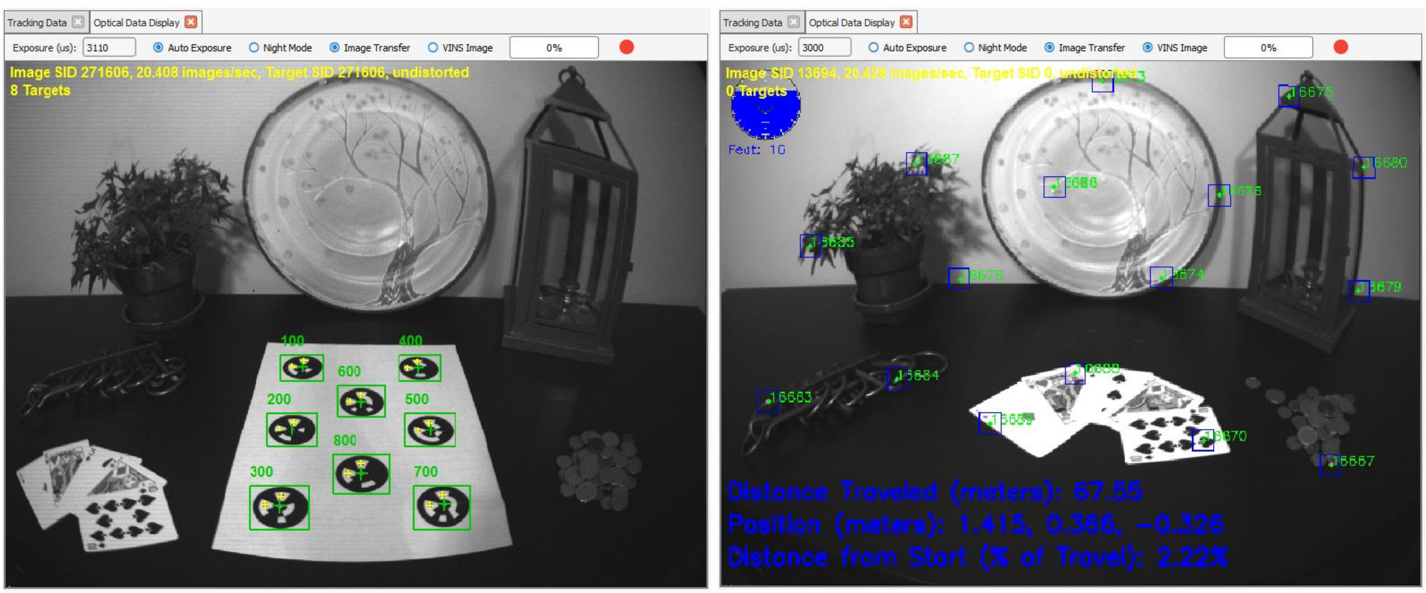


Figure 28 – Optical Data Display

4.4.5 3D Data Display

The 3D Data Display provides an orthographic view of the tracker position and orientation relative to the world reference frame. The tracker position is represented by small grey orb with colored X, Y, and Z axes showing its orientation. A thin yellow path shows the path of the InertiaCam. The grey grid shows the XY plane of the world reference frame. Initially, the 3D Data Display is an overhead view of the grid, with +X facing upwards and +Y to the right.

When using a mouse, the view can be rotated by holding the left mouse button, panned by holding the right mouse button, and zoomed using the scroll wheel. The 3D Data Display also supports multi-touch and can also be manipulated when using a tablet or touchscreen. A single touch will rotate the view, while dual touch swiping or pinching can be used to pan or zoom the image.

The Reset button in the 3D Data Display tab below will remove the existing path in order to de-clutter the display.

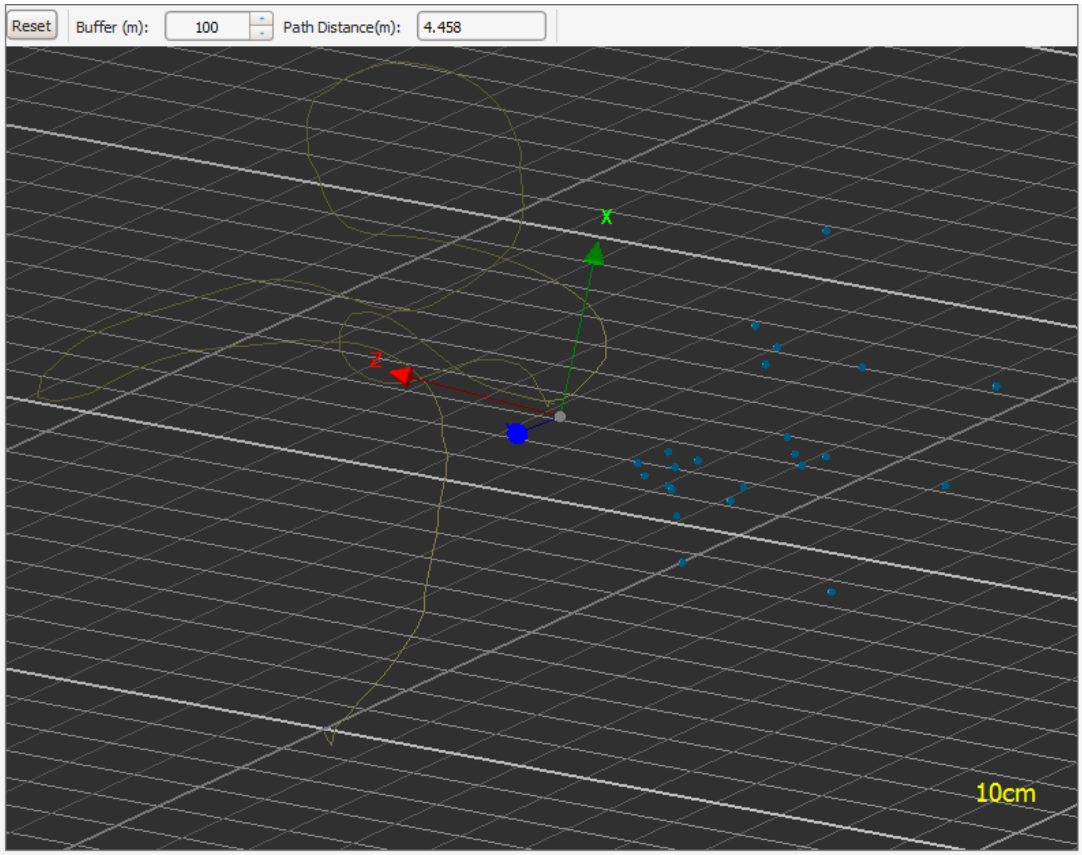


Figure 29 – 3D Data Display

4.4.6 Tracker Sigma Data Display

The sigmas represent the degree of uncertainty of orientation and position. A low value is optimal, while high values can indicate poor tracking. The Tracker Sigma Data display shows the current sigma values and their historical plots.

Note that in the plot shown in **Figure 30**, the yaw has a higher rotation sigma, and thus a degree of uncertainty, than pitch and roll. Additionally, the yaw and position sigmas are all steadily increasing as the system continues to use only VINS for tracking. The sudden drop in sigmas at about 618s occurred when the system had relocalized having viewed the constellation. These occurrences correlate with what was learned in the Understanding Tracking section about yaw and VINS drift.

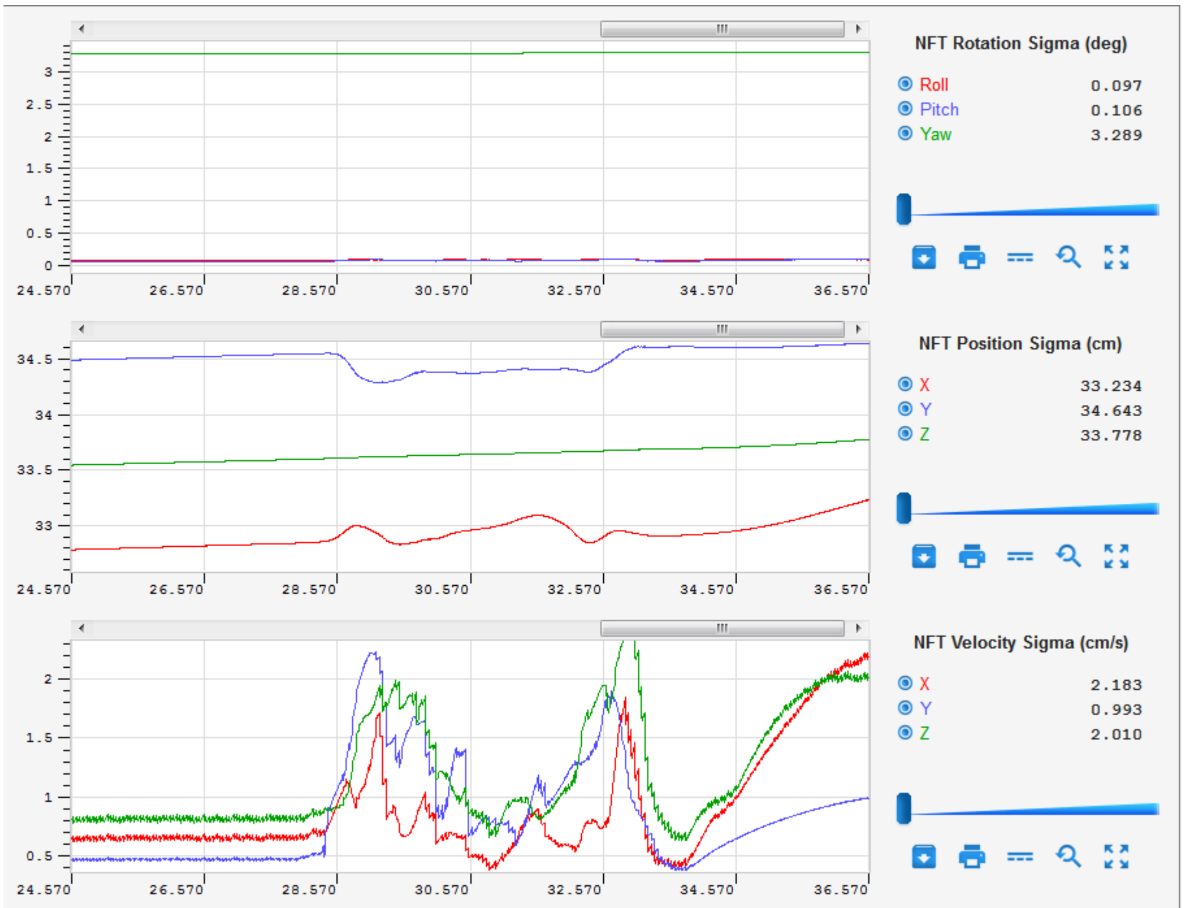


Figure 30 – NFT Sigmas Display

4.4.7 Make Fiducials Tool

The Make Fiducials tool can be used to generate fiducial patterns for those interested in creating custom constellations. The InertiaCam is a visible light tracker, so the images produced by this tool can be printed on any white surface. The Fiducial Constellation Guide provides instruction on how to create a constellation.

All fiducial IDs must be divisible by 10. This is because each eye is identified by the system as x0, x1, or x2. For instance, for fiducial ID 990, the center eye would be known to the system as 990 and the two outer eyes would be 991 and 992. Traditionally fiducial IDs are multiples of 100, so the default settings use a Fiducial Starting ID and Fiducial ID Increment of 100.

For use with the IS-1500, Invert Color and Center Mark must remain disabled when creating fiducials. For the IS-1500 to recognize fiducials, the center eye should be white and should not contain a crosshair at the center, like the one in Figure 31.

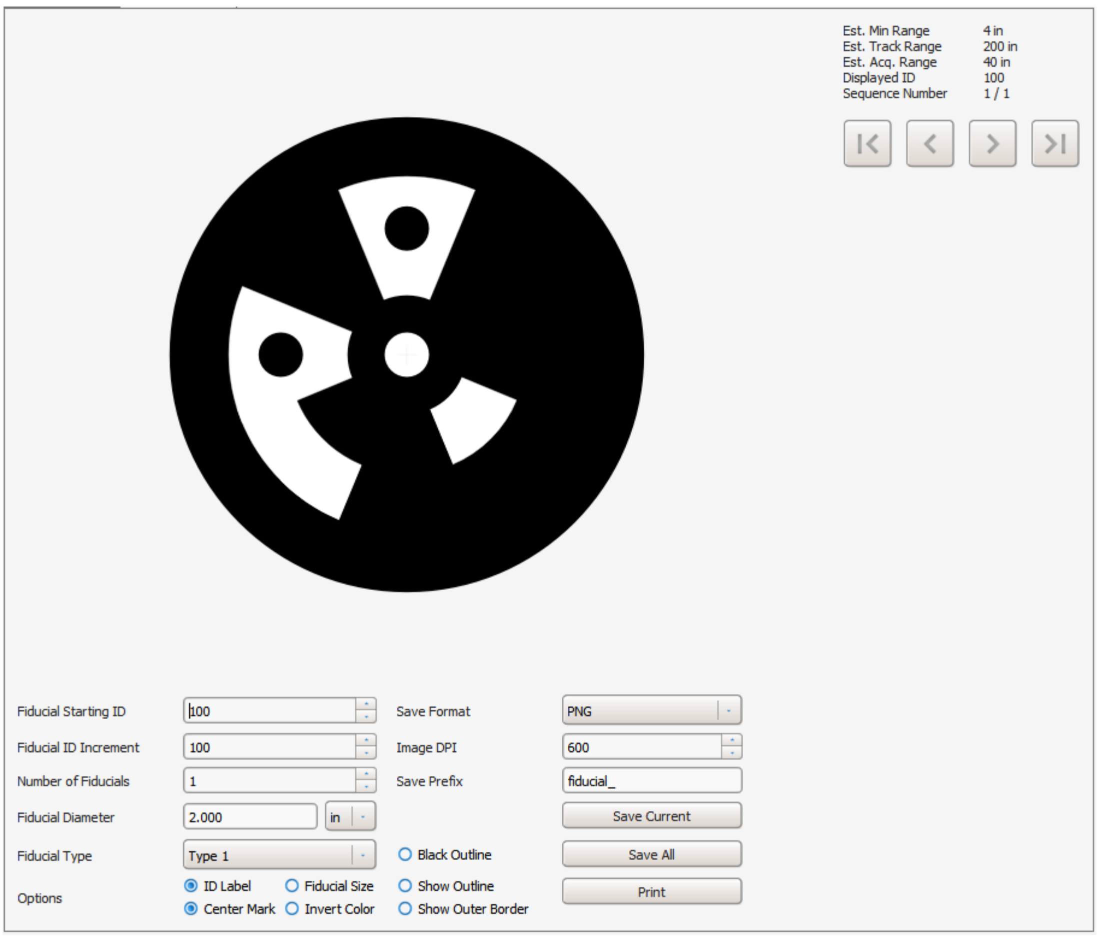


Figure 31 – Make Fiducials Tool

4.4.8 Simple Visual Mapper Tool

After a custom constellation has been created, it must be mapped using the Simple Visual Mapper (SVM) before it can be used by the system. The map file provides the system with information on the expected fiducials and their locations and orientations relative to the world reference frame. Fiducials with IDs that are not specified in the map file will not be used by the system for tracking.

At the side bar of SVM, there is a drop down with constellation presets that will autofill the rest of the values. Among them is the 8.5 x 11 Poster preset, which represents the IS-1500's default constellation- the poster that is included with the kit.

When the SVM mapping process reaches 100% completion and the map file is saved, a backup of the map will be saved in the Constellations folder of the sfStudio subdirectory. A prompt will appear to ask if the constellation should be used now. If 'No' was selected, the map can be used later by opening it in the Constellation Display and using the Send to Tracker button. Additionally, the default map can be found in the ExampleConstellation folder of the ConfigurationFiles subdirectory. **Any time the constellation file is changed, sfHub must be restarted.**

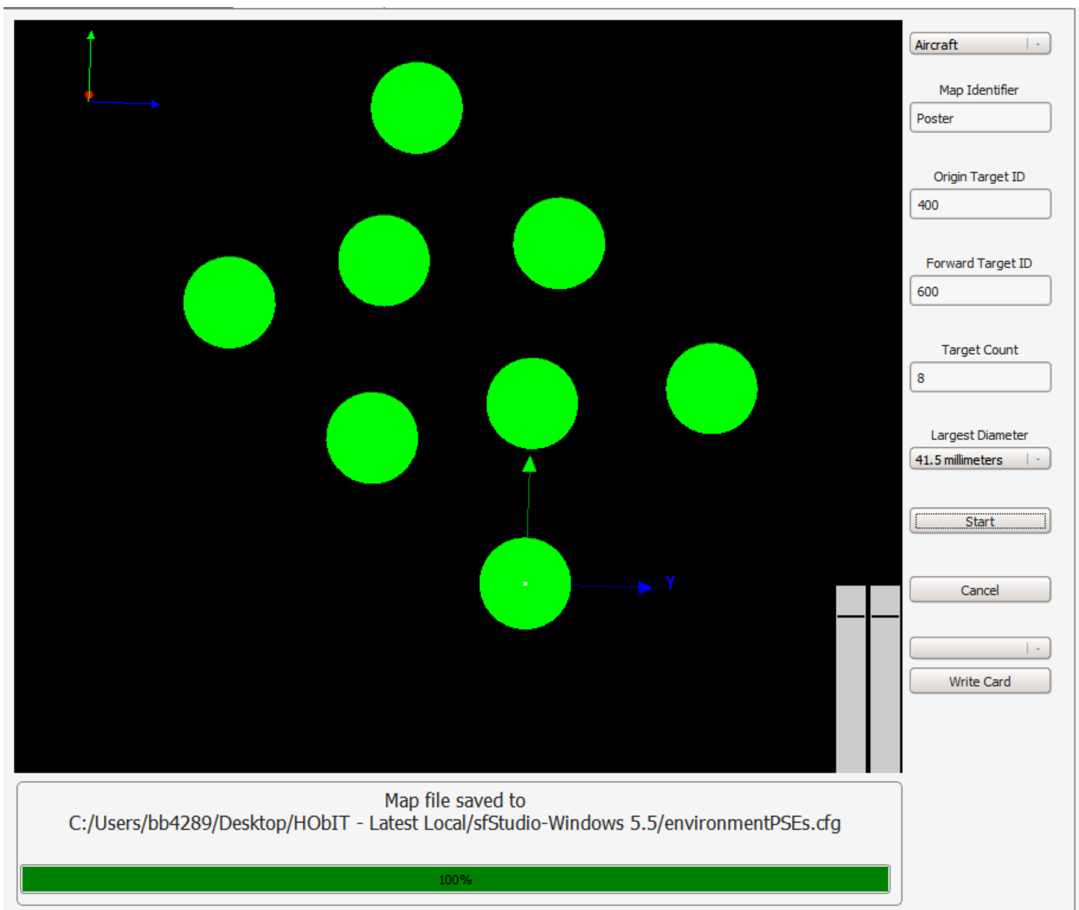


Figure 32 – SVM Tool

5 ADVANCED SETTINGS AND CONFIGURATION FILES

5.1 Introduction

This section explains the advanced settings found in some of the key software configuration files. The configuration files covered in this section include:

- sfHub.info
- sfAccess.ini
- sfStudio.ini
- SVM.ini

There are a few important warnings that should be noted before proceeding:

- **If at any time the configuration files need to be reverted to their defaults, copies of the default files can be found in the Configuration Files folder of the software directory. These files should never be overwritten.**
- **These settings are for advanced users only. Changing configuration files improperly can have severe impacts on the performance and operation of the system.**
- Generally, unless directed here by the guide or by a Thales Visionix technical support engineer, these files should not need to be changed from their original state.
- Any settings in the configuration files not covered in this section should be changed only by Thales Visionix engineers.
- It is recommended that a backup file of the current configurations be created in a folder outside of the main installation directory before making any changes.
- If the changes made to the file are intended to be permanent, further backups of the working edited files should be created.
- **Updates to the software may change the content of the configuration files.**
  - Any backups of configuration files saved in the installation directory may by overwritten or deleted by a software update.
  - After updating the software, rather than rolling back to an older configuration file, manually edit the newly installed configuration files to match the system to ensure updates do not get overwritten.
- If there is any confusion on how to change a configuration file, do not make any changes and instead contact Thales Visionix Technical Support using the contact information on **Page 2**.

5.2 Editing Configuration Files

Each of the files discussed in this section can be viewed and edited using a standard plain text editor, such as Notepad in Windows. After changes to the file have been made, it must be saved and the software restarted for the changes to take effect.

When reading configuration files, any text following a semicolon (;) is considered a comment and is not actually read by the system. Comments are used as notes on what the values represent or can be used for organization. In the sample files and example text presented here, comments will have green text. Comments can be edited or added without affecting the system in any way, but they must always be preceded by a semicolon. When changes from the default configuration are made, it is even recommended that they by commented by the user for future reference

Each setting has a variable followed by a value. Only the values should ever be changed – editing variables will render them unrecognizable by the software and could compromise the system. In example text, when a value should be changed (or has been changed) it will be marked in red.

When editing configuration files, only text marked green or red in the examples here can be changed without invalidating the configuration.

5.3 sfHub.info

The **sfHub.info** file is located in the **sfHub** subfolder of the software directory. It is broken up into four sections- General, IS1500, Vins, and FusionCore.

5.3.1 Logging and Playback of Tracking Data

In addition to the methods sfStudio uses for capturing tracking data, sfHub has a built in logging function that collects image and IMU data directly from the InertiaCam as it's received. This data can be used to replay tracking runs for experiments, simulations, presentations, or any other applications that require repeatable data. Once enabled, a new log is collected each time streaming is started in sfHub.

As a warning, this logging function is not to be confused with that which creates a text file (sfHub.log) indicating program errors, etc. This process saves each frame as an image rather than as a video to prevent encoding interfering with playback. While this ensures that the data captured is faithful to the original tracking data and is repeatable, it also means that log files tend to be large. **It is recommended that several gigabytes of hard drive space are free before collecting a number of logs. It is also very important to disable logging once the desired data has been collected.** The data is not overwritten when sfHub is restarted and, if it's not disabled, will continue to consume hard drive space until the hard drive is full.

Also, logging is more computationally intensive than tracking. **It is recommended that prior to and while logging tracking data, background and end-user applications are closed and only sfHub and sfStudio are run. For optimal performance, do not perform any tasks while sfHub is running with logging or playback enabled, including mouse clicks or typing.** This is because of the way sfHub processes incoming data. If the data queue is temporarily overloaded due to processing limitations, IMU records or image frames will be dropped. While not an issue during live tracking, dropping records during playback can cause a butterfly effect that will change the outcome of the otherwise reproducible replay.

A. To collect a log, follow the steps below.

1. If they are currently running, close the sfHub and sfStudio programs.
2. Open the **sfHub.info** file.
3. Under the **General** section, find the **Master** subsection and set **logData 1**.

```
General
{
    appMode     1   ; (1) Tracking (2) Playback (3) Configuration Utility
    numSensors  1   ; (1) Min (4) Max
    verbosity   0   ; (0) Low (1) Med (2) High

    Master
    {
        ; (1) sfCore 3DOF
        ; (2) sfCore 6DOF
        ; (3) NFT + Fid
        ; (4) NFT + GPS
        ; (5) NFT + GPS + Fid
        trkMode 3

        sensorType 2 ; (1) HObIT (2) InertiaCam
        initState  1 ; (1) Streaming, (0) Paused
        logData    1 ; (1) Enable data logging
    }
}
```

4. Save and close **sfHub.info**.

5. Now, any time streaming is set, a new log file will be created. It is possible for this to happen multiple times during a single sfHub session, especially since actions like connecting to sfStudio will start and stop streaming of data.
    a. Using sfHub by itself will be cleaner and ensure only one log is created, but will not allow the user to view the data as it is being collected.
        i. To do this, run the **sfHub** program directly from the sfHub folder that sfHub.info was found in.
        ii. Allow a few seconds for sfHub to find the tracker and begin tracking (see **Figure 33**).

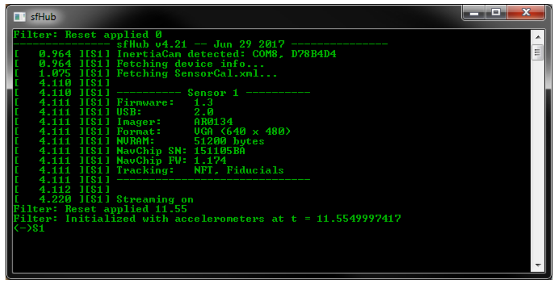


Figure 33 – sfHub Tracking

        iii. When the sfHub output appears as it does in **Figure 33**, with Streaming On, the Filter Initialized, and the whirligig next to S1 turning freely, tracking is running and the data collection has begun. Perform the tracking run to collect the desired data.
        iv. The log is generated continuously as sfHub is streaming and does not need a special save process. When finished with data collection, simply close the sfHub window.
    b. Using sfStudio will allow a user to look at the data as it's begin collected, but has the potential to create small, throwaway log files that will need to be discarded manually from the system later.
        i. Open **sfStudio**.
        ii. If you would like to watch any displays while tracking (such as 3D Data display or Optical Data Display), open these and drag them off to their own windows now. (These displays only need to be open for convenience, all displays will be usable when the log is played back, not only the ones currently in use.)
        iii. Click the **Connect** button to automatically launch and connect to **sfHub**.
        iv. Perform the tracking run to collect the desired data.
        v. The log is generated continuously as sfHub is streaming and does not need a special save process. When finished with data collection, simply close sfStudio and sfHub.
6. When the desired data logs have been collected, **remember to disable logging** by repeating steps 1-4, this time setting **logData 0**.

The log files are automatically generated in the **logs** subfolder of the sfHub folder. They can be found in **./sfHub/logs/<InertiaCam Serial Number>/<Date/Time Stamp>.** For instance, if a log was created using an InertiaCam with serial number ICM-1705998-3 at 12:52PM on July 11, 2017, this log would be the **./sfHub/logs/ICM-1705998-3/2017-07-11_12.52.23PM** folder. If sfStudio was used to create logs, the small throwaway log folder can usually be identified by looking at the folder size. Those with drastically smaller file sizes than the others are unlikely to contain the data that was collected. This can always be confirmed by simply playing back the log files.

B. To playback a log file, follow the steps below.

1. If they are currently running, close the sfHub and sfStudio programs.
2. If planning on using sfStudio to view data while the log is replaying, follow these added steps.
    a. **Auto Launch** for **sfHub** will need to be **disabled** by following the instructions in the sfStudio.ini section.
    b. Open **sfStudio** and any data displays to be viewed during playback. Drag each display tab into its own window, as clicking around during playback is inadvisable.
3. Open the **sfHub.info** file.
4. Under the **General** section, set **appMode 2**.

```
General
{
    appMode       1    ; (1) Tracking (2) Playback (3) Configuration Utility
    numSensors    1    ; (1) Min (4) Max
    verbosity     0    ; (0) Low (1) Med (2) High
```

5. In order to allow the image transfer streaming, under the **Communication** section, set **imgTx 1**.

```
Communication
{
    Master
    {
        ; Serial port for sensor connection
        ; Windows     - COMx
        ; Linux       - /dev/ttyACMx
        ; AutoDetect - auto
        serialPort auto

        imgTx      1                  ; (1) Enable image transfer
        imgTxMode 1                   ; (1) Shared mem (2) UDP
        tcpPort    51717              ; TCP port for communication with sfAccess
```

6. Save and close **sfHub.info**.
7. Run the **sfHub** program manually from the same folder sfHub.info was found in.
8. When sfHub opens, it will be awaiting user input. Before proceeding in sfHub, connect to sfHub in sfStudio or other end user application. It is normal for sfHub to not show any indication that it has connected to any program and for sfStudio (or other end user application) to not show tracking data yet.
9. **sfHub** will show a list the InertiaCam serial numbers logs have been recorded using. Enter the line number for the device, then the line number for the desired log. **Figure 34** follows the example provided earlier of a log a log was created using an InertiaCam with serial number ICM-1705998-3 at 12:52PM on July 11, 2017, located at the **./sfHub/logs/ICM-1705998-3/2017-07-11_12.52.23PM** folder.

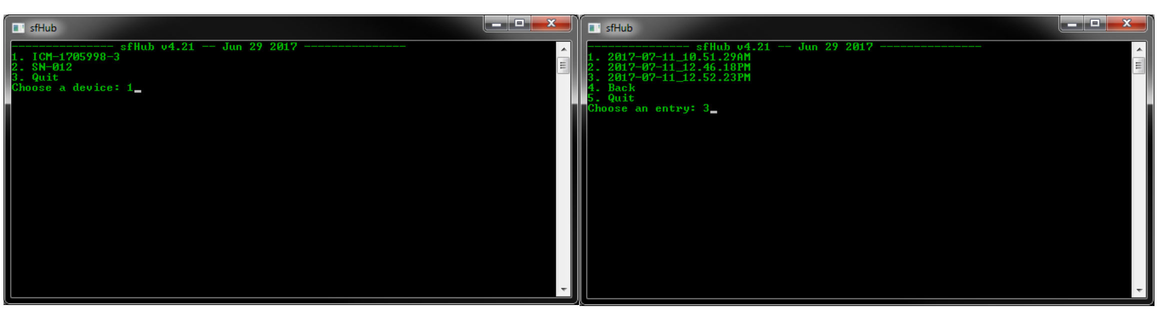


Figure 34 – sfHub Playback Menu

10. After the second entry is made, sfHub will begin running playing the log and data will be streamed to sfStudio or the end user application.

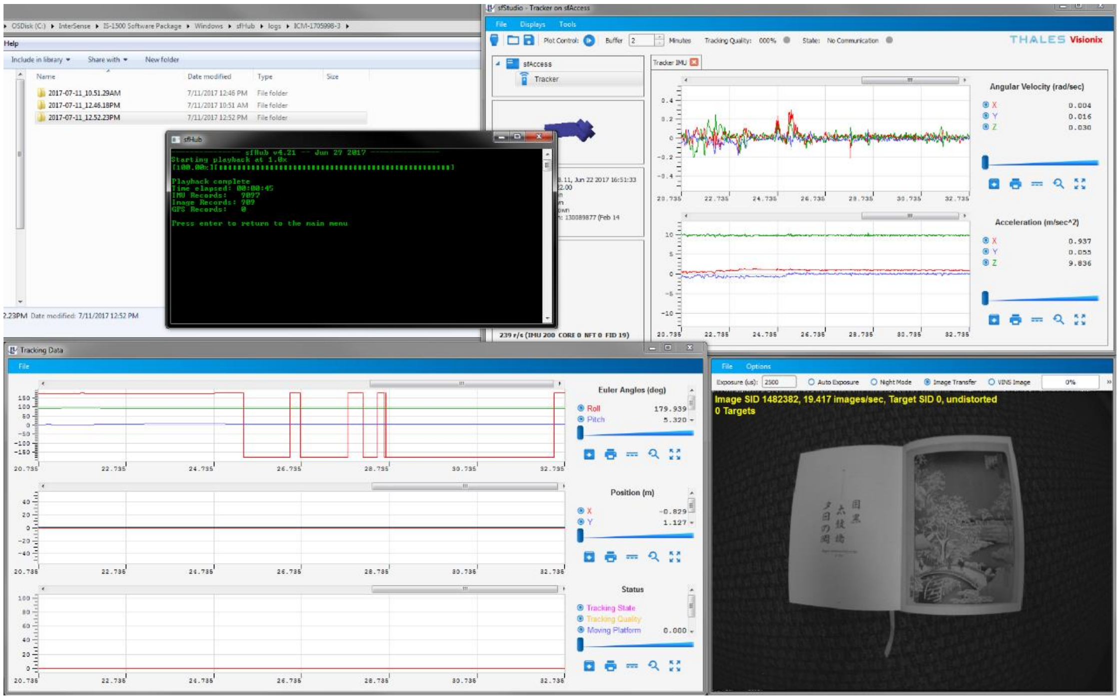


Figure 35 – Log Playback

5.3.2 Image Resolution in the Configuration Utility

The Configuration Utility mode of sfHub allows a user to access some of the hardware settings on the InertiaCam. Most of these should not be changed without the assistance of Thales Visionix technical support. One important setting for tracker performance can easily be changed here by a user, though- the resolution of the InertiaCam images. By default, InertiaCam's are now shipped with 640x480 resolution, which uses less CPU while maintaining tracking performance. This makes it more efficient for tracking on host computers with lower processing power and decreases the size of log files to a reasonable size. Some applications may require the output or use of higher resolution images, however.

Use the following steps to change the output resolution of the InertiaCam.

1. If they are currently running, close the sfHub and sfStudio programs.
2. Open the **sfHub.info** file.
3. Under the **General** section, set **appMode 3**.

```
General
{
     appMode      3    ; (1) Tracking (2) Playback (3) Configuration Utility
     numSensors   1    ; (1) Min (4) Max
     verbosity    0    ; (0) Low (1) Med (2) High
     replaySpeed 1.0 ; Playback speed multiplier
```

4. Save and close **sfHub.info**.
5. Run the **sfHub** program manually from the same folder sfHub.info was found in.
6. When sfHub opens, it will be awaiting user input. Enter **1** to connect to the InertiaCam.
7. Select which InertiaCam port to use by entering the line number it appears on.
8. Enter **4** to view the Video menu, then **3** to select Set Image Format.
9. Enter **1** to set the InertiaCam to use 1280x960 resolution, or enter **2** to use the default 640x480 resolution. The line with the asterisk and the number in brackets indicate the current setting. In **Figure 36**, the current setting is 1280x960, and entering **2** will change it to 640x480 resolution.
10. Back at the Video menu, select **5** to reset the device and apply the changes.
11. Close **sfHub**.
12. Unplug and re-plug in the InertiaCam to the host computer.
13. Repeat steps 1-4, this time setting **appMode 1** to return sfHub back to tracking mode.

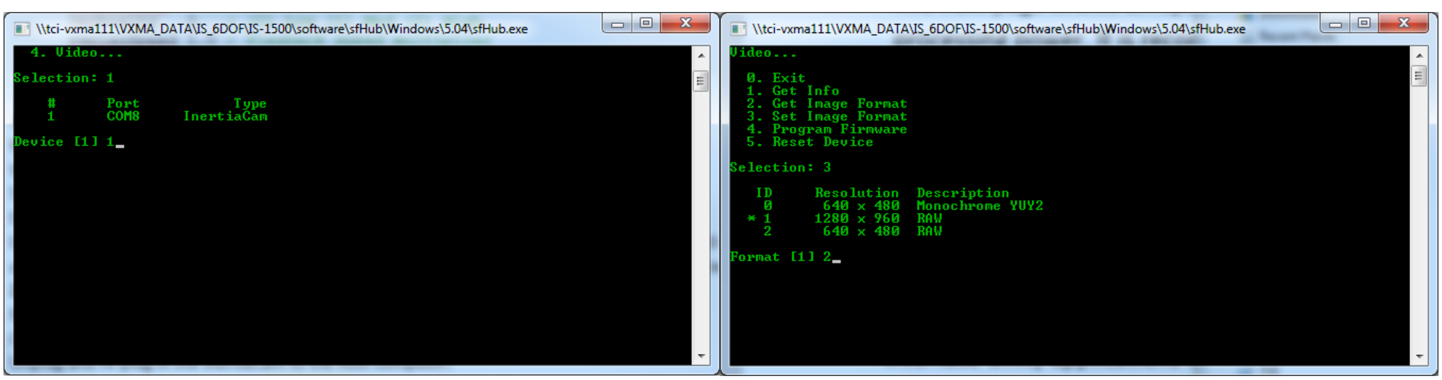


Figure 36 – Setting InertiaCam Image Resolution

5.3.3 InertiaCam Exposure Settings

The InertiaCam is set by default to use auto-exposure with a range of 100 to 15,000µs. In specific use scenarios, the auto-exposure range may need to be adjusted, or manual exposure set by default. The exposure should not exceed 15,000µs. If it appears that exposure over 15,000µs is required, the physical lighting in the tracking area should be increased instead. If the auto-exposure changes appropriately but is consistently too dark or too light, the luminosity should be changed.

A. To change the auto-exposure range, follow the steps below.

1. If they are currently running, close the sfHub and sfStudio programs.
2. Open the **sfHub.info** file.
3. Under the InertiaCam section, find the Master subsection and the **autoExposureMin**, **autoExposureMax**, or **autoExpsureLum** variables.

```
InertiaCam
{
 Master
    {
        ncDataDiv        5      ; NavChip data rate divisor (rate = 1000/divisor in Hz)
        frameInterval    50000  ; Frame interval in microseconds.
        manualExposure   2500   ; Initial manual exposure
        autoExposure     1      ; (1) Enable (0) Disable
        autoExposureMin  100    ; Microseconds.
        autoExposureMax  15000  ; Microseconds.
        autoExpsureLum   1000   ; Auto exposure target luminosity
    }
}
```

4. Change the values to reflect the desired minimum and maximum auto-exposure values in microseconds. The example below shows a change to a narrower auto-exposure range, from 2000 to 12,500µs, and the target luminosity is 2000.

```
autoExposure    1      ; (1) Enable (0) Disable
autoExposureMin 2000   ; Microseconds.
autoExposureMax 12500  ; Microseconds.
autoExpsureLum  2000   ; Auto exposure target luminosity
```

5. Once this change has been made, save the file.

B. Switching to a manual exposure requires disabling the auto-exposure and setting a value for the manual exposure. To do this, follow the steps below.

1. If they are currently running, close the sfHub and sfStudio programs.
2. Open the **sfHub.info** file.
3. Under the InertiaCam section, find the Master subsection and the **autoExposureMin** and **autoExposureMax** variables.

```
InertiaCam
{
 Master
    {
        ncDataDiv        5      ; NavChip data rate divisor (rate = 1000/divisor in Hz)
        frameInterval    50000  ; Frame interval in microseconds.
        manualExposure   2500   ; Initial manual exposure
        autoExposure     1      ; (1) Enable (0) Disable
        autoExposureMin  100    ; Microseconds.
        autoExposureMax  15000  ; Microseconds.
        autoExpsureLum   1000   ; Auto exposure target luminosity
        illuminator      1      ; IR illuminator enable
    }
}
```

4. Set **autoExposure 0** to disable auto-exposure. Set **exposure** to the desired default value in microseconds. The example below uses 5000µs.

```
manualExposure   5000    ; Initial manual exposure
autoExposure     0       ; (1) Enable (0) Disable
autoExposureMin 100      ; Microseconds.
autoExposureMax 15000    ; Microseconds.
```

5. Once these changes have been made, save the file.

Once set to use manual exposure, the exposure setting can be temporarily changed using sfStudio. Auto-Exposure can be toggled on or off at any time in in the Optical Data display toolbar, where the current exposure value is displayed. By default, the exposure setting will be the value entered in the sfHub.ini folder. Figure 37 follows the example provided in Step 6 above and shows the exposure as 5000µs, boxed in red.

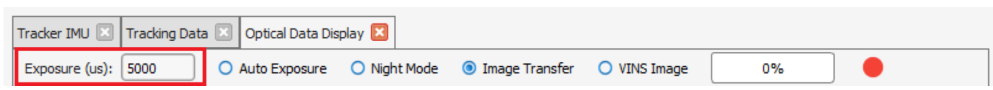


Figure 37 – Manual Exposure 5000µs

The exposure value entered in sfStudio will remain until the connection to the InertiaCam has been restarted or reset, or until the value is changed again. **Note that higher camera exposures lead to increased motion blur, which can impact tracking adversely. It is recommended that the lighting be such that the exposure remains under approximately 5 milliseconds.**

5.4 sfAccess.ini

The **sfAccess.ini** file can be found in two (or more) locations. The copy used for sfStudio is located in the **sfStudio** subfolder of the software directory. When the end user application is interfacing with the InterSense library or sfAccess library, the application's version of sfAccess.ini file must be changed instead (see the Interfacing with Tracking Data section). By default, each version of sfAccess.ini is identical. It is broken up into three sections- General, Communication, and Processing.

5.4.1 Boresight and Tip Offset

Boresight and **tip offset** define offsets from the default tracker orientation and position, respectively. For example, if a sensor is mounted at a 40° pitch at rest, a boresight in radians of (0, .698, 0) will set the system to output (0, 0, 0) for yaw, pitch, and roll when the tracker is level. Similarly, if the tracker is mounted 5cm from the center of a tablet in the +Y axis, a tip offset of (0, .05, 0) will change the (0, 0, 0) position of the tracker reference frame to be the center of the tablet.

The boresight and tip offset of the InertiaCam is typically defined in the end user application through function calls to the Intersense or sfAccess libraries. However, some applications may require or prefer setting these in the sfAccess.ini file. **Boresight and tip offset should be define in either the end user application or sfAccess.ini, not both.** Otherwise, the transformation will be applied twice. To change the boresight and tip offset, follow the steps below.

1. If they are currently running, close the sfHub and sfStudio or the end user application.
2. Open the applicable **sfAccess.ini** file (for either sfStudio or the end user application).
3. Under the Processing section, find the variables for boresight and tip offset.

```
[Processing]
PredictionS = 0.020
VirtualSyncCtrl = 1
BoresightRefRollRad = 0
BoresightRefPitchRad = 0
BoresightRefYawRad = 0
TipOffsetXM = 0
TipOffsetYM = 0
TipOffsetZM = 0
```

4. Change the values to match the desired offset in radians for boresight and meters for tip offset. Following the examples above, the changes below reflect a 40° pitch and 5cm +Y offset.

```
[Processing]
PredictionS = 0.020
VirtualSyncCtrl = 1
BoresightRefRollRad = 0
BoresightRefPitchRad = 0.698
BoresightRefYawRad = 0
TipOffsetXM = 0
TipOffsetYM = 0.05
TipOffsetZM = 0
```

5. Once the changes have been made, save the file.

5.4.2 Changing TCP and UDP Settings in sfAccess.ini and sfHub.ini

Configuring the communication ports requires changes to both sfAccess.ini and sfHub.ini. Under normal circumstances, these settings do not need to be changed. However, there are two scenarios that may require a change. The first is if there a port conflict, in which case the ports used by sfAccess and sfHub may need to be changed. The second is in a server setup, where data needs to be received or viewed by a different computer (the master) than the one running sfHub to perform the tracking computations (the host, or slave).

A. To change the TCP and UDP ports, follow the instructions below.

1. If they are currently running, close the sfHub and sfStudio or the end user application.
2. Open the applicable **sfAccess.ini** file (for either sfStudio or the end user application).
3. Under the Processing section, find the variables for each of the ports.

```
[Communication]
TcpSfRxEnabled = true
TcpSfRxAddr = localhost
TcpSfRxPort = 51717
UdpSfRxPort = 9001
UdpSfRxImgPort = 9003
UdpSfCorePort = 9002
UdpSfRxInputPort = 9004
UdpSfHubPort = 9005
UdpNftPort = 9007
```

4. Change the values as needed for the system. In this example, it will be assumed there is a conflict on all or most of the default UDP ports. However, ports 51718, 8001-8005 and 8007 are unused by the system and make for ideal replacements.

```
[Communication]
TcpSfRxEnabled = true
TcpSfRxAddr = localhost
TcpSfRxPort = 51718
UdpSfRxPort = 8001
UdpSfRxImgPort = 8003
UdpSfCorePort = 8002
UdpSfRxInputPort = 8004
UdpSfHubPort = 8005
UdpNftPort = 8007
```

5. Once the changes have been made, save the file.
6. The variable names for each port in sfHub.info differ from those in sfAccess.ini. The table below shows the sfAccess.ini variable names that correspond to each sfHub.info variable and their default ports (as seen in Step 3). The **UdpSfHubPort** is not used by sfHub. If only this port was changed in Step 4, the remaining steps can be skipped. Otherwise, sfHub.info needs to be edited to reflect the changes made in the sfAccess.ini.

sfAccess.ini Variable	sfHub.info Variable	Default Port
TcpSfRxPort	tcpPort	51717
UdpSfRxPort	udpMeta	9001
UdpSfRxImgPort	udpImg	9003
UdpSfCorePort	udpCore	9002
UdpSfRxInputPort	port	9004
UdpSfHubPort		9005
UdpNftPort	udpNft	9007

7. Open the **sfHub.info** file from the **sfHub** subfolder of the IS-1500 software directory.

8. Under the Master subsection of the General section, find the **udpMeta**, **udpCore**, **udpNFT**, and **udpImg** variables.

```
; TCP port for communication with sfAccess.
tcpPort 51717

udpMeta localhost:9001  ; Meta packet broadcast address
udpCore localhost:9002  ; Core packet broadcast address
udpNft  localhost:9007  ; NFT broadcast address (IS-1500 only)
udpImg  localhost:9003  ; Image packet broadcast address (IS-1200 only)
```

9. Change the port values as so the ports for **tcpPort**, **udpMeta**, **udpCore**, **udpNFT**, and **udpImg** match the ports for **TcpSfRxPort**, **UdpSfRxPort**, **UdpSfCorePort**, **UdpNftPort**, and **UdpSfRxImgPort** entered in Step 4.

```
; TCP port for communication with sfAccess.
tcpPort 51718

udpMeta localhost:8001  ; Meta packet broadcast address
udpCore localhost:8002  ; Core packet broadcast address
udpNft  localhost:8007  ; NFT broadcast address (IS-1500 only)
udpImg  localhost:8003  ; Image packet broadcast address (IS-1200 only)
```

10. Under the Gps subsection of the IS1500 section, find the **port** variable.

```
Gps
{
    enable   0    ; 1/0 - on/off
    port     9004 ; UDP input port
    accuracy 3    ; GPS antenna accuracy (meters)
}
```

11. Change the **port** values to match the port for **UdpSfRxInputPort** entered in Step 4.

```
Gps
{
    enable   0    ; 1/0 - on/off
    port     8004 ; UDP input port
    accuracy 3    ; GPS antenna accuracy (meters)
}
```

12. Once the changes have been made, save the file.

B. To allow the system to run sfHub on a host computer and sfStudio or the end user application on a master computer, use the following steps. NOTE: Image transfer to sfStudio or the user application is not supported in this configuration.

1. If they are currently running, close the sfHub and sfStudio or the end user application.
2. On the master computer, open the applicable **sfAccess.ini** file (for either sfStudio or the end user application).
3. Under the Processing section, find the TcpSfRxAddr variable. By default, the value is localhost (IP address 127.0.0.1).

```
[Communication]
TcpSfRxEnabled = true
TcpSfRxAddr = localhost
TcpSfRxPort = 51717
UdpSfRxPort = 9001
UdpSfRxImgPort = 9003
UdpSfCorePort = 9002
UdpSfRxInputPort = 9004
UdpSfHubPort = 9005
UdpNftPort = 9007
```

4. Change the value to the IP address of the host computer. In the example, the computer running sfHub has the IP address 192.168.1.101.

```
[Communication]
TcpSfRxEnabled = true
TcpSfRxAddr = 192.168.1.101
TcpSfRxPort = 51717
```

5. Once the changes have been made, save the file.
6. On the host computer, open the **sfHub.info** file from the **sfHub** subfolder of the IS-1500 software directory.
7. Under the Master subsection of the General section, find the **udpMeta**, **udpCore**, **udpNFT**, and **udpImg** variables.

```
; TCP port for communication with sfAccess.
tcpPort 51717

udpMeta localhost:9001   ; Meta packet broadcast address
udpCore localhost:9002   ; Core packet broadcast address
udpNft  localhost:9007   ; NFT broadcast address (IS-1500 only)
udpImg  localhost:9003   ; Image packet broadcast address (IS-1200 only)
```

8. Change the address values for **tcpPort**, **udpMeta**, **udpCore**, **udpNFT**, and **udpImg** to match the IP address of the master computer. In the example, the computer running sfStudio or the end user application is 192.168.1.100

```
; TCP port for communication with sfAccess.
tcpPort 51717

udpMeta 192.168.1.100:9001   ; Meta packet broadcast address
udpCore 192.168.1.100:9002   ; Core packet broadcast address
udpNft  192.168.1.100:9007   ; NFT broadcast address (IS-1500 only)
udpImg  192.168.1.100:9003   ; Image packet broadcast address (IS-1200 only)
```

9. Once the changes have been made, save the file.

5.5 sfStudio.ini

The **sfStudio.ini** file is located in the sfStudio subfolder of the software directory. The file divided into three short sections.

5.5.1 sfStudio Connect Button Settings for sfHub AutoLaunch

There are two important settings in the sfHub section of sfStudio.ini. The first sets the file directory path of the sfHub program. This is useful if software directory structure has been changed. The other defines whether sfHub will be automatically launched when the **Connect** button is pressed in sfStudio. Those using sfStudio with multiple InterSense product lines with other communication software (such as sfRxCore for a HObIT) may wish to disable this setting. When AutoLaunch is disabled, the sfHub software will need to be run manually for sfStudio to connect to the InertiaCam. Use the following directions to change either of these settings.

1. If they are currently running, close the sfHub and sfStudio programs.
2. Open **sfStudio.ini**.
3. Under the sfHub section, find the **Path** and **AutoLaunch** variables.

```
[sfHub]
Path=../sfHub
AutoLaunch=1
```

4. To specify the sfHub path, change the **Path** value. In the example below, sfHub is located in the C:\InterSense\IS-1500 Software Package\Windows-Software\sfHub folder on a Windows computer.

```
[sfHub]
Path= C:\InterSense\IS-1500 Software Package\Windows-Software\sfHub
AutoLaunch=1
```

5. To enable or disable sfHub AutoLaunch, set the value to 1 or 0 respectively. In the example, AutoLaunch has been disabled.

```
[sfHub]
Path=../sfHub
AutoLaunch=0
```

6. When the settings have been changed, save the file.

5.5.2 sfStudio Light and Dark Themes

By default, sfStudio is configured to use a light colored theme. However, a dark version is available, as seen in **Figure 38**. Use the following directions to change the sfStudio theme.

1. If they are currently running, close the sfHub and sfStudio programs.
2. Open **sfStudio.ini**.
3. Under the Theme section, find the **Style** variable.

```
[Theme]
Style = Light
```

4. To enable the light or dark theme, enter **Light** or **Dark**, respectively

```
[Theme]
Style = Dark
```

5. When the settings have been changed, save the file.

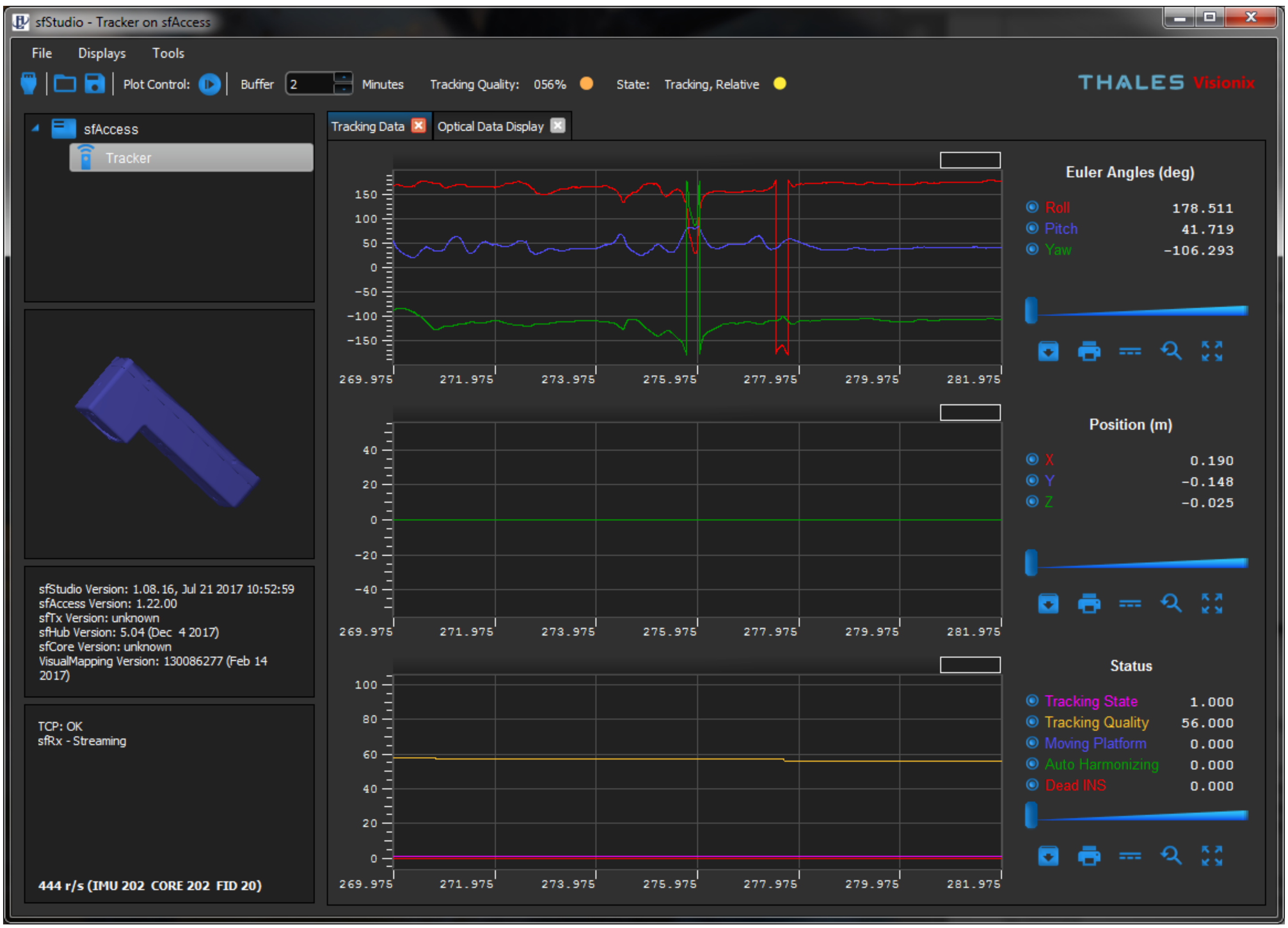


Figure 38 – sfStudio Dark Theme

6 Using HObIT Sensors with the IS-1500 Software Suite

The IS-1500 Software Suite is now backwards compatible for use with HObIT sensors, with limited functionality.

6.1 Differences and Limitations

6.1.1 Natural Feature Tracking

The IS-1500 Software suite is backwards compatible with the HObIT, but it is not forwards compatible. HObIT sensors are not able to perform NFT due to hardware incompatibility, but it can be used for fiducial tracking as originally designed with the updated software suite. This means that HObITs are limited to using sfCore 3-DOF and sfCore 6-DOF modes with sfHub. If sfHub.ini is configured to use any tracking modes that include the use of NFT, sfHub instead will automatically default to 6-DOF sfCore mode when a HObIT sensor is connected.

It is important to note that sfHub is a functional replacement for sfRxCore. Traditionally, the HObIT system uses the sfTx program on-board the sensor to process images for fiducial recognition, and fiducial data is passed to sfRxCore for tracking on the host computer. Sensor parameters such as exposure settings are configured in sfTx.ini while connection and tracking parameters are configured in sfRx.ini. By contrast, in the IS-1500 system, the InertiaCam does not have an on-board processor. Instead, raw image and IMU data is passed directly to sfHub, which performs both image processing and tracking on the host computer. When using the InertiaCam, sensor, communication, and tracking parameters are all configured in sfHub.info.

When sfHub is used to connect to a HObIT, it acts more closely like sfRxCore. The on-board processing performed by the HObIT is passed to sfHub, and sfHub only performs the tracking algorithms on the host-computer. This also means that HObIT tracking parameters are still configured on the sensor using sfTx.ini. sfHub.info does not apply any settings in the InertiaCam section when used with a HObIT, including exposure settings.

6.1.2 isUnitySample

The provided Unity sample is compatible with the HObIT, but because the HObIT is not capable of 6-DOF tracking without optical lock on fiducials, the quality and behavior of the tracking data may not function as desired.

6.1.3 sfStudio Features

Several sfStudio features, especially those specific to Natural Feature Tracking, are unavailable when using the HObIT. This means some displays may have inactive plots and/or buttons. For example, the following are features that are unavailable when using a HObIT:

- Optical Rate Plot in Additional Data
- VINS Image in Optical Data Display
- GPS Data Display

However, there are also a few features that can only be used with a HObIT, such as:

- Harmonization Data Display
- Night Mode in Optical Data Display (Only available on HObITs with Day and Night modes)
- Data Logger Tool

6.2 Configuring sfHub to Connect to a HObIT

By default, sfHub is configured to connect to an InertiaCam sensor. Some settings in sfHub.info need to be changed in order to connect to a HObIT sensor.

1. If they are currently running, close the sfHub and sfStudio programs.
2. Open **sfHub.info**.
3. Under the Master subsection of General settings, find the **sensorType** variable.

```
General
{
    appMode     1   ; (1) Tracking (2) Playback (3) Configuration Utility
    numSensors  1   ; (1) Min (4) Max
    verbosity   0   ; (0) Low (1) Med (2) High

    Master
    {
        ; (1) sfCore 3DOF
        ; (2) sfCore 6DOF
        ; (3) NFT + Fid
        ; (4) NFT + GPS
        ; (5) NFT + GPS + Fid
        trkMode 3

        sensorType 2 ; (1) HObIT (2) InertiaCam
        initState  1 ; (1) Streaming, (0) Paused
        logData    1 ; (1) Enable data logging
    }
}
```

4. To change sfHub to connect to a HObIT, set **sensorType 1**.

```
Master
{
    ; (1) sfCore 3DOF
    ; (2) sfCore 6DOF
    ; (3) NFT + Fid
    ; (4) NFT + GPS
    ; (5) NFT + GPS + Fid
    trkMode 3

    sensorType 1 ; (1) HObIT (2) InertiaCam
    initState  1 ; (1) Streaming, (0) Paused
    logData    1 ; (1) Enable data logging
}
```

5. If the **trkMode** parameter is set to any of the modes with NFT, when a HObIT is connected, the tracking mode will automatically switch to sfCore 6-DOF and this step can be skipped. However, to specify that 3-DOF sfCore mode should be used instead, set **trkMode 1**.

```
Master
{
    ; (1) sfCore 3DOF
    ; (2) sfCore 6DOF
    ; (3) NFT + Fid
    ; (4) NFT + GPS
    ; (5) NFT + GPS + Fid
    trkMode 1
```

6. When the settings have been changed, save the file.

7 IR INERTIACAM

The IR InertiaCam is a type of InertiaCam sensor that uses four 850 nm wavelength LEDs to illuminate the scene. The IR InertiaCam sensor is able to capture both visible light and the IR wavelength. Like the standard InertiaCam, the resulting image is in gray scale. Using both visible light and IR allows the IR InertiaCam to track in any conditions the standard InertiaCam can, but has the added ability to track in darkly lit environments and using retroreflective IR fiducials.

Figure 39 – IR InertiaCam

The IR InertiaCam operates largely the same way as the standard InertiaCam. The IR LEDs are turned on when sfHub connects to the IR InertiaCam and are turned off again when sfHub is closed. The blue status light is turned on when the IR illumination is active. For eye safety reasons, users are advised against looking directly at the IR LEDs while this blue light is on.

7.1 Data

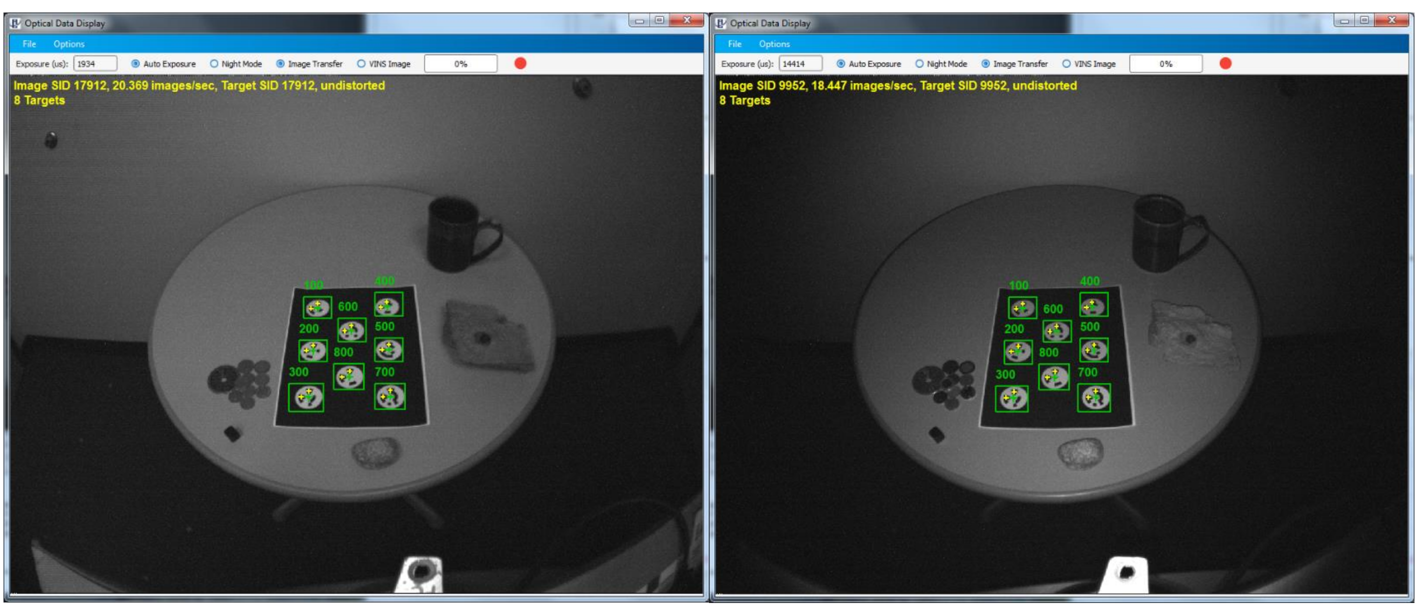


Figure 40 – IR InertiaCam Image in a Brightly Lit Room (Left) vs an Unlit Room (Right)

The IR illuminator provides sufficient light to see paper fiducials 4 feet away in pitch black visible light conditions with the camera set to auto exposure. It enables recognition of a 12.7 cm diameter fiducial from a distance of 4 feet with an auto exposure value of 14.2ms. At closer distances, the illumination can be too bright and tends to over expose smaller IR fiducials. To combat this, the IR InertiaCam needs to be set to manual exposure of around 150µs when used with smaller IR fiducials. The IR InertiaCam can read 1.2cm diameter retro reflective fiducials from a distance of 24cm with no ambient light. The illuminator also enables NFT at distances of 12-15ft with an auto exposure value of 14.2ms. Due to the high exposure value, the sensor should not be moved rapidly in darker environments as it may cause the captured images to be blurred, decreasing the accuracy of feature detection. The table below summarizes the performance of the IR InertiaCam. For more information on how to change the camera exposure, refer to the InertiaCam Exposure Settings section.

Fiducial Type	Fiducial Diameter (mm)	Distance to Target (cm)	Camera Exposure (µs)
Paper	127	1219.2	Auto (14200)
Retroreflective	12	24	Manual (150)
NFT	-	4572	Auto (14200)

7.2 Configuration Settings for IR InertiaCams

7.2.1 Enabling and Disabling the IR Illuminator

A setting in sfHub.info, enabled by default, can be used to enable or disable use of the IR LEDs. Enabling or disabling this setting with a standard InertiaCam will not have any effect. As shown below, in the InertiaCam section of sfHub.info, set **illuminator 1** to enable the IR illuminator and set **illuminator 0** to disable it.

```
InertiaCam
{
 Master
    {
        ncDataDiv        5      ; NavChip data rate divisor (rate = 1000/divisor in Hz)
        frameInterval    50000  ; Frame interval in microseconds.
        manualExposure   2500   ; Initial manual exposure
        autoExposure     1      ; (1) Enable (0) Disable
        autoExposureMin  100    ; Microseconds.
        autoExposureMax  15000  ; Microseconds.
        autoExpsureLum   1000   ; Auto exposure target luminosity
        illuminator      1      ; IR illuminator enable
    }
}
```

7.2.2 IR Retroreflective Fiducial Recognition

It should be noted that retroreflective fiducials are color-inverted from typical visible light fiducials. With visible light fiducials, the center eye and patterns are usually white set against a black circle. Retroreflective fiducials use a black center eye and pattern against a white background. By default, the IS-1500 is configured to read fiducials with white center eyes. In order to use retroreflective fiducials or visible light fiducials with black center eyes a setting in sfHub.info must be changed. As shown below, in the ImgProc section of sfHub.info, setting **whiteCenterEyes 1** configures the system to read fiducials with white center eyes, while setting **whiteCenterEyes 0** configures the system to read fiducials with black center eyes.

```
ImgProc
{
    Master
    {
        whiteCenterEyes 0   ; (1) white center (0) black center
        filterFidMod    100 ; Fiducial modulus
        imProcDiv       1   ; Image processing divisor
        undistort       1   ; (1) Enable undistortion
    }
}
```

8 TROUBLESHOOTING

8.1 Frequently Asked Questions

Do I need to map or scan the environment before using Natural Feature Tracking?

In short, no. Natural feature tracking using VINS with the IS-1500 is done 'on the fly' by finding whatever natural features happen to be in the camera's field of view in each frame. However, if a custom fiducial constellation is being used to stabilize tracking and provide fixed global alignment, it will need to be mapped prior to use. (The fiducial poster provided in the IS-1500 Hardware Kit is the default constellation and does not need to be mapped prior to use.) For more information on how IS-1500 tracking works, see the Understanding Tracking section.

Does the tracking work outdoors or in sunlight?

Yes. Since the InertiaCam does not use a depth sensor but instead relies solely on a visible light camera, the system works just as well indoors as outdoors. However, if the system is using manual exposure, a change to the defined exposure or setting the system to automatic exposure may be required. For more information on how to change exposure settings, see the InertiaCam Exposure Settings section. It should also be noted that the InertiaCam is not waterproof and should not be used in rain, high humidity, or other such conditions.

Can I track at night or in dark environments?

Some InertiaCam sensors can, but not all. The IR InertiaCam variant is able to use an IR illuminator that allows the tracking to continue working in low light environments.

If I have a HObIT or other InterSense optical-inertial tracker, can I update the firmware or software to use it to track natural features?

Unfortunately, no. While the same software suite is used for both HObIT and the InertiaCam, the HObIT and other legacy optical inertial trackers are not compatible with Natural Feature Tracking on a hardware level. The IS-1500 Natural Feature Tracker uses the InertiaCam hardware, which has been designed specifically for natural feature tracking in mind. Changes from other InterSense trackers include the type of lens used for the camera, lack of on board processing, and different communication protocols.

Can I get two InertiaCam trackers to run on the same system?

Connecting to multiple InertiaCam trackers on one host computer is not currently supported by the software suite. Separate host computers must be used to run each tracker.

How do I save the tracking data I've collected?

There are several ways to go about saving tracking data. Most of these methods are described in the sfStudio User Guide. Another method uses sfHub to collect the raw sensor data (both the camera images and IMU data) as it was originally received. These log files are typically quite large, but very useful for diagnostics and allow for replay. For more information on this method, see the Logging and Playback section.

Can I use the system to save the natural feature data and create a 3D map of the environment?

Currently, no. The algorithms used to create a 3D map using natural features are currently in development. However, when this feature is released, it will be provided as a software update and will not require any hardware changes.

Why does the image resolution seem so low?

Actually, the system is capable of sending higher quality images at 1280x960 resolution. However, after extensive testing it was found that using the higher resolution images did not noticeably increase tracking performance in most cases. Instead, to lighten the computational load on the host computer and allow for more reasonably sized log files, InertiaCams are now configured to send 640x480 resolution images by default. If you would like to change this setting, simply follow the instructions provided in the Image Resolution in the Configuration Utility section.

Can I use any USB-C cable to connect to the InertiaCam?

Theoretically, yes. However, because of the varying quality of USB cables, Thales Visionix is only able to support and recommend use of the USB-C cables packaged in the IS-1500 Hardware Kit. If a replacement USB-C cable is required, contact Thales Visionix Technical Support.

The fiducial poster included with the hardware kit has been lost or damaged. Do I need to order a new one?
No. A new fiducial poster of the default IS-1500 constellation can be printed on any white paper or surface using the **Fiducial Poster – 8.5 x 11.pdf** file in the **Documentation** folder of the IS-1500 software directory. When reprinting the poster, it is very important to use high quality settings and to not change original size. When printing has completed, double check the latter by measuring the fiducials. They should have a diameter of 41.5mm.

My tracking seems to be very erratic. I keep drifting, jittering, or giving very inaccurate data.
There are a number of ways tracking quality can be compromised. The Understanding Tracking section may provide a number of useful tips for use. The following are common problems to look out for when tracking with the IS-1500:

- There is dim lighting in the tracking area
- The camera on the InertiaCam is repeatedly or continually obstructed
- There are fingerprints or smudges on the InertiaCam lens
- The tracking area or InertiaCam field of view is largely empty without any defining features or objects
- The only objects in the InertiaCam's field of view are moving or are at very long distances away
- Tracking is performed over long distances (greater than 200 meters) without periodically achieving optical lock on fiducials
- The physical fiducial constellation does not match the map file (likely if the problem only exists when fiducials are in the field of view)
- Another IS-1500 system is being used on the same network, and is broadcasting on the same UDP ports

The majority of these problems can be investigated by watching the Optical Data display in sfStudio. The VINS Image overlay in particular will show what natural features are found and used by the system.

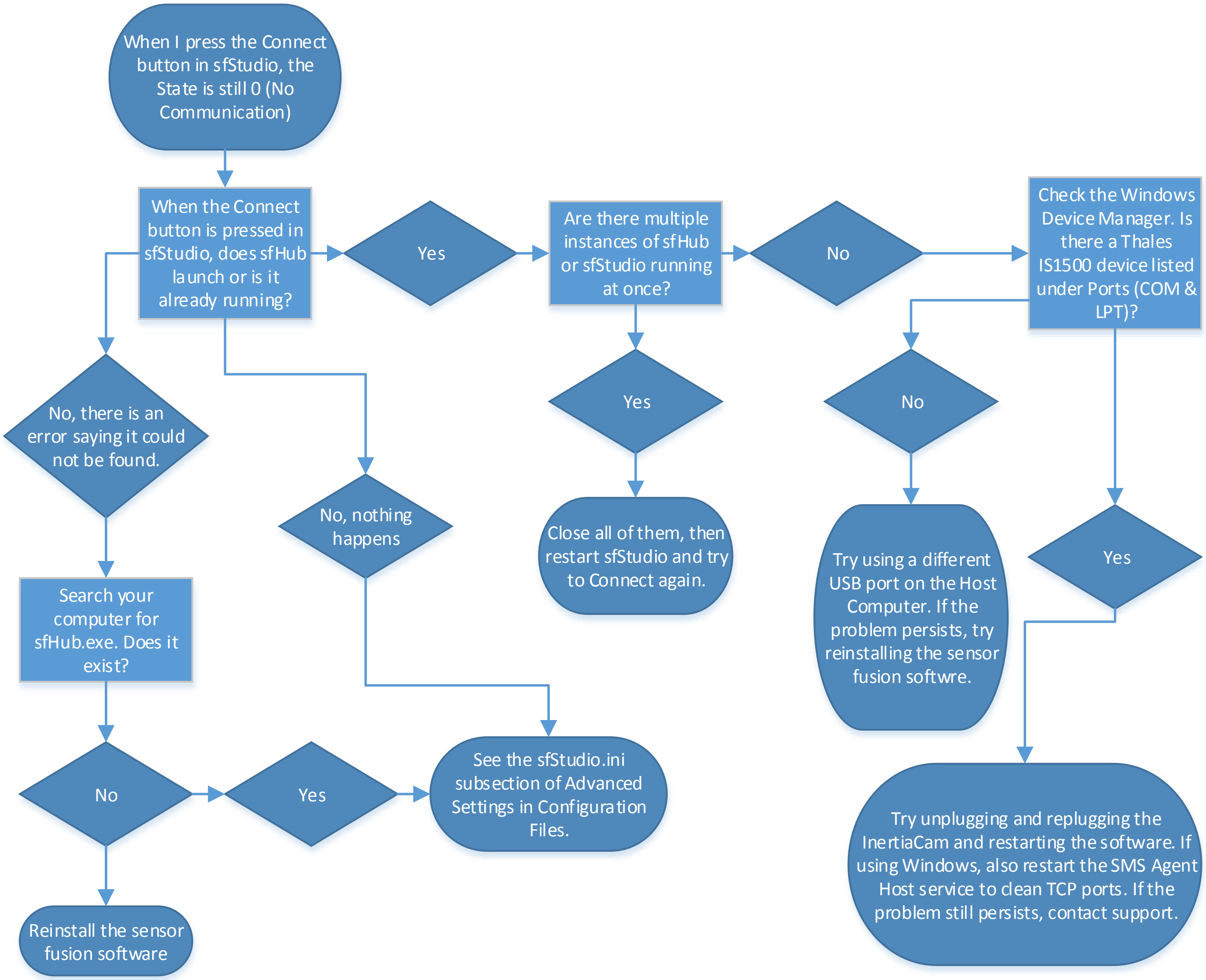
I believe my InertiaCam or cable might have a hardware problem. How do I confirm that or make a return?
Thales Visionix Technical Support can assist with determining if a hardware problem is likely. To return a piece of hardware, use the Request RMA (Return Material Authorization) form on the InterSense.com website:

http://www.intersense.com/pages/31/84

An RMA number will be issued by Thales Visionix Technical Support along with further instructions. All InterSense hardware comes with a one-year manufacturer's warranty. However, the warranty can be voided if the user accidentally damages or tampers with the hardware. InertiaCam trackers are individually calibrated using high precision proprietary techniques and will become decalibrated if tampered with. **User repair attempts are not advised.**

I still have a 0, No Communication Tracking State after using the sfStudio Connect button.

This likely indicates that sfStudio is not able to establish a connection to the InertiaCam via sfHub. Assuming that the InertiaCam is correctly plugged into the host computer, there are a number of possible causes for this. Try running through the flowchart below. If at any point further assistance is required, feel free to contact Thales Visionix Technical Support via the information provided on Page 2. It may assist with feedback to indicate flowchart results.



9 Glossary

9.1 Acronyms

NFT – Natural Feature Tracking
6-DOF – Six Degrees of Freedom
3-DOF – Three Degrees of Freedom
IMU – Inertial Measurement Unit
AHRS – Attitude and Heading Reference System
VINS – Vision-aided Inertial Navigation System
EKF – Extended Kalman Filter
PRA – Pose Recovery Algorithm
SVM – Simple Visual Mapper

9.2 Terms

3-DOF – Measuring three degrees of freedom; yaw, pitch, and roll orientation
6-DOF – Measuring six degrees of freedom; yaw, pitch, and roll orientation as well as X, Y, and Z position
InertiaCam - An InterSense optical-inertial 6-DOF motion tracker
NavChip – An InterSense 3-DOF IMU chip
sfHub – The program that receives sensory input from the InertiaCam, analyses it, and outputs tracking data to sfAccess
sfAccess – A low-level tracking data interface
InterSense library – The recommended method of interfacing with tracking data
IS-1500 Unity Plugin – A plug-in for interfacing IS-1500 tracking with Unity applications
sfStudio – The primary program for IS-1500 device setup, display of tracking information, and general diagnostics
tracking area/environment – The area or environment the IS-1500 is being used in
world reference frame – The position and orientation used for reference of global alignment
origin point - Defines the location in the tracking environment at which (X, Y, Z) is (0, 0, 0)
tracker/body reference frame – A predetermined point on and orientation of the InertiaCam used to define the tracker's position and orientation relative to the world reference frame
roll – Rotation about the X axis
pitch – Rotation about the Y axis
yaw – Rotation about the Z axis
natural features – pixel patches identified as stationary objects in the environment and used for tracking by the VINS algorithm
VINS – The algorithm used to identify natural features and use them to determine the InertiaCam's position and orientation
robocentric – A method of describing objects as relative to the point they are observed from
parallax – The visual concept that an object appears to change orientation/position as the point of observation is moved about it
EKF – A mathematical method of nonlinear state estimation using the current mean and covariance
IMU exercising – The manipulation of an IMU involving alternately periodically rotating it and leaving it still, allowing the EKF to initialize with accurate data when tracking begins
divergence – A tracking state that allows reported pose to drift off rapidly, typically induced when there is insufficient optical data the system is forced to rely on IMU data
fiducials – Circular black-and-white patterns that act as known physical landmarks for the system
constellation – The layout of all the fiducials in a system
fiducial ID – The numerical value of a fiducial pattern as decoded by the system that acts as an identifier
PRA – The algorithm used to identify fiducials and use them to determine the InertiaCam's position and orientation relative to the world reference frame defined by the constellation
environmentPSEs.cfg – The constellation map file used by PRA while tracking
constellation map file – A file containing a list of the IDs of the fiducials in the constellation and their known positions and orientations relative to a world reference frame
embedded settings – An alternative tracking configuration designed to lower the computational requirements
Fiducial Poster – 8.5 x 11.pdf – The PDF file of the system's default fiducial constellation
Tracking State – The sfStudio indicator of tracking status; State 0 indicates there is no connection to the tracker, State 1 indicates the system is tracking without a global reference frame, State 2 indicates optical lock on a fiducial constellation was acquired and tracking is relative to a globally defined world reference frame, and State 3 indicates the tracker is lost
SVM – A tool in sfStudio used to generate a map file of a fiducial constellation
sfCore Mode – A setting that, when enabled, allows for use of SVM but disables VINS and PRA