
ibeo
automotive

ibeo automotive

User Manual

Ibeo Scala Feature Fusion

PASSION FOR SCANNING

Ibeo Automotive Systems GmbH
Merkurring 60-62
D - 22143 Hamburg
Phone: +49 - (0) 40 298 676 - 0
Fax: +49 - (0) 40 298 676 - 10
E-mail: info@ibeo-as.com Copyright

Version History

Date	Version	Changes
08-Jul-2015	1.0	Initial Version

Table 0.1: Version History

ibeo automotive

Contents

1 About this document **6**

1.1 Principle of the document . . . 6
1.2 Related documents . . . 6
1.3 Target group . . . 6
1.4 Symbols and notes . . . 7
1.5 Terms and definitions . . . 7
1.5.1 Operator . . . 7
1.5.2 Trained Stuff . . . 8
1.5.3 Administrator . . . 8
1.6 Manufacturer . . . 8
1.7 Warranty . . . 8

2 Safety **10**

2.1 Intended Use . . . 10
2.2 Improper use . . . 10
2.3 General notes on safety . . . 10
2.3.1 Operation . . . 10
2.3.2 Disposal . . . 11

3 System Architecture **12**

3.1 Hardware components . . . 13

4 Ibeo ECU **20**

4.1 Web Interface . . . 22
4.2 System Files . . . 23
4.3 Packages Page . . . 24
4.4 Software Package Configuration System.xml . . . 26
4.5 Software Package Configuration CANParser.xml . . . 32

5 Software Architecture **36**

5.1 Required Input Signals . . . 36
5.2 Processing Structure . . . 36
5.3 Scan Fusion . . . 38
5.4 Ethernet Output Data . . . 38
5.5 CAN Output Data . . . 39

1 About this document

Please read this chapter carefully before working with the document and the Ibeo Feature Fusion System.

1.1 Principle of the document

This operating manual (OM) contains all information required for configuration and operation of the Ibeo Feature Fusion System. The operating manual is part of the technical documentation of the Ibeo Feature Fusion System.
This operating manual does not include instructions for operation of the vehicle into which the Ibeo Feature Fusion System is integrated. For information about that, refer to the operating manual of the vehicle.
This operating manual does not include instructions for operation of the Ibeo Lux sensors, which are part of the Ibeo Feature Fusion System. For information about that, refer to the operating manual of the Ibeo Lux sensor, which is provided together with this operating manual.
This operating manual shall help you to avoid improper use. Strict adherence to the instructions given in this operating manual ensures the proper function of the Ibeo Feature Fusion System.

1.2 Related documents

The following documents are referred to in this document.

- Ibeo Scala User Manual
- Ibeo Scala ethernet protocol description
- Ibeo CAN protocol description

1.3 Target group

This operating manual is written for trained and qualified staff that will integrate, configure and operate the Ibeo Feature Fusion System in the operating environment, e.g. on a test vehicle.
The following conventions apply in this operating manual:

- The text uses abbreviations. In each chapter, abbreviations appearing the first time are explained. The abbreviation is written in parenthesis behind the term. Example: Operating Manual (OM)
- The pages of each chapter are numbered subsequently.
- Tables and figures are numbered consecutively within each chapter.
- Links in the text lead you to supplementary or more detailed information.
- In chapter you find general safety notes about possible dangers when operating the Ibeo Feature Fusion System. Additionally, specific safety notes are printed directly before instructions which, if not followed properly, could result in danger for persons or the equipment.
- This OM is subject to changes. If the Ibeo Feature Fusion System is changed as a result of technical advancement, you must include the additional or changed pages at the corresponding place.

In chapter 2 "Safety", page 10, you find general safety notes about possible dangers when operating the Feature Fusion. Additionally, specific safety notes are printed directly before instructions which, if not followed properly, could result in danger for persons or the vehicle.

This OM is subject to changes. If the Feature Fusion is changed as a result of technical advancement, you must include the additional or changed pages at the corresponding place.

1.4 Symbols and notes

This operating manual uses different symbols for notes on safety and information.
The notes on safety include the source of the danger, possible or probable results as well as means to stop the hazard.

DANGER

This symbol indicates an immediate danger which could lead to severe bodily harm or to death.

WARNING

This symbol indicates a pssible danger which could lead to severe bodily harm.

CAUTION

This symbol indicates a possible danger which could lead to bodily harm.

CAUTION

This symbol indicates a possible danger which could lead to property damage.

NOTE

This symbol indicates special information about key functions or special usage tips which shall help you to use all functions optimally.

1.5 Terms and definitions

1.5.1 Operator

The operator uses the Ibeo Feature Fusion System as owner or hirer. He is responsible for

- proper and safe operation of the Ibeo Feature Fusion System
- intended use of the Ibeo Feature Fusion System
- the appointment of suitable trained staff to mount, install, configure, adjust and clean the Ibeo Feature Fusion System.

1.5.2 Trained Stuff

Trained staff has undergone a specific training and is thus able to perform the assigned tasks and to detect possible dangers. The trained staff must be qualified for handling

- mechanic and electric assemblies
- control and feedback control systems.

1.5.3 Administrator

An administrator is a specifically trained employee of the customer or an accordingly qualified employee of Ibeo. An administrator can access all software functions. The administrator may

- open, delete or lock user accounts
- change the configuration of the Feature Fusion System
- reset the devices connected to the Feature Fusion System
- launch or shut down the software of the Feature Fusion System
- update new software for the Feature Fusion System

An administrator has the highest rank of permissions in the user administration.

1.6 Manufacturer

Ibeo Automotive Systems GmbH
Merkurring 60-62
D - 22143 Hamburg
Phone: +49 - (0) 40 298 676 - 0
Fax: +49 - (0) 40 298 676 - 10
E-mail: info@ibeo-as.com

1.7 Warranty

The Ibeo Feature Fusion System has not yet been constructed according to all required legal requirements and safety-related rules. It is not yet state-of-the-art nor does it fulfill the requirements of the EC conformity.

DANGER

The Ibeo Feature Fusion System including its software and the OM are still in the development and prototype state. Therefore, its usage is limited to test purposes only.

In case of using the Ibeo Feature Fusion System and its connected components and software for other purposes than the intended use, Ibeo will not assume any responsibility nor be liable to third parties; this applies to direct, indirect or exceeding damages, accidental damages or consequential damages.
These include
The Ibeo Feature Fusion System is:

- not used as intended
- not used according to the instructions in this OM
- modified in terms of construction or functions without written consent by ibeo
- equipped with spare parts not delivered or approved in writing by ibeo
- repaired improperly
- repaired by trained staff not authorized in writing by ibeo
- damaged by an "act of God"

Ibeo does not assume any liability for external devices connected to the system which cause faults and thus damages.

2 Safety

2.1 Intended Use

The Ibeo Feature Fusion System serves for detecting and identifying objects around a vehicle under a specific angle. It is integrated into the vehicle.
The area for intended use is the area of the vehicle.

NOTE
Any other use is only permitted after consulting Ibeo.

2.2 Improper use

Every use other those listed above is considered improper. Ibeo is not liable for damages to persons and property resulting from such improper use.

2.3 General notes on safety

The Ibeo Feature Fusion System can cause danger for persons and property if handled or used improperly. Therefore the operator must ensure that every person working with the Ibeo Feature Fusion System has read and understood this OM.
For installation and usage of the Ibeo Feature Fusion System as well as for commissioning and regular technical inspection, national/international legal requirements apply.
The operator of a vehicle equipped with a Ibeo Feature Fusion System is responsible for consulting the responsible authorities about applicable safety rules and regulations, and adhere to them.
The notes, especially the inspection notes of this operating manual (e.g. usage, mounting, installation or integration into the vehicle control system) must be observed.
Adhere to the following safety notes in order to prevent dangers for persons and/or property:

- The operator must ensure by suitable instructions and inspections that the Ibeo sensors, which are part of the Ibeo Feature Fusion System, are always clean.
- Additionally, the local safety and accident prevention regulations apply for operating the Ibeo Feature Fusion System.
- A defect of the control functions can cause danger for human life or property damage at the Ibeo Feature Fusion System.

2.3.1 Operation

The operator must

- provide a permanently perfect operating state of the Ibeo Feature Fusion System
- take measures for antistatic protection
- make sure that only trained staff modifies the Ibeo Feature Fusion System

The trained staff must report any relevant modifications in the functional sequence of the Ibeo Feature Fusion System immediately to the operator.

2.3.2 Disposal

The Ibeo Feature Fusion System is designed to burden the environment as little as possible and to consume a minimum of energy and resources.

3 System Architecture

This section describes the system architecture of the Ibeo Feature Fusion system.

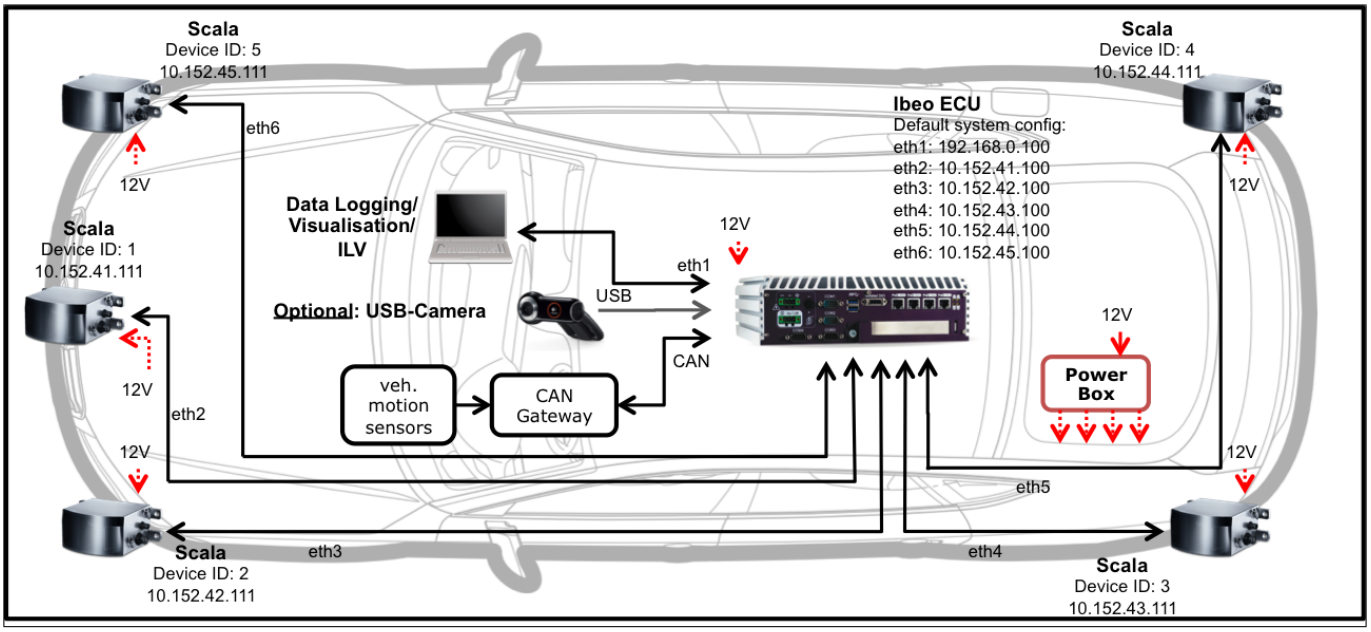

Figure 3.1: Car Architecture

Figure 3.1 shows an example setup with a five Scala sensor fusion system, mounted on a vehicle.
Up to five laserscanners are connected to the central computation unit (Ibeo ECU, Ethernet port 2-6) via ethernet. Each Scala sensor requires the usage of one Valeo Media Converter Box. It converts the direct sensor output into standard ethernet. The laserscanners provide their measurements to the Ibeo ECU which performs the fusion of measured features, the object detection and tracking algorithms.
The Ibeo ECU receives vehicle motion data, including the vehicle speed and yaw rate, via CAN. This vehicle motion data is preferably provided by the vehicle internal communication network, which is interfaced to the Ibeo ECU via a gateway.
A USB camera can be connected to the Ibeo ECU. The image from the camera can be merged with the raw data and object data stream as a feedback for the operator.
There are two interfaces on which the Ibeo ECU provides processed data output.

- Ethernet port 1 can be used for data logging and visualization. This interface provides
  - fused raw scan data of all scanners in the fusion system
  - object data (up to 254)
  - vehicle motion data
  - log messages and
  - optional video image from the USB camera

- Alternatively the CAN bus can be used as a system output for Object data (up to 10) only. Please note, that the maximum data rate of the CAN bus is much lower than the data rate of the Ethernet port. Therefore, only a reduced number of objects can be provided via CAN. To ensure that important objects are provided first, there are different object prioritization criteria to choose from.

3.1 Hardware components

This section describes the hardware components in more detail.

Ibeo Scala

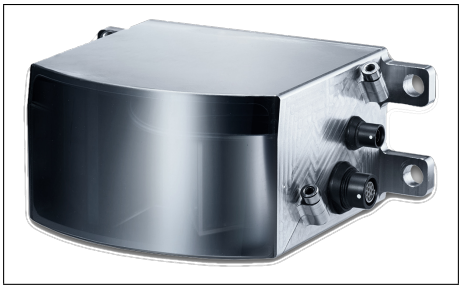

Figure 3.2: Ibeo Scala

Connector layout with three separated connectors for:

- Power
- Ethernet connection (Valeo Media Converter)
- Additional Interfaces (CAN)

NOTE
All Scala sensors must be configured with the correct mounting position (x, y, z position and yaw angle). Please refer to the Ibeo Scala User Manual for further instructions.

External processing unit (ECU)

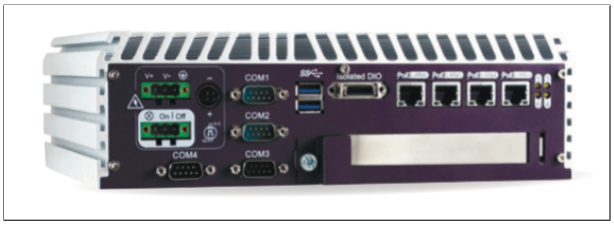

Figure 3.3: External processing unit (ECU)

Dimension: 260mm x 215mm x 79mm
This unit receives the raw data from all scanners and performs the object detection and tracking algorithms. Also the ECU requires vehicle motion data via CAN as an input to perform the object detection and tracking. The output (processed data, object data, etc.) of the ECU is provided via Ethernet. Object Data is also available via CAN. Please refer to the Ethernet Specification and CAN Specification document for more details about the output format.

Interfaces for:

- Power
- 6x Ethernet
- CAN

Required software versions:

- Firmware: 6.0.6 or later. Please check the main page of the web interface for the version number.

Power Box

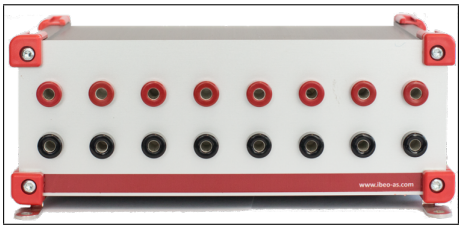

Figure 3.4: Power Box

Dimension: 170mm x 141mm x 67mm

The Ibeo Power Box is used for powering all components within the Fusion System. It needs one input connection that can be distributed to all other devices connected to the Power Box. The power supply (input) can be plugged to any connector.

CAUTION
Make sure there is just one power supply connected to the Power Box!

Recommended input voltage is 12V.

CAN Gateway/ Bridge (optional)

Figure 3.5: CAN Gateway/ Bridge

A CAN Gateway/ Bridge is a device that separates/ connects two CAN-Buses. In this case the Ibeo CAN bus and the Car CAN bus where the vehicle motion data comes from (refer to Figure 3.1 Car Architecture). The Ibeo CAN object output causes a high bus load, which might lead to conflicts and slow its speed dramatically down when using the Car and the Ibeo CAN bus together.

NOTE
This device is strongly recommended by Ibeo. However it is an optional part and is not needed for a correct function of the Fusion System.

Ibeo Automotive Systems is using the IXXAT CANbridge. This device can be purchased as a part of the Fusion Package at Ibeo.
Operating temperature: -20°C ... +70°C

XSens GPS/INS (optional)

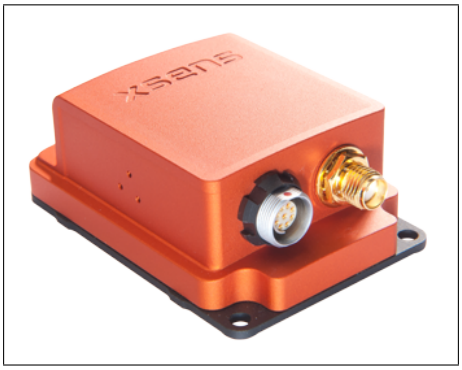

Figure 3.6: XSens GPS/INS - Reference: XSens MTi User Manual

Figure 3.6 shows the XSens GPS/INS device, which can optionally be added to the Ibeo Sensor System. It will add GPS and gyro information, as figure 3.7 illustrates, to the Ibeo data stream. This data can be accessed by using the IbeoSDK or can also be visualized using the ibeo Laser View.

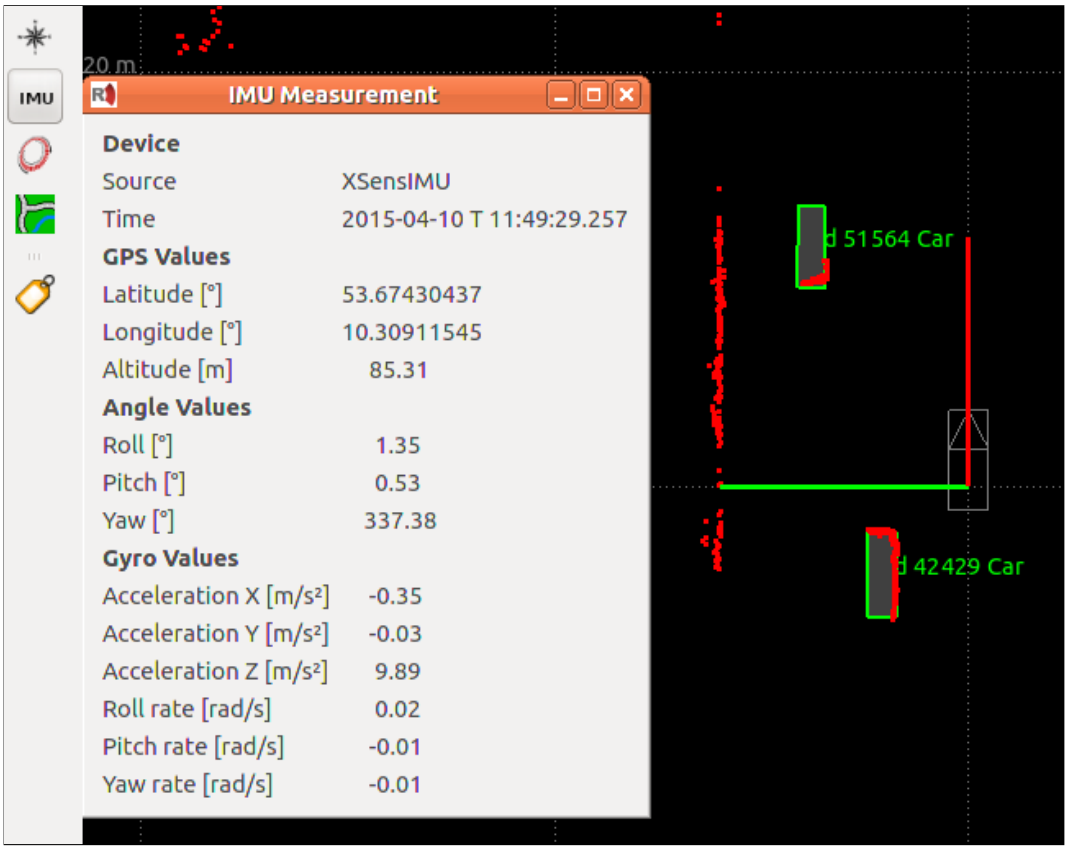

Figure 3.7: ibeo Laser View - IMU Visualization widget

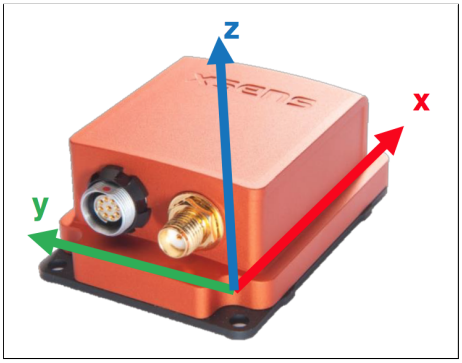

Figure 3.8: XSens Coordinate System - Reference: XSens MTi User Manual

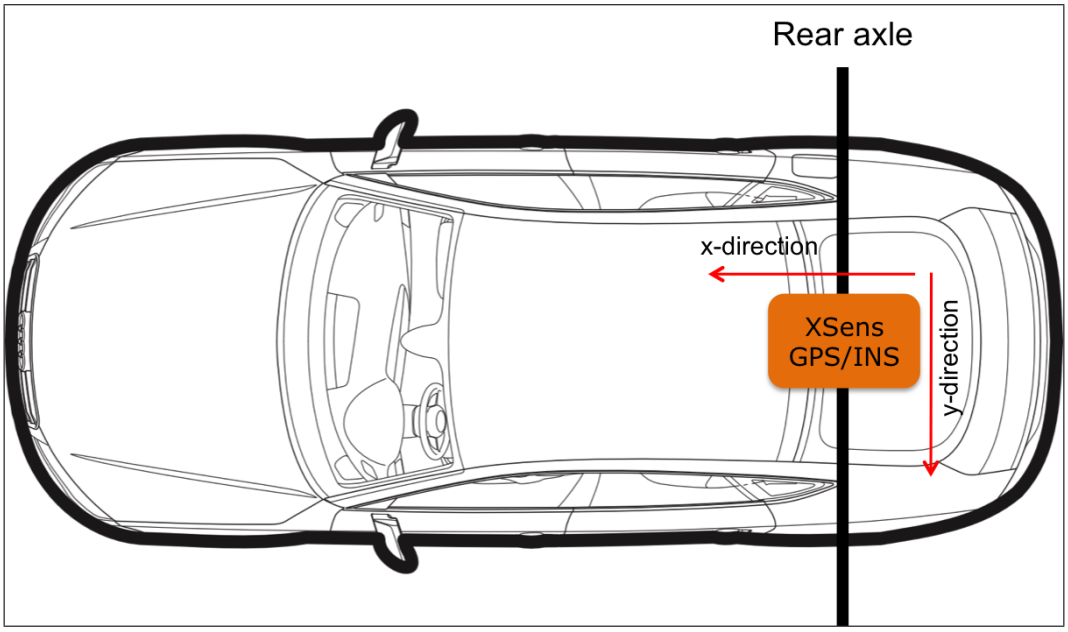

Figure 3.9: XSens GPS/INS mounting

NOTE
Ibeo recommends to mount the XSens GPS/INS in x-axis direction above the center of rear axle with 0° Pitch, 0° Roll and 0° Yaw.

For installation the XSens GPS/INS has to be connected via USB to the ECU and the GPS antenna has to be connected to the XSens GPS/INS. Please use the appropriate USB cable (with included converter in the USB cable).

For configuration the following section has to be added to the ibeo ECU "system.xml" configuration file:

```
<config type="Xsens" name="Xsens">
    <param name="DeviceID">7</param>
    <param name="Device">/dev/ttyUSB0</param>
    <param name="XKFScenarioId">2</param>
    <param name="XsensType">2</param>
</config>
```

NOTE

The "DeviceID" parameter has to be adjusted to an available value.

No further configuration is needed, the ibeo ECU performs all necessary configuration on the XSens GPS/INS device. For validation of correct functionality, please see the ibeo Laser View "IMU Measurement" widget if all values are filled and the XSens GPS/INS is working properly.

4 Ibeo ECU

The Electronic Control unit is an embedded PC, running all scan data processing. It uses an embedded LINUX operating system, installed on a solid state disc.

Processor	Intel Quad Core i7 4x2.3Ghz
RAM	8GB
SSD	64GB
Voltage Supply DC	6V...36V
Dimensions	(260x215x79)mm
Operating Temperature	-25°C...+70°C
Relative Humidity	10%...95%

Table 4.1: ECU Specification

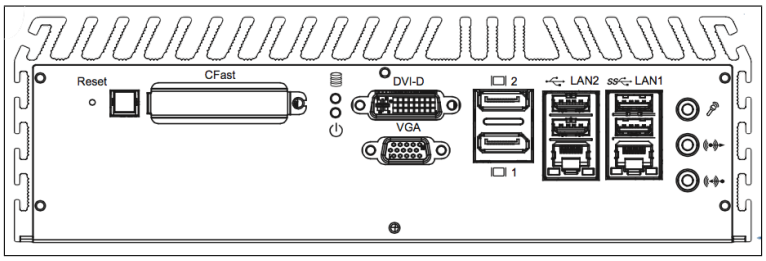

Figure 4.1: ECU Front View

Figure 4.1 shows the front view of the ECU. The power switch is not used in normal operation, because the device is set up to boot automatically as soon power is supplied.
For element function refer to Table 4.2

LAN1	Ethernet Port 1
LAN2	Ethernet Port 2
USB	USB 3.0 connection for optional devices. E.g. Video Camera, XSens
DVI-D	Debug only
VGA	Debug only
CFast	Not used
DisplayPorts	Not used

Table 4.2: ECU front components

Figure 4.2 shows the rear view of the ECU with typical connectors for a PC.

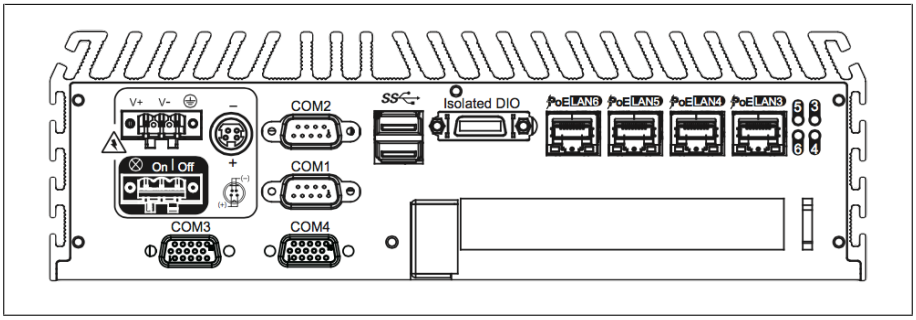

Figure 4.2: ECU rear view

Within this system the following connectors are used:

LAN3-6	Ethernet Port 3-6
Isolated DIO	Not used
USB	USB 3.0 connection for optional devices. E.g. Video Camera, XSens
COM1-4	Not used
DC Power Supply	Power supply (6V to 36V)

Table 4.3: ECU rear components

4.1 Web Interface

The Web Interface for status and configuration of the system is reachable on Ethernet Port 1 by using your web browser (default Address: 192.168.0.100, Port 80).

Figure 4.3: System Status

Figure 4.3 shows the startup page of the ECU web interface. This page provides information about

- The firmware version of the Ibeo ECU, in this case 6.0.4
- The system status, such as CPU usage and temperature
- The SSD memory usage
- The current software package and the status. In this case the software package FeatureFusion 5.7.0 is running

Four tabs on the left provide submenus.

4.2 System Files

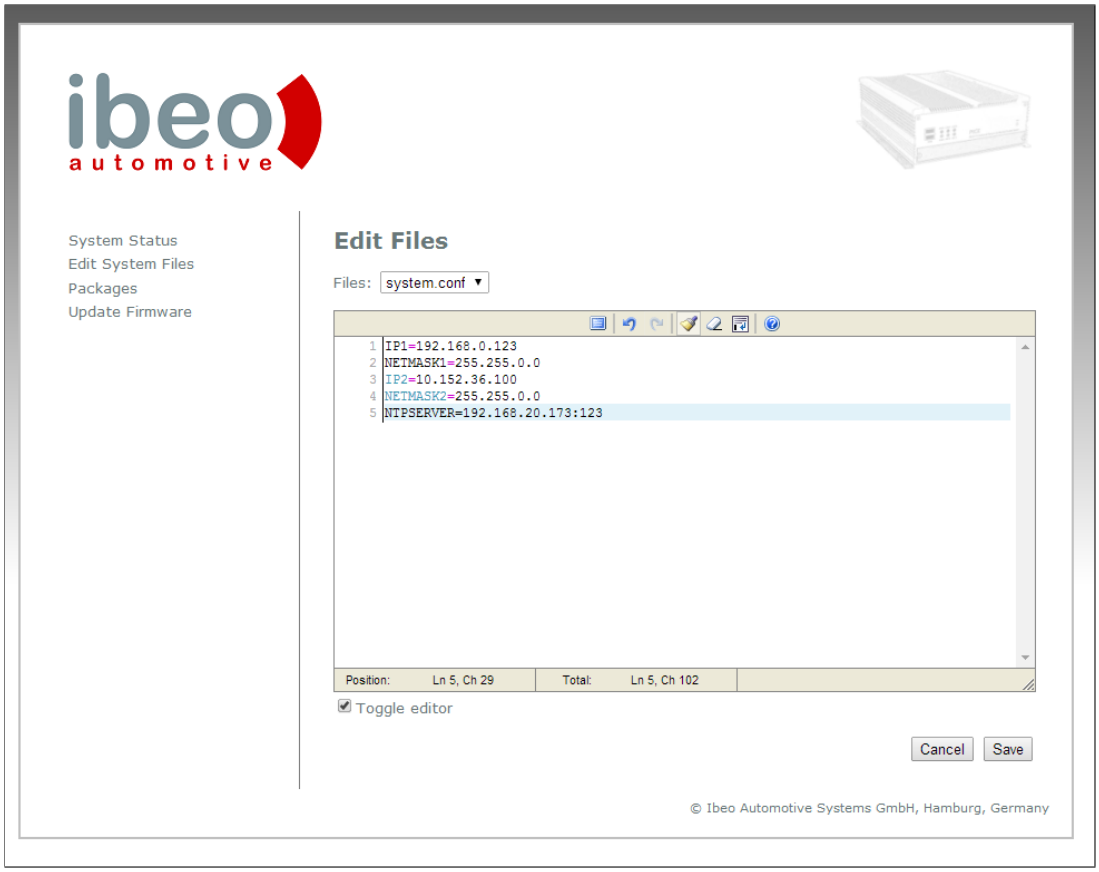

Figure 4.4: system.conf configuration

The tab “Edit System Files” allows you to change the IP address and netmask for Ethernet port one (interface network to ILV, Data logging ...) and two (sensor network). The ECU can connect to a NTP time server via ethernet port one. To enable NTP time sync the IP address of the time server needs to be configured via the web interface of the ECU. Select "Edit System Files" in the menu on the left (Refer to Figure 4.4).

Example:

- NTPSERVER=192.168.0.100:123
- Important: (IP-Address):(Port)
- Default Port for NTP is 123.

All datatypes will be timestamped according to the synchronized ECU clock.

NOTE
NTP time server synchronization is only available for Firmware 6.0.6 and newer!

4.3 Packages Page

The tab “Packages” provides an overview of installed packages as illustrated in Figure 4.5. It shows a table of installed packages. These packages can be started, stopped and deleted via this page. New packages can be installed using the drop down box at the top of the page.

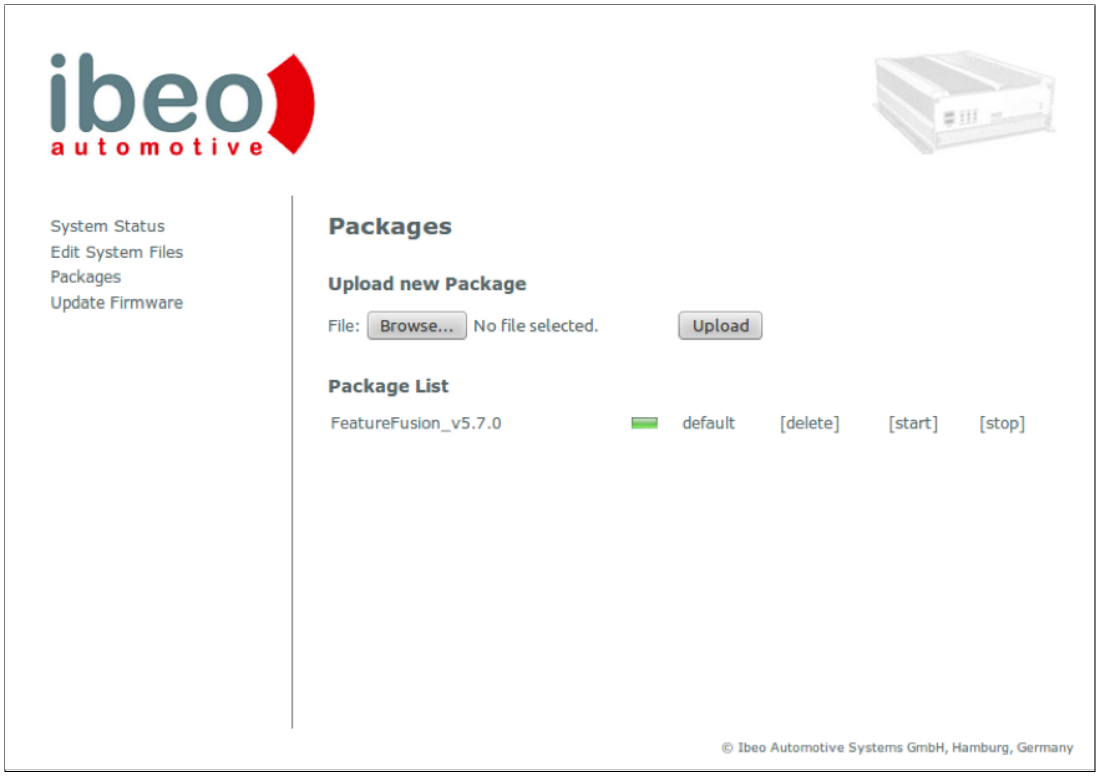

Figure 4.5: Packages Page

More detailed information and configuration possibilities for each package are shows by clicking on the name of a package.

At the very bottom of each package page a text editors allows to change the .xml-files which configure the system. These files are

- **System.xml**
  - Names the attached sensors to connect to with their IP addresses
  - Names and configures the CAN interface for vehicle motion data and optional CAN object output
  - Sets required vehicle parameters (length, width, axle distance,..) which are necessary information for the ego motion estimation
  - Select data type version for Ethernet output
  - Select output validation criteria for Objects
  - Optional object CAN spec configuration
  - Optional names camera parameters
- **CAN Parser.xml**
  Within this file the configuration of the CAN parser for vehicle speed and yaw rate is configured:
  - Assignment of CAN Message IDs to the different ego motion data
  - Specification of endianess, range, bits used within the message

4.4 Software Package Configuration System.xml

This section describes the system configuration.

Sensor configuration

Each device connected to the system is assigned to a unique name and a device ID. Make sure that all names and device IDs are unique. The range for device IDs can be selected from 1 to 254. The sensor's IP address must be selected according to the IP address of the sensor that should be placed into the application.

Parameter	Data type	Default	Description
DeviceID	UINT8	0	Unique Device ID in the whole Ibeo System
IPAddress	UINT8.UINT8.UINT8.UINT8	-	IP address of the sensor

Table 4.4: Sensor Configuration

Camera Configuration

The camera is connected to any USB port on the Fusion ECU. In case there is no camera connected to the system, this configuration block must be removed from the configuration file.

Parameter	Data type	Default	Description
Device	String	/dev/video0	Path to device file
DeviceID	UINT8	20	Unique device ID in the whole Ibeo System
MaxFPS	UINT8	25	Camera recording speed (frames per second)
Height	UINT16	480	Camera dependent pixel height
Width	UINT16	640	Camera dependent pixel width

Table 4.5: Camera Configuration

Following values can be set:

- DeviceID: 20, 21, 22, 23
- MaxFPS: 15, 25, 30
- (Height x Width): 600x800, 480x640, 240x320

For the use with more than 2 cameras it is recommended to lower down either the MaxFPS or the resolution, depending on the processing and ethernet speed of the system.

Example configuration for 4 cameras:

```
<config type="Camera" name="Camera">
    <param name="Device">/dev/video0</param>
    <param name="DeviceID">21</param>
    <param name="MaxFPS">25</param>
    <param name="Height">240</param>
    <param name="Width">320</param>
</config>
<config type="Camera" name="Camera1">
    <param name="Device">/dev/video1</param>
    <param name="DeviceID">22</param>
    <param name="MaxFPS">25</param>
    <param name="Height">240</param>
    <param name="Width">320</param>
</config>
<config type="Camera" name="Camera2">
    <param name="Device">/dev/video2</param>
    <param name="DeviceID">23</param>
    <param name="MaxFPS">25</param>
    <param name="Height">240</param>
    <param name="Width">320</param>
</config>
<config type="Camera" name="Camera3">
    <param name="Device">/dev/video3</param>
    <param name="DeviceID">24</param>
    <param name="MaxFPS">25</param>
    <param name="Height">240</param>
    <param name="Width">320</param>
</config>
```

CAN Device Configuration

Besides the sensors a CAN bus should be connected to the system to enable the use of ego motion data from the vehicle as well as for providing object data on this interface.
The assignment principle is the same as for the connected sensors: A CAN device generally is provided to the system as a device from type CAN bus. In this example the CAN bus is assigned to the (unique) name CAN bus and a device ID of 10.
Ensure that the used device ID is different from all other device IDs in the systems, including the sensor device IDs.
The baudrate of the bus can be set to 10 kbit/s, 125 kbit/s, 250 kbit/s, 500 kbit/s, 1000 kbit/s.
The last column names the device driver to be used for the CAN interface. The device driver must never be changed since this is a system internal and system dependent option. Use always /dev/pcan0.
If no CAN bus wiring is connected to the system, this part of the configuration file should be deleted or commented out.

Parameter	Data type	Default	Description
Device	String	/dev/pcan0	Path to device file
DeviceID	UINT8	10	Unique device ID in the whole Ibeo System
BaudRate	UINT16	500	Baudrate in kBaud
MsgFilter	CSV(UINT11)	-	Whitelist of allowed CAN messages. If no IDs are specified all messages are passed through.

Table 4.6: CAN Device Configuration

ObjectFilterWorker Configuration

This worker enables you to only let validated objects be put on the interfaces like Ethernet (optional : or CAN)

Parameter	Data type	Default	Description
filterIsActive	Boolean	True	Should this worker be used
acceptInvalidDynamic	Boolean	False	Refer to definition of object flags
acceptInvalidStationary	Boolean	False	Refer to definition of object flags

Table 4.7: ObjectFilterWorker Configuration

Definition of object flags

Valid	Mobile detected	Static Model/ Dynamic model	Description	Customer IF relevant	Put out for evaluation
0	0	0	Unvalidated object hypotheses	no	yes
0	0	1		no	yes
0	1	0	n.a.	no	no
0	1	1		no	no
1	0	0		no	no
1	0	1	Validated stationary object (“a priori stationary”)	yes	yes
1	1	0	validated dynamic object with validated track (“moving”)	yes	yes
1	1	1	Validated stationary object (“a priori stationary”)	yes	yes

Table 4.8: Definition of object flags

DataTypeSelectorWorker Configuration

This worker enables you to chose the datatypes sent over Ethernet.

Parameter	Data type	Default	Description
ObjectListDataType	Hex16	0x2280	Chose 0x2225 for backward compability, refer to the Customer Ethernet Document for contents and structure of the datatypes
VehicleStateDataType	Hex16	0x2807	Chose 0x 2806 for backward compability, refer to the Customer Ethernet Document for contents and structure of the datatypes

Table 4.9: DataTypeSelectorWorker Configuration

Motion Estimation Settings

For correct motion estimation knowledge, some basic mechanical parameters of the vehicle are necessary

Parameter	Data type	Default	Description
FrontAxleToRearAxle	Float m	1.0 m	Distance front to rear axle in m
MinTurningDiameter	Float m	1.0 m	Minimum diameter for turning in m
RearAxleToVehicleRear	Float m	1.0 m	Distance vehicle rear to rear axle in m
VehicleFrontToFrontAxle	Float m	1.0 m	Distance vehicle front to front axle in m
VehicleWidth	Float m	1.5 m	Width of the vehicle in m
VelCorrectFactor	Float	1.0	Factor to be multiplied with the velocity signal in order to compensate for a linear error in the vehicle speed. Might be caused by variations in the tire diameter.

Table 4.10: Motion Estimation Settings

Object to CAN Conversion (optional)

Parameter	Data type	Default	Description
CANBaseID	HEX 11Bit	0x500	The Base ID to send object information
SortByDynamicFirst	Boolean	True	Defines prioritization method when not all objects can be sent on the CAN bus. If true, all dynamic objects will be sent first followed by stationary objects; if false, closest objects will be sent first.
MaxCANObjects	UINT	5	Max. number of objects that should be transmitted on CAN bus
SendRelativeVelocity	Boolean	False	Defines whether object velocities are given relative to the host vehicle or in the absolute frame.
SendBoundingBox	Boolean	False	Defines whether a bounding box in x/y coordinates (true) or an oriented object box (false) is sent.
SendObjectContour	Boolean	True	Defines whether the object contour is sent or not. If this is set to false, only the closest point is sent out. This ensures that always at least one point is sent out.
Version	Float	2.0	Please refer for v2.0 to the Customer CAN Document

Table 4.11: Object to CAN Conversion

4.5 Software Package Configuration CANParser.xml

The CAN Parser is responsible for extracting the necessary data from received CAN messages. For correct ego motion compensation at least the vehicle velocity and yaw rate must be available.
For each value to retrieve and extract from the CAN interface the following parameters must be configured

Parameter	Data type	Default	Description
Identifier	UINT11		CAN-ID
Factor	Float	1.0	CANmsg*Factor+Offset = Value [SI-Unit]
Offset	Float	0.0	CANmsg*Factor+Offset = Value [SI-Unit]
Picture	Mask		64 Bit Mask; x is the location of the message in the whole 64bit CAN message
UseLittleEndian	True	False	Little Endian: Intel; Big Endian: Motorola
TwosComplement	False	True	Whether the value should be interpreted using twos complement representation
SignBit	UINT6	0	Location of the sign bit (ignored if SignBitAvailable is false)
SignBitAvailable	Boolean	False	Whether a sign bit is available
ErrorValue	UINT64	0xFFFFFFFF	If the value is equal to this error value, the output value will be considered invalid. Value must fit to the message length.
Picture	Char[64]	-	The picture is an array of dashes and x. It indicates which bits are used for the corresponding signal.

Table 4.12: CAN Parser

Example

Here is a description of an example CANParser configuration.
Assuming the following data on the CAN bus for velocity:

- The velocity is provided on the vehicle CAN with message ID 0x023
- The resolution is 1/100 km/h per bit
- No negative values are available and no dedicated sign bit
- When the vehicle is running backwards, the velocity is set to 0 on CAN Bus
- The value is coded in big endian format
- The data size is 15 bit long
- The start bit is located in bit 35

```
<config type="CANParser" name="VEH_Velocity">
    <param name="Identifier">0x023</param>
    <!-- System wants meters per second -->
    <!-- significance in m/s of one digit, eg. 0.002778 for conversion from 1/100
        km/h to m/s -->
    <param name="Factor">0.0027777777</param>
    <!-- Offset is applied after Multiplication with "Factor" -->
    <param name="Offset">0.0</param>
    <!-- Byteorder 0 1 2 3 4 5 6 7 -->
    <param name="Picture">-------- ------xx xxxxxxxx xxxxx--- -------- --------
        -------- --------</param>
    <param name="UseLittleEndian">false</param>
    <param name="SignBitAvailable">false </param> -->
    <param name="TwosComplement">false</param>
</config>
```

Assuming the following data on the CAN bus for yaw rate:

- The yaw rate is provided on the vehicle CAN with message ID 0x023
- The resolution is 0.05 deg/s per bit
- Negative values are available and a dedicated sign bit is available (Pos. 19)
- The value is coded in big endian format
- The data size is 15 bit long
- The start bit is located in bit 20

```
<config type="CANParser" name="VEH_YawRate">
<param name="Identifier">0x023</param>
<param name="Factor">0.000872</param> <!-- conversion from 0.05 deg/s to rad/s -->
<param name="Offset">0</param>
<param name="Picture">-------- -------- -------- ------xxx xxxxxxxx xxxx---- --------
    --------</param>
<param name="UseLittleEndian">true</param>
<param name="TwosComplement">true</param>
<param name="SignBit">19</param>
<param name="SignBitAvailable">true</param>
</config>
```

In case CAN message for velocity does not contain any sign bit for driving backwards, there is a third parser called Driving Condition that will read the sign bit from another CAN message:

```
<config type="CANParser" name="VEH_DrivingCondition">
<param name="Identifier">0x1A0</param>
<param name="Factor">1</param>
<param name="Offset">-3</param> <!-- = -1 when driving backwards -->
<param name="Picture">-------- -xxx---- -------- -------- -------- -------- --------
    --------</param>
<param name="UseLittleEndian">true</param>
<param name="TwosComplement">false</param>
<param name="ErrorValue">0x7</param>
</config>
```

Factor calculation

The factor must be configured such that the CAN bus unit of 1/100 km/h is transferred into the unit of the system which is m/s.

$$1km = 1000m, 1h = 3600s$$

$$1/100 * km/h = 1/100 * 1000m/3600s = \ 0.002778m/s \tag{1}$$

Offset determination

In this example no offset setting is necessary, because the velocity value increases linearly from 0 to max.

Bit pattern picture

The bit pattern picture represents the complete 64 Bit field of the CAN-Message from the left, starting with byte 0. The location of the value to be extracted is marked with 'X', the remaining other (don't cares) with '-'. In this case 15 bits long starting at bit 35 ending at bit 49:
"––– ––xx xxxxxxxx xxxxx– ––– ––– ––– ––– "

Endianess

As the values on the CAN bus often exceeds single bytes, the endianess needs to be defined in which the data is coded.
Please refer to the following table, to understand which endianess you are using.

Endianess	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Little	7 0	15 8	23 16	31 24	39 32	47 40	55 48	63 56
Big	63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0

Table 4.13: Endianess

5 Software Architecture

5.1 Required Input Signals

In order to perform the object detection and tracking, the Ibeo ECU requires some input signals. These are mainly the motion data of the host vehicle.
The required signals are:

Parameter	Data type	Description
Velocity	float	The speed of the host vehicle in m/s
Yaw rate	float	The turn rate of the host vehicle in rad/s

Table 5.1: Required Input Signals

For a robust function of the object detection and tracking these input signals should be provided at an update rate of more than 50Hz.

NOTE
The accuracy of the provided the input signals is directly influences the motion estimation of detected objects. Therefore the accuracy as well as the latency of these signals has major influence on the object detection performance.

5.2 Processing Structure

This section describes the processing structure of the object detection and tracking algorithms.

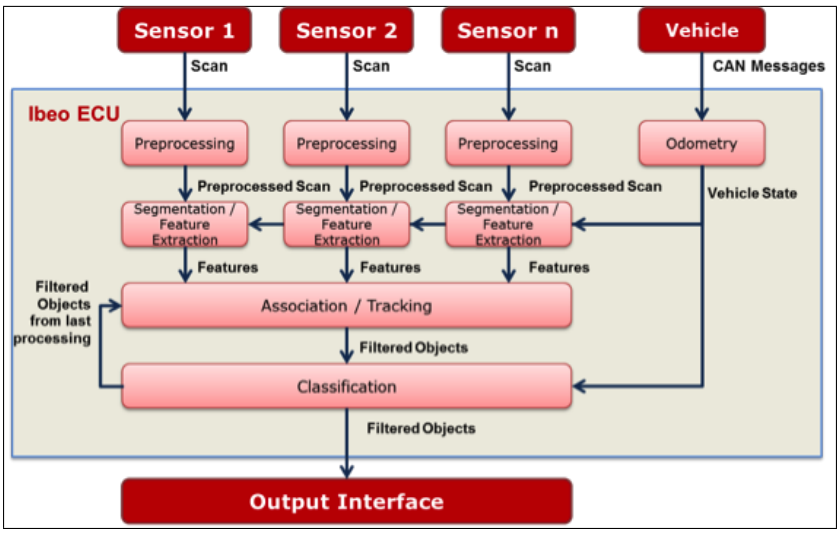

Figure 5.1: Object detection and tracking processing structure

Preprocessing

In order to not use non valid points for tracking and classification, several computation steps are done. One of these steps is called Preprocessing where dirt, clutter and ground points are marked. They are not used for segmentation, but remain available for further analysis steps.

Segmentation and feature extraction

These segments are groups of associated scan points. The segmentation depends on the vehicle's own movement. Furthermore a set of features are extracted from the segments. Among others, contour information extracted from segments is part of the set of features. In this computation step there is an additional validation algorithm based on some plausibility checks that is able to make single measurements to be disregarded by the tracking. This could be called "preprocessing on segmentation level".

Tracking

During tracking, objects are built from the segments. In this process, the vehicles own movement and the objects of the previous tracking calculations (history) are also used. Objects have the following main properties:

- Position of a reference point
- Size, orientation
- Outline
- Speed

Different representations are available for an object:

- Outline: the contour of the object as facing the Feature Fusion System as a polygon, based on scan data
- Object box: object position, orientation and size from the tracking

Objects are generated from segments and tracked on a time-base. In doing so, object speeds and other object properties are determined.
The outline of the object is a contour that is filtered over time, rather than an instantaneous measurement based on current scan points. This filtering improves the robustness of the system to noisy measurements, and allows the shape of the object to be maintained if part of the object cannot be seen for a short time.
The speed is issued both relative and absolute (if the vehicle's movement is available via a CAN bus):

- Absolute speed is the speed of a vehicle on the street or viewed from a stationary observer
- Relative speed is the speed of a vehicle relative to an observer. While approaching a stationary target with a constant speed v, the corresponding object has the relative speed -v Outline
- Relative and absolute speed are identical as long as the own vehicle does not move

Classification

In the last processing step, the classification, the following classes are assigned to the objects, on the basis of properties and history:

- Car
- Bike (motorcycle, bicycle)
- Truck
- Pedestrian
- Unknown class, large object
- Unknown class, small object

5.3 Scan Fusion

All laser scanners in the fusion system are synchronized to take their scan all at the same time with the same mirror side. Although the fusion algorithm is not based on raw scan data fusion, all scans that are taken at the same time are fused by the Ibeo ECU for visualization and output.
For each set of scans that were taken simultaneously, an update of the object tracking is calculated. In order ot minimize the object output latency, the object output is triggered as soon as the computation is done.

5.4 Ethernet Output Data

For more detailed information about the Ethernet interface please refer to the Ethernet specification document that is delivered with this system.
The following data is transmitted by the Ibeo ECU via Ethernet.

- Scan Data: 0x2205 (default)
- Object List: 0x2280 (default), 0x2225
  - Timestamps: Objects are always given the timestamp of the latest measurement that was used to update the object. In case a GPS receiver is connected to the system all timestamps are given with respect to the GPS time signal.
- Vehicle State: 0x2807 (default), 0x2806
- Image: 0x2403 (default)

5.5 CAN Output Data

The software package allows sending object data via CAN at an update rate of 25Hz. Due to the high update rate only few objects can be transmitted per cycle. The number of objects to be output by the Ibeo ECU via CAN each cycle can be configured via the web interface. This number should not be more than ten objects per cycle, to allow for a stable data transmission. Additionally criteria for object prioritization can be specified.
For the interface description please refer to the CAN specification document that is delivered with this system.