Sensor Stream Pipe

Generated by Doxygen 1.8.13

Contents

Chapter 1

Sensor Stream Pipe Instalation

Linux instructions

Windows instructions

Linux

To get our Sensor Stream Pipe up and running, you will require the following:

The following steps were tested on Ubuntu 18.04. Installing on other recent Linux distributions should be pretty similar, but please check the installation instructions for OpenCV and Kinect DK on your respective platform first. Installation instructions for Windows should be ready soon. If you encounter any problems or have any suggestions, please let us know by emailing contact@moetsi.com or post on our forum.

Dependencies

To get our Sensor Stream Pipe up and running, you will require the following:

- OpenCV 3.2.0 (tested on version available on Ubuntu 18.04 repo) is used for image processing.
- libav 3.4.6 (tested on version available on Ubuntu 18.04 repo) encodes, decodes and processes image frames.
- Cereal 1.2.2 (headers only) serializes data for network transmission.
- ZeroMQ and cppzmq (libzmq3 4.3.1, cppzmq 4.3.0) perform network and low-level I/O operations.
- spdlog 1.4.1 Logging library.
- yaml-cpp 0.6.0 reads server configuration files.
- Zdepth: compresses depth data.
- NvPipe (*optional*, but **recommended if you have an NVidia GPU**) encodes and decodes frames. This is optional, but recommended for users with Nvidia GPUs.
- Azure Kinect SDK 1.3 (to support the Azure Kinect Body Tracking SDK), 1.4 otherwise (*optional*) accesses Kinect DK data.
- Azure Kinect Body Tracking SDK 1.0 (*optional*) SSP Body Tracking client.

Download and install repo libraries

OpenCV 3.2.0

```
sudo apt install libopencv-dev libopencv-core-dev uuid-dev
```

Libav 3.4.6

```
sudo apt install libavformat-dev libavutil-dev libavcodec-dev libavfilter-dev
```

Download and extract "out-of-repo" libraries

First, create a folder where local libs are to be installed:

```
mkdir ~/libs
mkdir ~/libs/srcOriginal
```

Cereal 1.2.2

```
cd ~/libs/srcOriginal
wget https://codeload.github.com/USCiLab/cereal/tar.gz/v1.2.2
tar xf v1.2.2
cp -r cereal-1.2.2/include ~/libs
```

ZeroMQ

libzmq3 4.3.1

```
cd ~/libs/srcOriginal
wget https://github.com/zeromq/libzmq/releases/download/v4.3.1/zeromq-4.3.1.tar.gz
tar xf zeromq-4.3.1.tar.gz
cd zeromq-4.3.1
mkdir build
cd build
cmake .. -DCMAKE_INSTALL_PREFIX=~/libs
make install -j4
```

If you want to take advantage of ZMQ pulling support to check for new frames, compile ZMQ and CPPZMQ with the `draft API suppport`. Replace the cmake line with:

```
cmake .. -DENABLE_DRAFTS=ON -DCMAKE_INSTALL_PREFIX=~/libs
```

Also, you must set SSP_WITH_ZMQ_POLLING at SSP build time.

cppzmq 4.3.0

```
cd ~/libs/srcOriginal
wget https://github.com/zeromq/cppzmq/archive/v4.3.0.tar.gz
tar xf v4.3.0.tar.gz
cd cppzmq-4.3.0
cp *.hpp ~/libs/include
```

yaml-cpp 0.6.0

```
cd ~/libs/srcOriginal
wget https://github.com/jbeder/yaml-cpp/archive/yaml-cpp-0.6.0.tar.gz
tar xf yaml-cpp-0.6.0.tar.gz
cd yaml-cpp-yaml-cpp-0.6.0
mkdir build
cd build
cmake .. -DCMAKE_INSTALL_PREFIX=~/libs
make install
```

Zdepth

```
cd ~/libs/srcOriginal
git clone https://github.com/catid/Zdepth.git
cd Zdepth
mkdir build
cd build
cmake .. -DCMAKE_INSTALL_PREFIX=~/libs
make install
cp libzdepth.a ~/libs/lib/
cp zstd/libzstd.a ~/libs/lib/
```

spdlog

```
cd ~/libs/srcOriginal
wget https://github.com/gabime/spdlog/archive/v1.4.1.tar.gz
tar xf v1.4.1.tar.gz
cd spdlog-1.4.1 && mkdir build && cd build
cmake .. -DCMAKE_INSTALL_PREFIX=~/libs
make -j
make install
```

NVPipe (optional, recommended for users with Nvidia GPU)

```
cd ~/libs/srcOriginal
git clone https://github.com/NVIDIA/NvPipe.git
cd NvPipe/
mkdir build && cd build
cmake .. -DCMAKE_INSTALL_PREFIX=~/libs
make
make install
```

Azure Kinect SDK 1.3/4 (optional)

Note: to avoid getting a password prompt, run any command as sudo before starting this section of the tutorial

1) Add the Linux Software Repository for Microsoft Products.

```
curl https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add -
sudo apt-add-repository https://packages.microsoft.com/ubuntu/18.04/prod
sudo apt-get update
```

2) Install Azure Kinect SDK 1.3 (change 1.3 to 1.4 if you do not need body tracking)

```
sudo apt install libk4a1.3 libk4a1.3-dev
```

3) To be able to use the Kinect as non-root, please run the following:

```
wget https://raw.githubusercontent.com/microsoft/Azure-Kinect-Sensor-SDK/develop/scripts/99-k4a.rules
sudo cp 99-k4a.rules /etc/udev/rules.d/
```

4 a) If using 1.4, in the current package, the link to the canonical version of the depth lib is missing. You can create it by running the following command:

```
sudo ln -s /usr/lib/x86_64-linux-gnu/libk4a1.4/libdepthengine.so.2.0
       /usr/lib/x86_64-linux-gnu/libdepthengine.so
```

4 a) If using 1.3, the depth engine is missing from the package. Microsoft is aware of the problem, but it only corrected it in 1.4.

https://github.com/microsoft/Azure-Kinect-Sensor-SDK/blob/develop/docs/depthengine.md

You can get the libdepthengine.so.2.0 file from the package at https://packages.microsoft.com/ubuntu/18.04/prod/pool/main/libk/libk4a1.4/libk4a1.4_1.4.1_amd64.deb. Open with Archive Manager (or equivalent), the file is in (data/./usr/lib/x86_64-linux-gnu/libk4a1.4/), and extract it to ~/libs/lib. You can then perform an equivalent command as above.

```
cd ~/libs/lib
ln -s libdepthengine.so.2.0 libdepthengine.so
```

Azure Kinect Body Tracking SDK (optional)

Check instructions above to add the Linux Software Repository for Microsoft Products and then do:

```
sudo apt install libk4abt1.0-dev
```

Building Sensor Stream Pipe

Download and build the project (the ssp_server, ssp_client and ssp_tester):

```
git clone git@github.com:moetsi/Sensor-Stream-Pipe.git
cd Sensor-Stream-Pipe
mkdir build
cd build
cmake .. -DSSP_WITH_KINECT_SUPPORT=OFF -DSSP_WITH_K4A_BODYTRACK=OFF -DSSP_WITH_NVPIPE_SUPPORT=OFF
make
```

You can turn on Kinect, Bodytrack and NVPipe support by adding the following to the cmake .. line respectively:

```
-DSSP_WITH_KINECT_SUPPORT=ON
-DSSP_WITH_K4A_BODYTRACK=ON
-DSSP_WITH_NVPIPE_SUPPORT=ON
```

Windows

Windows installation process was performed using vcpkg to install most dependencies. Tested on Windows 10 Build 19041, Visual Studio 2019 Community Edition (VS).

This process may also work for Linux, but this was not tested.

Install vcpkg

Follow vcpkg installation instructions available here

Install dependencies available on vcpkg

Install dependencies using vcpkg.

```
vcpkg install azure-kinect-sensor-sdk:x64-windows cereal:x64-windows cppzmq:x64-windows ffmpeg:x64-windows
       opencv3:x64-windows spdlog:x64-windows yaml-cpp:x64-windows zeromq:x64-windows
```

Build and install remaining dependecies

Prepare a directory to place the remaining dependecies lib and include files (refered henceforth as `$LIBS`). This directory should have a `lib` and `include` subfolders with the corresponding `.lib` and headers respectively.

Zdepth

Clone Zdepth repo

```
git clone https://github.com/catid/Zdepth.git
```

Open CMakeLists file in VS and build accorcing to your desired profile (x86 or x64; Debug or Release).

If you did not specify an install dir during the CMake configuration, copy the `Zdepth\include` and output lib folders (e.g. `ZDepth\out\*`) to `$LIBS`.

Azure Kinect Body Tracking SDK (optional)

Install Azure Body Tracker SDK from the instructions available `here`.

Copy the SDK include and lib files from the SDK install list to `$LIBS`, or add the SDK path to SSP CMakeLists (see below)

Building Sensor Stream Pipe

Clone the SSP repo

```
git clone git@github.com:moetsi/Sensor-Stream-Pipe.git
```

Due to the diferences in the build process, the Windows CMake file is named CMakeListsWindows.txt at the root of the SSP repo.

Thus, you shoud delete CMakeLists.txt and rename CMakeListsWindows.txt to CMakeLists.txt.

Open CMakeLists.txt in VS.

Replace/Add the include ("C://Users//Andre//source//repos//vcpkg//installed//x64-windows//include") and link paths ("C://Users//Andre//source//repos//vcpkg//installed//x64-windows//lib") at the top of the file with your `$LIBS` paths

```
include_directories("C://Users//Andre//source//repos//vcpkg//installed//x64-windows//include")
link_directories("C://Users//Andre//source//repos//vcpkg//installed//x64-windows//lib")
```

You can also add your `vcpkg//installed//` dir to the include and link paths.

After replacing the paths, set the desired compile options (SSP_WITH_KINECT_SUPPORT, SSP_WITH_K4A_B↵ODYTRACK, ...), regenerate CMakeCache and build the project.

Linking errors?

if you have linking errors (missing .lib files), try replacing the short lib name with the full lib path in CMake: "libzmq" -> "C:/Users/Andre/source/repos/vcpkg/installed/x64-windows/lib/libzmq.lib"

Chapter 2

[Checkout the Sensor Stream Pipe gitbook for full documentation](https://moetsi.gitbook.io/sensor-stream-pipe/)

What is Sensor Stream Pipe?

Moetsi's Sensor Stream Pipe (SSP) is the first open-source C++ modular kit-of-parts that compresses, streams, and processes sensor data (RGB-D). It does this by efficiently compressing raw data streams, allowing developers to send multiple video types over the network in real time. Frame data can be sent in its raw form (JPG/PNG frames), or compressed using a myriad of codecs, leveraged on FFmpeg/LibAV and NV Codec to considerably reduce bandwidth strain.

SSP is designed to help overcome the limitations of on-device sensor data processing. By taking data processing off device, you will be able to run far more powerful computations on your sensor data and make the most of the tools at your disposal.

The Moetsi Sensor Stream Pipe is designed to overcome the limitations of on-device sensor data processing. It does this by encoding and compressing your device's color or depth frames, and transmitting them to a remote server where they can be decoded and processed at scale.

Currently, Moetsi's Sensor Stream Pipe supports:

- .mkv (matroska) RGB-D recordings
- Azure Kinect DK RGB-D camera
- seminal computer vision/spatial computing datasets (e.g. BundleFusion, MS RGB-D 7 scenes and VSFS)
- iOS ARKit data (streams ARFrame data)

We're planning to support other cameras and devices (e.g. Kinect v2 and Structure Core) in the near future. At the same time, we've made the Moetsi SSP totally open source so that anyone can build out support for any device.

Checkout the Sensor Stream Pipe gitbook for full documentation

Features include:

- Synchronized streaming of color, depth and IR frames
- Support for Azure Kinect DK (live and recorded video streaming) and image datasets (e.g. BundleFusion, MS RGB-D 7 scenes and VSFS) and .mkv (matroska) files
- Hardware-accelerated encoding (e.g. Nvidia codec), providing you with the lowest possible latency and bandwidth without compromising on quality
- Interoperability with Libav and FFmpeg creates a hyperflexible framework for all the use cases you brilliant developers can come up with!
- Access to the calibration data for each of the sensors on the Kinect, enabling you to build a point cloud from the color and depth images, perform body tracking, etc.

But why though...?

- If you have 4 sensor streams and want to do an environment reconstruction using their data feeds
- If you have a couple of sensors and want to find where they are relative to each other
- You want to run pose detection algorithms on a dozen sensors and synthesize the results into a single 3D model
- Basically if you want to do any spatial computing/computer vision on multiple incoming data streams

You can use Sensor Stream Server to send compressed sensor data to reduce bandwidth requirements and Sensor Stream Client to receive these streams as an ingestion step for a computer vision/spatial computing pipeline.

If you want to synthesize RGB-D+ data from multiple feeds in real-time, you will probably need something like Sensor Stream Pipe.

Component parts

Checkout the Sensor Stream Pipe gitbook for full documentation

Sensor Stream Server

The ssp_server is the frame encoder and sender.

"Frames" are a sample of data from a frame source. For example, the Azure Kinect collects: RGB (color), depth, and IR data. If we want to stream RGB-D and IR, we sample our frame source (the Azure Kinect), and create 3 frames, one for each frame type: 1 for color data, 1 for depth data, and 1 for ir data. We then package these 3 frames as a zmq message and send through a zmq socket.

Sensor Stream Server reads its configurations from a yaml file (examples in /configs). The config file provides Sensor Stream Server: a destination for its frames, the frame source (video, Azure Kinect, or dataset), and how each frame type should be encoded.

Sensor Stream Client

The ssp_clients are the frame receiver and decoder. They run on the remote processing server and receive the frames from the ssp_server for further processing.

There are a few templates for how you can use Sensor Stream Client in

Sensor Stream Client with OpenCV processing

If you run Sensor Stream Client with OpenCV visualization:

You can see it's receiving real-time data from a Kinect DK and rendering it for on-screen display. In this scenario we achieved a substantial 20x data compression, reducing the stream size from 400 Mbps to just 20 Mbps, along with a PSNR of ∼39 dB and a processing overhead of ∼10-15 ms .

Sensor Stream Client is built so it can be an ingestion step for a spatial computing/computer vision pipeline.

`Sensor Stream Tester`

A reproducible tester for measuring SSP compression and quality. You can use this to measure how different encodings and settings affect bandwidth/compression.

Getting started

We recommend going through `Streaming a Video from our Gitbook`

to get up to speed quickly. You will stream using Sensor Stream Server and receive on Sensor Stream Client a pre-recorded RGB-D+ stream to get a quick feel of what Sensor Stream Pipe does.

Sensor Stream Pipe Development

Feedback

Moetsi's Sensor Stream Pipe is currently in alpha. Features will probably change, bugs will probably be found. It's a work in progress after all! That said, we welcome both feedback and pull requests.

We would also love to hear more about how you plan to use the Moetsi Sensor Stream Pipe! So if you have any problems, questions, feature requests, or ideas for improvement, please feel free to reach out at `olenka@moetsi.↵ com`.

The better we understand how you're using the Moetsi SSP, the better we can plan future developments!

About Moetsi

At Moetsi we are super excited about the idea of digitizing reality. Creating a seamless interface between the world as we know it, and a world augmented, improved and expressed through new technologies is plain cool. But we think there's a problem. On-device computation is limited, platform-specific frameworks are restrictive, and sorting raw depth data is seriously challenging.

To address the first problem, we've created the Moetsi Sensor Stream Pipe; to make it easier to process off-device without throttling bandwidth. It means you are no longer confined to the computational limits of your local device, and you don't have to make a massive trade-off on time-to-computation because our pipeline is super fast (latency is less than 30 ms for Kinect data).

But it doesn't end here.

Our pipeline is just one of the first pieces of the puzzle. To develop a robust enough infrastructure to support a true digital twin of the physical world, a lot more needs to be done. This includes creating algorithms that can turn this raw depth data into real, usable applications.

How to Contribute

We're always excited to work with like-minded people, and invite you to experiment with our pipeline however you like! If you enjoy our work and think you can help take this project to the next level, feel free to drop us a message on olenka@moetsi.com to get involved.

If you happen to discover any bugs in our code, we'd really appreciate knowing about them. Please just create an issue here on GitHub.

In terms of related projects that fall outside of this repo's scope, we'd be super excited to see, and think the community could benefit from development on:

∗∗More devices∗∗

Working with the Kinect v2, and other sensors such as the Structure Core sensor.

∗∗Integrations∗∗

Any other sort of output that you can imagine!

∗∗Encoding∗∗

Improve encoding performance on AMD/Intel graphic cards, by using the AMD Media Codec/Intel Quick Sync Video instead of relying on libav (VAAPI or OpenCK) for hardware accelerated encoding. Feel free to do the same for Intel cards using Intel Quick Sync Video too!

Moetsi's Permissive License

Moetsi's Sensor Stream Pipe is licensed under the MIT license. That means that we don't require attribution, but we'd really like to know what cool things you're using our pipe for. Drop us a message on olenka@moetsi.com or post on our forum to tell us all about it!

Support Moetsi!

Our Sensor Stream Pipe is always going to be free, but it has taken a lot of blood, sweat and tears to get to this point. If you love what we've made, please consider reaching out to olenka@moetsi.com.

Authors

- **André Mourão** - amourao
- **Olenka Polak** - olenkapolak
- **Adam Polak** - adammpolak

Chapter 3

Namespace Index

3.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 4

Hierarchical Index

4.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 5

Class Index

5.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 6

File Index

6.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 7

Namespace Documentation

7.1 moetsi::ssp Namespace Reference

MOETSI_RAAS

Classes

- struct AVCodecContextDeleter
- struct AVCodecDeleter
- struct AVCodecParametersDeleter
- struct AVCodecParametersNullDeleter
- struct AVFormatContextDeleter
- struct AVFrameDeleter
- struct AVIOContextDeleter
- struct AVPacketDeleter
- struct buffer_data
- struct CameraCalibrationStruct
- struct CodecParamsStruct
- struct ExtendedAzureConfig
- struct FrameStruct
- class IDecoder
- class IEncoder
- class ImageDecoder
- class ImageReader
- class iPhoneReader
- class iPhoneReaderImpl
- class IReader
- class KinectReader
- class LibAvDecoder
- class LibAvEncoder
- class MultiImageReader
- class NetworkReader
- class NullEncoder
- class NvDecoder
- class NvEncoder
- struct NVPipeDeleter
- struct SwsContextDeleter
- class VideoFileReader
- class ZDepthDecoder
- class ZDepthEncoder

Typedefs

- typedef std::unique_ptr< AVFrame, AVFrameDeleter > **AVFrameSafeP**
- typedef std::shared_ptr< AVFrame > **AVFrameSharedP**
- typedef std::unique_ptr< AVCodecContext, AVCodecContextDeleter > **AVCodecContextSafeP**
- typedef std::unique_ptr< AVPacket, AVPacketDeleter > **AVPacketSafeP**
- typedef std::shared_ptr< AVPacket > **AVPacketSharedP**
- typedef std::unique_ptr< AVCodecParameters, AVCodecParametersDeleter > **AVCodecParametersSafeP**
- typedef std::unique_ptr< AVCodecParameters, AVCodecParametersNullDeleter > **AVCodecParametersSafePNullDelete**
- typedef std::unique_ptr< struct SwsContext, SwsContextDeleter > **SwsContextSafeP**
- typedef std::unique_ptr< AVFormatContext, AVFormatContextDeleter > **AVFormatContextSafeP**
- typedef std::unique_ptr< AVCodec, AVCodecDeleter > **AVCodecSafeP**
- typedef std::unique_ptr< AVIOContext, AVIOContextDeleter > **AVIOContextSafeP**
- typedef std::unique_ptr< NvPipe, NVPipeDeleter > **NvPipeSafeP**

Enumerations

- enum video_reader_k4a_depth_mode_t {
  VIDEO_READER_K4A_DEPTH_MODE_OFF, VIDEO_READER_K4A_DEPTH_MODE_NFOV_2X2BINNED, VIDEO_READER_K4A_DEPTH_MODE_NFOV_UNBINNED, VIDEO_READER_K4A_DEPTH_MODE_WFOV_2X2BINNED,
  VIDEO_READER_K4A_DEPTH_MODE_WFOV_UNBINNED, VIDEO_READER_K4A_DEPTH_MODE_PASSIVE_IR }
- enum video_reader_k4a_color_resolution_t {
  VIDEO_READER_K4A_COLOR_RESOLUTION_OFF, VIDEO_READER_K4A_COLOR_RESOLUTION_720P, VIDEO_READER_K4A_COLOR_RESOLUTION_1080P, VIDEO_READER_K4A_COLOR_RESOLUTION_1440P,
  VIDEO_READER_K4A_COLOR_RESOLUTION_1536P, VIDEO_READER_K4A_COLOR_RESOLUTION_2160P, VIDEO_READER_K4A_COLOR_RESOLUTION_3072P }
- enum CameraCalibrationType : short { CameraCalibrationType::CameraCalibrationTypeDefault = -1, CameraCalibrationType::CameraCalibrationTypeKinect = 0 }
- enum FrameType : short { FrameType::FrameTypeColor = 0, FrameType::FrameTypeDepth = 1, FrameType::FrameTypeIR = 2, FrameType::FrameTypeConfidence = 3 }
- enum CodecParamsType : short { CodecParamsType::CodecParamsTypeDefault = -1, CodecParamsType::CodecParamsTypeAv = 0, CodecParamsType::CodecParamsTypeNvPipe = 1, CodecParamsType::CodecParamsTypeZDepth = 2 }
- enum SSPMessageType : short { SSPMessageType::MessageTypeDefault = 0 }
- enum FrameDataType : short {
  FrameDataType::FrameDataTypeImageFrame = 0, FrameDataType::FrameDataTypeLibavPackets = 1, FrameDataType::FrameDataTypeRawRGBA = 2, FrameDataType::FrameDataTypeGRAY16LE = 3,
  FrameDataType::FrameDataTypeNvPipePacket = 4, FrameDataType::FrameDataTypeRaw32FC1 = 5, FrameDataType::FrameDataTypeYUV = 6, FrameDataType::FrameDataTypeU8C1 = 7 }
- enum SensorType : short { SensorType::SensorTypeColor = 0, SensorType::SensorTypeDepth = 1, SensorType::SensorTypeIR = 2, SensorType::SensorTypeConfidence = 3 }

Functions

- std::shared_ptr< IDecoder > IDecoderFactory (const std::string &config)
- std::shared_ptr< IEncoder > IEncoderFactory (const std::string &config)
- std::shared_ptr< IReader > IReaderFactory (const std::string &config)
- void SetupLogging (std::string &level, std::string &file)
- std::atomic_bool **exiting** (false)
- unsigned long **elapsed** (unsigned long start, unsigned long end)

- bool FrameStructToMat (FrameStruct &f, cv::Mat &img, std::unordered_map< std::string, std::shared_ptr< IDecoder >> &decoders)
- ExtendedAzureConfig BuildKinectConfigFromYAML (YAML::Node config)
- void FrameStructToK4A (std::vector< FrameStruct > &fs, k4a::capture &sensor_capture, std::unordered_↩ map< std::string, std::shared_ptr< IDecoder >> &decoders)
- double GetMSE (const Mat &I1, const Mat &I2)
- double GetPSNR (const Mat &I1, const Mat &I2, double max_value)
- Scalar GetMSSIM (const Mat &i1, const Mat &i2)
- uint64_t _CurrentTimeMs ()
- uint64_t CurrentTimeUs ()
- uint64_t CurrentTimeNs ()
- std::string RandomString (size_t length)
- void SetupLogging (YAML::Node &general_parameters)
- void AVFrameToMatYUV (AVFrameSharedP &frame, cv::Mat &image)
- void AVFrameToMatGray (AVFrameSharedP &frame, cv::Mat &image)
- AVCodecParameters ∗ getParams (FrameStruct &frame_struct)
- template<typename T >
  void **MinMaxFilter** (cv::Mat &in_mat, cv::Mat &out_mat, double min, double max)

Variables

- std::atomic_bool **exiting**

7.1.1 Detailed Description

MOETSI_RAAS

Namespace libav_types.h Types for libav support

Namespace video_utils.cc Video utilities

7.1.2 Enumeration Type Documentation

7.1.2.1 CameraCalibrationType

```
enum moetsi::ssp::CameraCalibrationType : short [strong]
```

Camera calibration type i.e. the kind of sensor calibration data present.

Enumerator

CameraCalibrationTypeDefault	Default camera calibration type.
CameraCalibrationTypeKinect	Kinect format calibration type

7.1.2.2 CodecParamsType

```
enum moetsi::ssp::CodecParamsType : short [strong]
```

Codec parameters type.

Enumerator

CodecParamsTypeDefault	Default type
CodecParamsTypeAv	Libav codec configuration
CodecParamsTypeNvPipe	NvPipe configuration
CodecParamsTypeZDepth	ZDepth compression configuration

7.1.2.3 FrameDataType

```
enum moetsi::ssp::FrameDataType : short [strong]
```

Frame data type. This is a precise binary format information.

Enumerator

FrameDataTypeImageFrame	Image frame
FrameDataTypeLibavPackets	Libav packets
FrameDataTypeRawRGBA	Raw RGBA data
FrameDataTypeGRAY16LE	GRAY16LE data
FrameDataTypeNvPipePacket	NvPipe packet
FrameDataTypeRaw32FC1	Raw 32FC1 data
FrameDataTypeYUV	YUV data
FrameDataTypeU8C1	U8C1 data

7.1.2.4 FrameType

```
enum moetsi::ssp::FrameType : short [strong]
```

Frame type: color, depth, IR as well as confidence matrices.

Enumerator

FrameTypeColor	Color/BGR frame type
FrameTypeDepth	Int16 depth type in mm
FrameTypeIR	IR sensor frame type
FrameTypeConfidence	Confidence levels

7.1.2.5 SensorType

```
enum moetsi::ssp::SensorType :  short  [strong]
```

Sensor type: color or depth

Enumerator

SensorTypeColor	Color sensor
SensorTypeDepth	Depth sensor
SensorTypeIR	IR sensor
SensorTypeConfidence	Confidence

7.1.2.6 SSPMessageType

```
enum moetsi::ssp::SSPMessageType :  short  [strong]
```

SSP Message type.

Enumerator

MessageTypeDefault	Default only

7.1.2.7 video_reader_k4a_color_resolution_t

```
enum moetsi::ssp::video_reader_k4a_color_resolution_t
```

Enumerator

VIDEO_READER_K4A_COLOR_RESOLUTION_OFF	Color camera will be turned off with this setting
VIDEO_READER_K4A_COLOR_RESOLUTION_720P	1280 ∗ 720 16:9
VIDEO_READER_K4A_COLOR_RESOLUTION_1080P	1920 ∗ 1080 16:9
VIDEO_READER_K4A_COLOR_RESOLUTION_1440P	2560 ∗ 1440 16:9
VIDEO_READER_K4A_COLOR_RESOLUTION_1536P	2048 ∗ 1536 4:3
VIDEO_READER_K4A_COLOR_RESOLUTION_2160P	3840 ∗ 2160 16:9
VIDEO_READER_K4A_COLOR_RESOLUTION_3072P	4096 ∗ 3072 4:3

7.1.2.8 video_reader_k4a_depth_mode_t

```
enum moetsi::ssp::video_reader_k4a_depth_mode_t
```

Enumerator

VIDEO_READER_K4A_DEPTH_MODE_OFF	Depth sensor will be turned off with this setting.
VIDEO_READER_K4A_DEPTH_MODE_NFOV_2↩X2BINNED	Depth captured at 320x288. Passive IR is also captured at 320x288.
VIDEO_READER_K4A_DEPTH_MODE_NFOV_U↩NBINNED	Depth captured at 640x576. Passive IR is also captured at 640x576.
VIDEO_READER_K4A_DEPTH_MODE_WFOV_2↩X2BINNED	Depth captured at 512x512. Passive IR is also captured at 512x512.
VIDEO_READER_K4A_DEPTH_MODE_WFOV_U↩NBINNED	Depth captured at 1024x1024. Passive IR is also captured at 1024x1024.
VIDEO_READER_K4A_DEPTH_MODE_PASSIVE↩_IR	Passive IR only, captured at 1024x1024.

7.1.3 Function Documentation

7.1.3.1 _CurrentTimeMs()

```
uint64_t moetsi::ssp::_CurrentTimeMs ( )
```

Get current time in ms.

Returns

ms since UTC epoch

7.1.3.2 AVFrameToMatGray()

```
void moetsi::ssp::AVFrameToMatGray (
             AVFrameSharedP & frame,
             cv::Mat & image )
```

Convert an AVFrame to grayscale image

Parameters

frame	AVFrame
image	dest opencv image

7.1.3.3 AVFrameToMatYUV()

```
void moetsi::ssp::AVFrameToMatYUV (
            AVFrameSharedP & frame,
            cv::Mat & image )
```

Convert an AVFrame to YUV image

Parameters

frame	AVFrame
image	dest opencv image

7.1.3.4 BuildKinectConfigFromYAML()

```
ExtendedAzureConfig moetsi::ssp::BuildKinectConfigFromYAML (
            YAML::Node config )
```

Build Kinect configuration from YAML configuration

Parameters

config	yaml confirguration

Returns

Azure Kinect configuration

7.1.3.5 CurrentTimeNs()

```
uint64_t moetsi::ssp::CurrentTimeNs ( )
```

Get current time in ns/nanosecconds

Returns

nsec since UTC epoch

7.1.3.6 CurrentTimeUs()

```
uint64_t moetsi::ssp::CurrentTimeUs ( )
```

Get current time in usec/microseconds.

Returns

usec since UTC epoch

7.1.3.7 FrameStructToK4A()

```
void moetsi::ssp::FrameStructToK4A (
            std::vector< FrameStruct > & f,
            k4a::capture & sensor_capture,
            std::unordered_map< std::string, std::shared_ptr< IDecoder >> & decoders )
```

Transform frame structure to K4A format Update decoder dictionary

Parameters

f	source frame structure
sensor_capture	destination "capture" structure
decoders	decoders map - updated

7.1.3.8 FrameStructToMat()

```
bool moetsi::ssp::FrameStructToMat (
            FrameStruct & f,
            cv::Mat & img,
            std::unordered_map< std::string, std::shared_ptr< IDecoder >> & decoders )
```

Convert frame struct to opencv matrix.

Parameters

f	Frame struct
img	Target opencv image
decoders	decoder dictionary

7.1.3.9 GetMSE()

```
double moetsi::ssp::GetMSE (
            const Mat & I1,
            const Mat & I2 )
```

Get Mean Square Error (distance) between images

Parameters

I1	image 1
I2	image 2

Returns

MSE between these 2 images

7.1.3.10 GetMSSIM()

```
Scalar moetsi::ssp::GetMSSIM (
            const Mat & i1,
            const Mat & i2 )
```

Get Structural Similarity between 2 images cf. for instance `http://amroamroamro.github.io/mexopencv/opencv/image_similarity_demo.html` for a simple SSIM introduction

Parameters

I1	image 1
I2	image 2

Returns

3 channel similarity measure

7.1.3.11 getParams()

```
AVCodecParameters * moetsi::ssp::getParams (
            FrameStruct & frame_struct )
```

Get AVCodec parameters from a FrameStruct

Parameters

frame_struct	frame struct

Returns

AVCodec parameters

7.1.3.12 GetPSNR()

```
double moetsi::ssp::GetPSNR (
              const Mat & I1,
              const Mat & I2,
              double max_value )
```

Get Peak Signal to Noise Ration similarity

Parameters

I1	image 1
I2	image 2
max_value	max value in the PSNR formula

Returns

PSNR image similarity

7.1.3.13 IDecoderFactory()

```
std::shared_ptr< IDecoder > moetsi::ssp::IDecoderFactory (
              const std::string & config )
```

IDecoder factory.

Parameters

config	configuration

Returns

IDecoder instance

7.1.3.14 IEncoderFactory()

```
std::shared_ptr< IEncoder > moetsi::ssp::IEncoderFactory (
              const std::string & config )
```

IEncoder factory

Parameters

config	configuration

Returns

IEncoder instance

7.1.3.15 IReaderFactory()

```
std::shared_ptr< IReader > moetsi::ssp::IReaderFactory (
             const std::string & config )
```

IReader factory

Parameters

config	configuration

Returns

an IReader instance

7.1.3.16 RandomString()

```
std::string moetsi::ssp::RandomString (
             size_t length )
```

Build a random string

7.1.3.17 SetupLogging() [1/2]

```
void moetsi::ssp::SetupLogging (
             std::string & level,
             std::string & file )
```

Setup logging

Parameters

level	logging level
file	logging file

Setup SSP logging.

Parameters

level	logging level
file	log file

7.1.3.18 SetupLogging() [2/2]

```
void moetsi::ssp::SetupLogging (
            YAML::Node & general_parameters )
```

Setup SSP logging.

Parameters

general_parameters	configuration

Chapter 8

Class Documentation

8.1 _custom_k4abt_body_t Struct Reference

Public Attributes

- int32_t **Id**
- float **pelvis_x**
- float **pelvis_y**
- float **pelvis_z**
- float **pelvis_QX**
- float **pelvis_QY**
- float **pelvis_QZ**
- float **pelvis_QW**
- BYTE **pelvis_conf**
- float **spine_navel_x**
- float **spine_navel_y**
- float **spine_navel_z**
- float **spine_navel_QX**
- float **spine_navel_QY**
- float **spine_navel_QZ**
- float **spine_navel_QW**
- BYTE **spine_navel_conf**
- float **spine_chest_x**
- float **spine_chest_y**
- float **spine_chest_z**
- float **spine_chest_QX**
- float **spine_chest_QY**
- float **spine_chest_QZ**
- float **spine_chest_QW**
- BYTE **spine_chest_conf**
- float **neck_x**
- float **neck_y**
- float **neck_z**
- float **neck_QX**
- float **neck_QY**
- float **neck_QZ**
- float **neck_QW**
- BYTE **neck_conf**

- float **clavicle_left_x**
- float **clavicle_left_y**
- float **clavicle_left_z**
- float **clavicle_left_QX**
- float **clavicle_left_QY**
- float **clavicle_left_QZ**
- float **clavicle_left_QW**
- BYTE **clavicle_left_conf**
- float **shoulder_left_x**
- float **shoulder_left_y**
- float **shoulder_left_z**
- float **shoulder_left_QX**
- float **shoulder_left_QY**
- float **shoulder_left_QZ**
- float **shoulder_left_QW**
- BYTE **shoulder_left_conf**
- float **elbow_left_x**
- float **elbow_left_y**
- float **elbow_left_z**
- float **elbow_left_QX**
- float **elbow_left_QY**
- float **elbow_left_QZ**
- float **elbow_left_QW**
- BYTE **elbow_left_conf**
- float **wrist_left_x**
- float **wrist_left_y**
- float **wrist_left_z**
- float **wrist_left_QX**
- float **wrist_left_QY**
- float **wrist_left_QZ**
- float **wrist_left_QW**
- BYTE **wrist_left_conf**
- float **hand_left_x**
- float **hand_left_y**
- float **hand_left_z**
- float **hand_left_QX**
- float **hand_left_QY**
- float **hand_left_QZ**
- float **hand_left_QW**
- BYTE **hand_left_conf**
- float **handtip_left_x**
- float **handtip_left_y**
- float **handtip_left_z**
- float **handtip_left_QX**
- float **handtip_left_QY**
- float **handtip_left_QZ**
- float **handtip_left_QW**
- BYTE **handtip_left_conf**
- float **thumb_left_x**
- float **thumb_left_y**
- float **thumb_left_z**
- float **thumb_left_QX**
- float **thumb_left_QY**
- float **thumb_left_QZ**
- float **thumb_left_QW**

- BYTE **thumb_left_conf**
- float **clavicle_right_x**
- float **clavicle_right_y**
- float **clavicle_right_z**
- float **clavicle_right_QX**
- float **clavicle_right_QY**
- float **clavicle_right_QZ**
- float **clavicle_right_QW**
- BYTE **clavicle_right_conf**
- float **shoulder_right_x**
- float **shoulder_right_y**
- float **shoulder_right_z**
- float **shoulder_right_QX**
- float **shoulder_right_QY**
- float **shoulder_right_QZ**
- float **shoulder_right_QW**
- BYTE **shoulder_right_conf**
- float **elbow_right_x**
- float **elbow_right_y**
- float **elbow_right_z**
- float **elbow_right_QX**
- float **elbow_right_QY**
- float **elbow_right_QZ**
- float **elbow_right_QW**
- BYTE **elbow_right_conf**
- float **wrist_right_x**
- float **wrist_right_y**
- float **wrist_right_z**
- float **wrist_right_QX**
- float **wrist_right_QY**
- float **wrist_right_QZ**
- float **wrist_right_QW**
- BYTE **wrist_right_conf**
- float **hand_right_x**
- float **hand_right_y**
- float **hand_right_z**
- float **hand_right_QX**
- float **hand_right_QY**
- float **hand_right_QZ**
- float **hand_right_QW**
- BYTE **hand_right_conf**
- float **handtip_right_x**
- float **handtip_right_y**
- float **handtip_right_z**
- float **handtip_right_QX**
- float **handtip_right_QY**
- float **handtip_right_QZ**
- float **handtip_right_QW**
- BYTE **handtip_right_conf**
- float **thumb_right_x**
- float **thumb_right_y**
- float **thumb_right_z**
- float **thumb_right_QX**
- float **thumb_right_QY**
- float **thumb_right_QZ**

- float **thumb_right_QW**
- BYTE **thumb_right_conf**
- float **hip_left_x**
- float **hip_left_y**
- float **hip_left_z**
- float **hip_left_QX**
- float **hip_left_QY**
- float **hip_left_QZ**
- float **hip_left_QW**
- BYTE **hip_left_conf**
- float **knee_left_x**
- float **knee_left_y**
- float **knee_left_z**
- float **knee_left_QX**
- float **knee_left_QY**
- float **knee_left_QZ**
- float **knee_left_QW**
- BYTE **knee_left_conf**
- float **ankle_left_x**
- float **ankle_left_y**
- float **ankle_left_z**
- float **ankle_left_QX**
- float **ankle_left_QY**
- float **ankle_left_QZ**
- float **ankle_left_QW**
- BYTE **ankle_left_conf**
- float **foot_left_x**
- float **foot_left_y**
- float **foot_left_z**
- float **foot_left_QX**
- float **foot_left_QY**
- float **foot_left_QZ**
- float **foot_left_QW**
- BYTE **foot_left_conf**
- float **hip_right_x**
- float **hip_right_y**
- float **hip_right_z**
- float **hip_right_QX**
- float **hip_right_QY**
- float **hip_right_QZ**
- float **hip_right_QW**
- BYTE **hip_right_conf**
- float **knee_right_x**
- float **knee_right_y**
- float **knee_right_z**
- float **knee_right_QX**
- float **knee_right_QY**
- float **knee_right_QZ**
- float **knee_right_QW**
- BYTE **knee_right_conf**
- float **ankle_right_x**
- float **ankle_right_y**
- float **ankle_right_z**
- float **ankle_right_QX**
- float **ankle_right_QY**

- float **ankle_right_QZ**
- float **ankle_right_QW**
- BYTE **ankle_right_conf**
- float **foot_right_x**
- float **foot_right_y**
- float **foot_right_z**
- float **foot_right_QX**
- float **foot_right_QY**
- float **foot_right_QZ**
- float **foot_right_QW**
- BYTE **foot_right_conf**
- float **head_x**
- float **head_y**
- float **head_z**
- float **head_QX**
- float **head_QY**
- float **head_QZ**
- float **head_QW**
- BYTE **head_conf**
- float **nose_x**
- float **nose_y**
- float **nose_z**
- float **nose_QX**
- float **nose_QY**
- float **nose_QZ**
- float **nose_QW**
- BYTE **nose_conf**
- float **eye_left_x**
- float **eye_left_y**
- float **eye_left_z**
- float **eye_left_QX**
- float **eye_left_QY**
- float **eye_left_QZ**
- float **eye_left_QW**
- BYTE **eye_left_conf**
- float **ear_left_x**
- float **ear_left_y**
- float **ear_left_z**
- float **ear_left_QX**
- float **ear_left_QY**
- float **ear_left_QZ**
- float **ear_left_QW**
- BYTE **ear_left_conf**
- float **eye_right_x**
- float **eye_right_y**
- float **eye_right_z**
- float **eye_right_QX**
- float **eye_right_QY**
- float **eye_right_QZ**
- float **eye_right_QW**
- BYTE **eye_right_conf**
- float **ear_right_x**
- float **ear_right_y**
- float **ear_right_z**
- float **ear_right_QX**

- float **ear_right_QY**
- float **ear_right_QZ**
- float **ear_right_QW**
- BYTE **ear_right_conf**

The documentation for this struct was generated from the following file:

- ssp_client_k4a.cc

8.2 moetsi::ssp::AVCodecContextDeleter Struct Reference

Public Member Functions

- void **operator()** (AVCodecContext ∗ptr) const

The documentation for this struct was generated from the following file:

- libav_types.h

8.3 moetsi::ssp::AVCodecDeleter Struct Reference

Public Member Functions

- void **operator()** (AVCodec ∗ptr) const

The documentation for this struct was generated from the following file:

- libav_types.h

8.4 moetsi::ssp::AVCodecParametersDeleter Struct Reference

Public Member Functions

- void **operator()** (AVCodecParameters ∗ptr) const

The documentation for this struct was generated from the following file:

- libav_types.h

8.5 moetsi::ssp::AVCodecParametersNullDeleter Struct Reference

Public Member Functions

- void **operator()** (AVCodecParameters ∗ptr) const

The documentation for this struct was generated from the following file:

- libav_types.h

8.6 moetsi::ssp::AVFormatContextDeleter Struct Reference

Public Member Functions

- void **operator()** (AVFormatContext ∗ptr) const

The documentation for this struct was generated from the following file:

- libav_types.h

8.7 moetsi::ssp::AVFrameDeleter Struct Reference

Public Member Functions

- void **operator()** (AVFrame ∗ptr) const

The documentation for this struct was generated from the following file:

- libav_types.h

8.8 moetsi::ssp::AVIOContextDeleter Struct Reference

Public Member Functions

- void **operator()** (AVIOContext ∗ptr) const

The documentation for this struct was generated from the following file:

- libav_types.h

8.9 moetsi::ssp::AVPacketDeleter Struct Reference

Public Member Functions

- void **operator()** (AVPacket ∗ptr) const

The documentation for this struct was generated from the following file:

- libav_types.h

8.10 BodyTracker Class Reference

Public Member Functions

- **BodyTracker** (int port)
- int **update** ()
- int **getBodyCount** () const
- int **getBodiesStruct** (k4abt_body_t ∗pBodies, int n) const
- custom_k4abt_body_t **getCustomBodiesStruct** (int n) const
- int **getBodies** (k4abt_skeleton_t ∗pSkeletons, int ∗pIds, int n) const

The documentation for this class was generated from the following file:

- ssp_client_k4a.cc

8.11 moetsi::ssp::buffer_data Struct Reference

Public Attributes

- uint8_t ∗ **ptr**
- size_t size

  size left in the buffer

The documentation for this struct was generated from the following file:

- image_decoder.cc

8.12 moetsi::ssp::CameraCalibrationStruct Struct Reference

```
#include <frame_struct.h>
```

Public Member Functions

- CameraCalibrationStruct ()
- CameraCalibrationStruct (CameraCalibrationType t, std::vector< unsigned char > d, std::vector< unsigned char > ed)
- template<class Archive > void **serialize** (Archive &ar)

Public Attributes

- CameraCalibrationType type = CameraCalibrationType::CameraCalibrationTypeDefault
- std::vector< unsigned char > data
- std::vector< unsigned char > extra_data

8.12.1 Detailed Description

Camera calibration data

8.12.2 Constructor & Destructor Documentation

8.12.2.1 CameraCalibrationStruct() [1/2]

```
moetsi::ssp::CameraCalibrationStruct::CameraCalibrationStruct ( )  [inline]
```

Default constructor

8.12.2.2 CameraCalibrationStruct() [2/2]

```
moetsi::ssp::CameraCalibrationStruct::CameraCalibrationStruct (
            CameraCalibrationType t,
            std::vector< unsigned char > d,
            std::vector< unsigned char > ed )  [inline]
```

Structure constructor

Parameters

t	camera calibration type
d	opaque data blob #1
ed	opaque data blob #2

8.12.3 Member Data Documentation

8.12.3.1 data

```
std::vector<unsigned char> moetsi::ssp::CameraCalibrationStruct::data
```

Opaque data blob #1

8.12.3.2 extra_data

```
std::vector<unsigned char> moetsi::ssp::CameraCalibrationStruct::extra_data
```

Opaque data blob #2

8.12.3.3 type

```
CameraCalibrationType moetsi::ssp::CameraCalibrationStruct::type = CameraCalibrationType::↩
CameraCalibrationTypeDefault
```

Camera calibration type

The documentation for this struct was generated from the following file:

- frame_struct.h

8.13 moetsi::ssp::CodecParamsStruct Struct Reference

```
#include <frame_struct.h>
```

Public Member Functions

- CodecParamsStruct ()
- CodecParamsStruct (CodecParamsType t, std::vector< unsigned char > d, std::vector< unsigned char > ed)
- template<class Archive >
  void **serialize** (Archive &ar)

Public Attributes

- CodecParamsType type = CodecParamsType::CodecParamsTypeDefault
- std::vector< unsigned char > data
- std::vector< unsigned char > extra_data

8.13.1 Detailed Description

Codec parameters

8.13.2 Constructor & Destructor Documentation

8.13.2.1 CodecParamsStruct() [1/2]

```
moetsi::ssp::CodecParamsStruct::CodecParamsStruct ( )  [inline]
```

Default constructor

8.13.2.2 CodecParamsStruct() [2/2]

```
moetsi::ssp::CodecParamsStruct::CodecParamsStruct (
            CodecParamsType t,
            std::vector< unsigned char > d,
            std::vector< unsigned char > ed )  [inline]
```

Structural constructor

Parameters

t	codec type
d	opaque data blob #1
ed	opaque data blob #2

8.13.3 Member Data Documentation

8.13.3.1 data

```
std::vector<unsigned char> moetsi::ssp::CodecParamsStruct::data
```

Opaque data blob #1

8.13.3.2 extra_data

```
std::vector<unsigned char> moetsi::ssp::CodecParamsStruct::extra_data
```

Opaque data blob #2

8.13.3.3 type

```
CodecParamsType moetsi::ssp::CodecParamsStruct::type = CodecParamsType::CodecParamsTypeDefault
```

Codec parameters type

The documentation for this struct was generated from the following file:

- frame_struct.h

8.14 color_point_t Struct Reference

Public Attributes

- int16_t **xyz** [3]
- uint8_t **rgb** [3]

The documentation for this struct was generated from the following file:

- ssp_client_pointcloud.cc

8.15 moetsi::ssp::ExtendedAzureConfig Struct Reference

```
#include <kinect_utils.h>
```

Public Attributes

- k4a_device_configuration_t device_config
- bool stream_color
- bool stream_depth
- bool stream_ir
- int absolute_exposure_value

8.15.1 Detailed Description

Azure Kinect configuration

8.15.2 Member Data Documentation

8.15.2.1 absolute_exposure_value

```
int moetsi::ssp::ExtendedAzureConfig::absolute_exposure_value
```

Absolute exposure value

8.15.2.2 device_config

```
k4a_device_configuration_t moetsi::ssp::ExtendedAzureConfig::device_config
```

Device configuration

8.15.2.3 stream_color

```
bool moetsi::ssp::ExtendedAzureConfig::stream_color
```

If true, stream color frames

8.15.2.4 stream_depth

```
bool moetsi::ssp::ExtendedAzureConfig::stream_depth
```

If true, stream depth frames

8.15.2.5 stream_ir

```
bool moetsi::ssp::ExtendedAzureConfig::stream_ir
```

If true, stream infrared frames

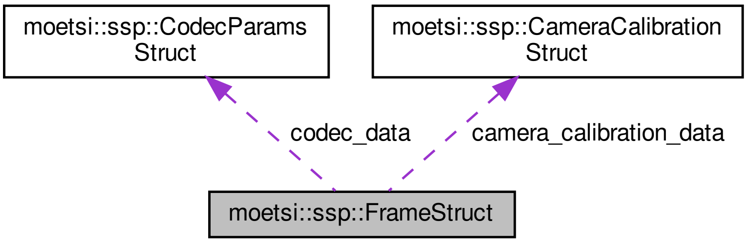
The documentation for this struct was generated from the following file:

- kinect_utils.h

8.16 moetsi::ssp::FrameStruct Struct Reference

```
#include <frame_struct.h>
```

Collaboration diagram for moetsi::ssp::FrameStruct:


moetsi::ssp::CodecParams Struct
moetsi::ssp::CameraCalibration Struct
codec_data
camera_calibration_data
moetsi::ssp::FrameStruct

Public Member Functions

- template<class Archive >
  void **serialize** (Archive &ar)

Public Attributes

- SSPMessageType message_type
- FrameType frame_type
- FrameDataType frame_data_type
- std::string stream_id
- std::vector< unsigned char > frame
- CodecParamsStruct codec_data
- CameraCalibrationStruct camera_calibration_data
- std::string scene_desc
- unsigned int sensor_id
- unsigned int device_id
- unsigned int frame_id
- std::vector< uint64_t > timestamps

8.16.1 Detailed Description

Frame struct: SSP frame.

8.16.2 Member Data Documentation

8.16.2.1 camera_calibration_data

```
CameraCalibrationStruct moetsi::ssp::FrameStruct::camera_calibration_data
```

Codec info for video frames, null for image frames

8.16.2.2 codec_data

```
CodecParamsStruct moetsi::ssp::FrameStruct::codec_data
```

Codec info for video frames, null for image frames Video decoder needs to know about the last receive frame Requires to know the codec as well as additional parameters

8.16.2.3 device_id

```
unsigned int moetsi::ssp::FrameStruct::device_id
```

Integer device id: distingish between devices in the same scene Can be set by user.

8.16.2.4 frame

```
std::vector<unsigned char> moetsi::ssp::FrameStruct::frame
```

Frame binary data We use a vector to know the size, basically a vector of bytes to store binary data

8.16.2.5 frame_data_type

```
FrameDataType moetsi::ssp::FrameStruct::frame_data_type
```

Frame data type

8.16.2.6 frame_id

```
unsigned int moetsi::ssp::FrameStruct::frame_id
```

Current frame number (increases over time) Increases by 1 for each frame automatically when SSP server starts

8.16.2.7 frame_type

```
FrameType moetsi::ssp::FrameStruct::frame_type
```

Frame type

8.16.2.8 message_type

```
SSPMessageType moetsi::ssp::FrameStruct::message_type
```

SSP message type

8.16.2.9 scene_desc

```
std::string moetsi::ssp::FrameStruct::scene_desc
```

Optional: scene description

8.16.2.10 sensor_id

```
unsigned int moetsi::ssp::FrameStruct::sensor_id
```

Sensor id

8.16.2.11 stream_id

```
std::string moetsi::ssp::FrameStruct::stream_id
```

Random 16 char string that uniquely ids the frame stream. Some decoders (like video) are stateful and so must keep track of streams. This is automatically generated.

8.16.2.12 timestamps

```
std::vector<uint64_t> moetsi::ssp::FrameStruct::timestamps
```

Use for logging and timing to understand processing speeds. Times are in ns

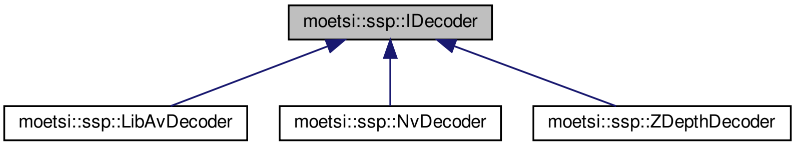
The documentation for this struct was generated from the following file:

- frame_struct.h

8.17 moetsi::ssp::IDecoder Class Reference

```
#include <idecoder.h>
```

Inheritance diagram for moetsi::ssp::IDecoder:



Public Member Functions

- virtual ∼IDecoder ()
- virtual cv::Mat Decode (FrameStruct &data)=0

8.17.1 Detailed Description

IDecoder abstract decoder interface

8.17.2 Constructor & Destructor Documentation

8.17.2.1 ∼IDecoder()

```
virtual moetsi::ssp::IDecoder::∼IDecoder ( ) [inline], [virtual]
```

Virtual destructor

8.17.3 Member Function Documentation

8.17.3.1 Decode()

```
virtual cv::Mat moetsi::ssp::IDecoder::Decode (
            FrameStruct & data )  [pure virtual]
```

Extract an opencv image from a FrameStruct data FrameStruct

Implemented in moetsi::ssp::LibAvDecoder, moetsi::ssp::NvDecoder, and moetsi::ssp::ZDepthDecoder.

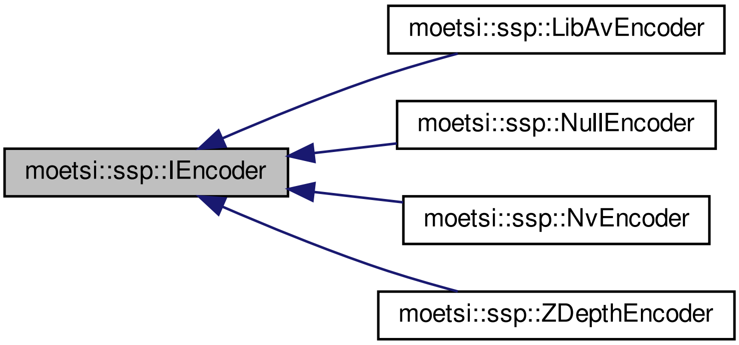
The documentation for this class was generated from the following file:

- idecoder.h

8.18 moetsi::ssp::IEncoder Class Reference

```
#include <iencoder.h>
```

Inheritance diagram for moetsi::ssp::IEncoder:



Public Member Functions

- virtual ∼IEncoder ()
- virtual void AddFrameStruct (std::shared_ptr< FrameStruct > &frame_struct)=0
- virtual void NextPacket ()=0
- virtual bool HasNextPacket ()=0
- virtual std::shared_ptr< FrameStruct > CurrentFrameEncoded ()=0
- virtual std::shared_ptr< FrameStruct > CurrentFrameOriginal ()=0
- virtual std::shared_ptr< CodecParamsStruct > GetCodecParamsStruct ()=0
- virtual unsigned int GetFps ()=0

8.18.1 Detailed Description

IEncoder abstract encoder class

8.18.2 Constructor & Destructor Documentation

8.18.2.1 ∼IEncoder()

```
virtual moetsi::ssp::IEncoder::∼IEncoder ( ) [inline], [virtual]
```

Virtual destructor

8.18.3 Member Function Documentation

8.18.3.1 AddFrameStruct()

```
virtual void moetsi::ssp::IEncoder::AddFrameStruct (
            std::shared_ptr< FrameStruct > & frame_struct ) [pure virtual]
```

Add a frame struct

Parameters

frame_struct	FrameStruct to add

Implemented in moetsi::ssp::LibAvEncoder, moetsi::ssp::NvEncoder, moetsi::ssp::ZDepthEncoder, and moetsi↵ ::ssp::NullEncoder.

8.18.3.2 CurrentFrameEncoded()

```
virtual std::shared_ptr<FrameStruct> moetsi::ssp::IEncoder::CurrentFrameEncoded ( ) [pure
virtual]
```

Get current encoded frame

Returns

current encoded frame

Implemented in moetsi::ssp::LibAvEncoder, moetsi::ssp::NvEncoder, moetsi::ssp::ZDepthEncoder, and moetsi↵ ::ssp::NullEncoder.

8.18.3.3 CurrentFrameOriginal()

```
virtual std::shared_ptr<FrameStruct> moetsi::ssp::IEncoder::CurrentFrameOriginal ( ) [pure
virtual]
```

Get current frame in its original format

Returns

current frame in its original format

Implemented in moetsi::ssp::LibAvEncoder, moetsi::ssp::NvEncoder, moetsi::ssp::ZDepthEncoder, and moetsi::ssp::NullEncoder.

8.18.3.4 GetCodecParamsStruct()

```
virtual std::shared_ptr<CodecParamsStruct> moetsi::ssp::IEncoder::GetCodecParamsStruct ( )
[pure virtual]
```

Get codec parameters

Returns

codec parameters

Implemented in moetsi::ssp::LibAvEncoder, moetsi::ssp::NvEncoder, moetsi::ssp::ZDepthEncoder, and moetsi::ssp::NullEncoder.

8.18.3.5 GetFps()

```
virtual unsigned int moetsi::ssp::IEncoder::GetFps ( ) [pure virtual]
```

Get FPS

Returns

FPS in frame per second

Implemented in moetsi::ssp::LibAvEncoder, moetsi::ssp::NvEncoder, moetsi::ssp::ZDepthEncoder, and moetsi::ssp::NullEncoder.

8.18.3.6 HasNextPacket()

```
virtual bool moetsi::ssp::IEncoder::HasNextPacket ( )  [pure virtual]
```

Check if there is a next packet

Returns

true if there is a next packet

Implemented in moetsi::ssp::LibAvEncoder, moetsi::ssp::NvEncoder, moetsi::ssp::ZDepthEncoder, and moetsi↩ ::ssp::NullEncoder.

8.18.3.7 NextPacket()

```
virtual void moetsi::ssp::IEncoder::NextPacket ( )  [pure virtual]
```

Go to next packet

Implemented in moetsi::ssp::LibAvEncoder, moetsi::ssp::NvEncoder, moetsi::ssp::ZDepthEncoder, and moetsi↩ ::ssp::NullEncoder.

The documentation for this class was generated from the following file:

- iencoder.h

8.19 moetsi::ssp::ImageDecoder Class Reference

```
#include <image_decoder.h>
```

Public Member Functions

- ImageDecoder ()
- ∼ImageDecoder ()
- void ImageBufferToAVFrame (std::shared_ptr< FrameStruct > &fs, AVFrameSharedP pFrame)

8.19.1 Detailed Description

DEcode image to AV frame

8.19.2 Constructor & Destructor Documentation

8.19.2.1 ImageDecoder()

```
moetsi::ssp::ImageDecoder::ImageDecoder ( )
```

Contructor

8.19.2.2 ∼ImageDecoder()

```
moetsi::ssp::ImageDecoder::∼ImageDecoder ( )
```

Destructor

8.19.3 Member Function Documentation

8.19.3.1 ImageBufferToAVFrame()

```
void moetsi::ssp::ImageDecoder::ImageBufferToAVFrame (
             std::shared_ptr< FrameStruct > & fs,
             AVFrameSharedP pFrame )
```

Read frame structs to AVFrame.s

Parameters

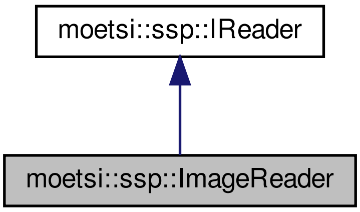
fs	frame structs
pFrame	destination AVFrame

The documentation for this class was generated from the following files:

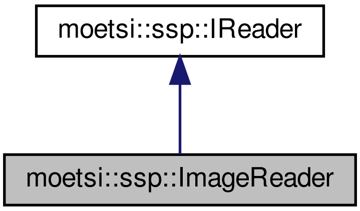
- image_decoder.h
- image_decoder.cc

8.20 moetsi::ssp::ImageReader Class Reference

Inheritance diagram for moetsi::ssp::ImageReader:



Collaboration diagram for moetsi::ssp::ImageReader:



Public Member Functions

- **ImageReader** (std::string filename)
- void Reset ()
- void GoToFrame (unsigned int frame_id)
- bool HasNextFrame ()
- void NextFrame ()
- std::vector< std::shared_ptr< FrameStruct > > GetCurrentFrame ()
- unsigned int GetCurrentFrameId ()
- std::vector< FrameType > GetType ()
- unsigned int GetFps ()

8.20.1 Member Function Documentation

8.20.1.1 GetCurrentFrame()

```
std::vector< std::shared_ptr< FrameStruct > > moetsi::ssp::ImageReader::GetCurrentFrame ( )
[virtual]
```

Get current frame data

Implements moetsi::ssp::IReader.

8.20.1.2 GetCurrentFrameId()

```
unsigned int moetsi::ssp::ImageReader::GetCurrentFrameId ( ) [virtual]
```

Get current frame number

Returns

current frame number.

Implements moetsi::ssp::IReader.

8.20.1.3 GetFps()

```
unsigned int moetsi::ssp::ImageReader::GetFps ( ) [virtual]
```

Get indicative FPS in frame per second.

Returns

the FPS number

Implements moetsi::ssp::IReader.

8.20.1.4 GetType()

```
std::vector< FrameType > moetsi::ssp::ImageReader::GetType ( ) [virtual]
```

Get frame types

Returns

a vector of FrameType, listing available data types

Implements moetsi::ssp::IReader.

8.20.1.5 GoToFrame()

```
void moetsi::ssp::ImageReader::GoToFrame (
            unsigned int frame_id ) [virtual]
```

Go to a given frame

Parameters

frame←↩ _id	target frame number

Implements moetsi::ssp::IReader.

8.20.1.6 HasNextFrame()

```
bool moetsi::ssp::ImageReader::HasNextFrame ( )  [virtual]
```

Check if there is a next frame

Returns

true if there is a next frame

Implements moetsi::ssp::IReader.

8.20.1.7 NextFrame()

```
void moetsi::ssp::ImageReader::NextFrame ( )  [virtual]
```

Go to next frame

Implements moetsi::ssp::IReader.

8.20.1.8 Reset()

```
void moetsi::ssp::ImageReader::Reset ( )  [virtual]
```

Reset this reader

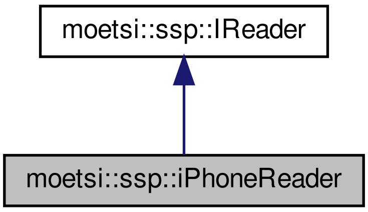
Implements moetsi::ssp::IReader.

The documentation for this class was generated from the following files:

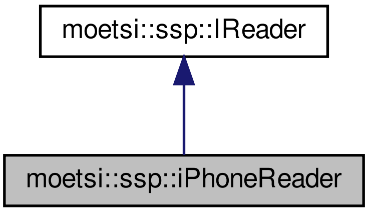
- image_reader.h
- image_reader.cc

8.21 moetsi::ssp::iPhoneReader Class Reference

Inheritance diagram for moetsi::ssp::iPhoneReader:



Collaboration diagram for moetsi::ssp::iPhoneReader:



Public Member Functions

- void Reset () override
- bool HasNextFrame () override
- void NextFrame () override
- std::vector< std::shared_ptr< FrameStruct > > GetCurrentFrame () override
- unsigned int GetCurrentFrameId () override
- void GoToFrame (unsigned int frame_id) override
- unsigned int GetFps () override
- std::vector< FrameType > GetType () override

8.21.1 Member Function Documentation

8.21.1.1 GetCurrentFrame()

```
vector< shared_ptr< FrameStruct > > moetsi::ssp::iPhoneReader::GetCurrentFrame ( ) [override],
[virtual]
```

Get current frame data

Implements moetsi::ssp::IReader.

8.21.1.2 GetCurrentFrameId()

```
unsigned int moetsi::ssp::iPhoneReader::GetCurrentFrameId ( ) [override], [virtual]
```

Get current frame number

Returns

current frame number.

Implements moetsi::ssp::IReader.

8.21.1.3 GetFps()

```
unsigned int moetsi::ssp::iPhoneReader::GetFps ( ) [override], [virtual]
```

Get indicative FPS in frame per second.

Returns

the FPS number

Implements moetsi::ssp::IReader.

8.21.1.4 GetType()

```
vector< FrameType > moetsi::ssp::iPhoneReader::GetType ( ) [override], [virtual]
```

Get frame types

Returns

a vector of FrameType, listing available data types

Implements moetsi::ssp::IReader.

8.21.1.5 GoToFrame()

```
void moetsi::ssp::iPhoneReader::GoToFrame (
            unsigned int frame_id ) [override], [virtual]
```

Go to a given frame

Parameters

frame←_id	target frame number

Implements moetsi::ssp::IReader.

8.21.1.6 HasNextFrame()

```
bool moetsi::ssp::iPhoneReader::HasNextFrame ( )  [override], [virtual]
```

Check if there is a next frame

Returns

true if there is a next frame

Implements moetsi::ssp::IReader.

8.21.1.7 NextFrame()

```
void moetsi::ssp::iPhoneReader::NextFrame ( )  [override], [virtual]
```

Go to next frame

Implements moetsi::ssp::IReader.

8.21.1.8 Reset()

```
void moetsi::ssp::iPhoneReader::Reset ( )  [override], [virtual]
```

Reset this reader

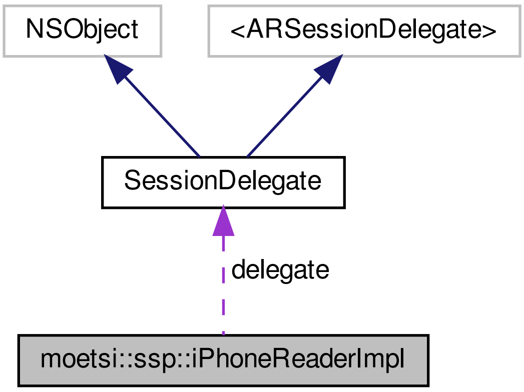
Implements moetsi::ssp::IReader.

The documentation for this class was generated from the following files:

- iphone_reader.h
- iphone_reader.mm

8.22 moetsi::ssp::iPhoneReaderImpl Class Reference

Collaboration diagram for moetsi::ssp::iPhoneReaderImpl:



Public Attributes

- ARSession ∗ **session**
- SessionDelegate ∗ **delegate**
- unsigned int **fps**
- std::shared_ptr< FrameStruct > **image**
- std::shared_ptr< FrameStruct > **depth**
- std::shared_ptr< FrameStruct > **confidence**

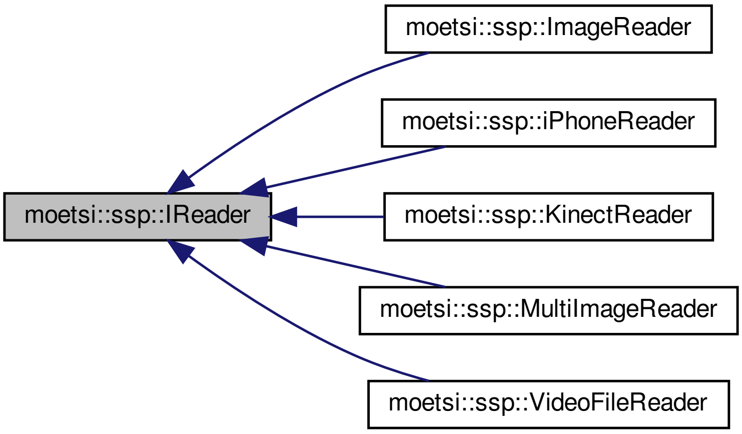
The documentation for this class was generated from the following file:

- iphone_reader.mm

8.23 moetsi::ssp::IReader Class Reference

```
#include <ireader.h>
```

Inheritance diagram for moetsi::ssp::IReader:


moetsi::ssp::ImageReader
moetsi::ssp::iPhoneReader
moetsi::ssp::IReader
moetsi::ssp::KinectReader
moetsi::ssp::MultiImageReader
moetsi::ssp::VideoFileReader

Public Member Functions

- virtual ∼IReader ()
- virtual std::vector< std::shared_ptr< FrameStruct > > GetCurrentFrame ()=0
- virtual std::vector< FrameType > GetType ()=0
- virtual bool HasNextFrame ()=0
- virtual void NextFrame ()=0
- virtual void Reset ()=0
- virtual void GoToFrame (unsigned int frame_id)=0
- virtual unsigned int GetCurrentFrameId ()=0
- virtual unsigned int GetFps ()=0

8.23.1 Detailed Description

SSP reader interface - abstract class. Question: BLOCKING operations ????

8.23.2 Constructor & Destructor Documentation

8.23.2.1 ∼IReader()

```
virtual moetsi::ssp::IReader::∼IReader ( )  [inline], [virtual]
```

Destructor

8.23.3 Member Function Documentation

8.23.3.1 GetCurrentFrame()

```
virtual std::vector<std::shared_ptr<FrameStruct> > moetsi::ssp::IReader::GetCurrentFrame ( )
[pure virtual]
```

Get current frame data

Implemented in moetsi::ssp::KinectReader, moetsi::ssp::VideoFileReader, moetsi::ssp::ImageReader, moetsi::ssp::MultiImageReader, and moetsi::ssp::iPhoneReader.

8.23.3.2 GetCurrentFrameId()

```
virtual unsigned int moetsi::ssp::IReader::GetCurrentFrameId ( )  [pure virtual]
```

Get current frame number

Returns

current frame number.

Implemented in moetsi::ssp::KinectReader, moetsi::ssp::VideoFileReader, moetsi::ssp::ImageReader, moetsi::ssp::MultiImageReader, and moetsi::ssp::iPhoneReader.

8.23.3.3 GetFps()

```
virtual unsigned int moetsi::ssp::IReader::GetFps ( )  [pure virtual]
```

Get indicative FPS in frame per second.

Returns

the FPS number

Implemented in moetsi::ssp::KinectReader, moetsi::ssp::VideoFileReader, moetsi::ssp::ImageReader, moetsi::ssp::MultiImageReader, and moetsi::ssp::iPhoneReader.

8.23.3.4 GetType()

```
virtual std::vector<FrameType> moetsi::ssp::IReader::GetType ( )  [pure virtual]
```

Get frame types

Returns

a vector of FrameType, listing available data types

Implemented in moetsi::ssp::KinectReader, moetsi::ssp::VideoFileReader, moetsi::ssp::ImageReader, moetsi↩ ::ssp::MultiImageReader, and moetsi::ssp::iPhoneReader.

8.23.3.5 GoToFrame()

```
virtual void moetsi::ssp::IReader::GoToFrame (
            unsigned int frame_id )  [pure virtual]
```

Go to a given frame

Parameters

frame↩ _id	target frame number

Implemented in moetsi::ssp::KinectReader, moetsi::ssp::VideoFileReader, moetsi::ssp::ImageReader, moetsi↩ ::ssp::MultiImageReader, and moetsi::ssp::iPhoneReader.

8.23.3.6 HasNextFrame()

```
virtual bool moetsi::ssp::IReader::HasNextFrame ( )  [pure virtual]
```

Check if there is a next frame

Returns

true if there is a next frame

Implemented in moetsi::ssp::KinectReader, moetsi::ssp::VideoFileReader, moetsi::ssp::ImageReader, moetsi↩ ::ssp::MultiImageReader, and moetsi::ssp::iPhoneReader.

8.23.3.7 NextFrame()

```
virtual void moetsi::ssp::IReader::NextFrame ( )  [pure virtual]
```

Go to next frame

Implemented in moetsi::ssp::KinectReader, moetsi::ssp::VideoFileReader, moetsi::ssp::ImageReader, moetsi↩ ::ssp::MultiImageReader, and moetsi::ssp::iPhoneReader.

8.23.3.8 Reset()

```
virtual void moetsi::ssp::IReader::Reset ( )  [pure virtual]
```

Reset this reader

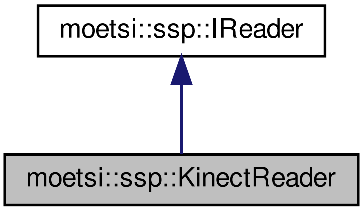
Implemented in moetsi::ssp::KinectReader, moetsi::ssp::VideoFileReader, moetsi::ssp::ImageReader, moetsi↩ ::ssp::MultiImageReader, and moetsi::ssp::iPhoneReader.

The documentation for this class was generated from the following file:

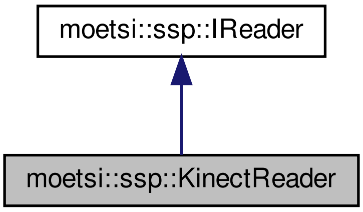
- ireader.h

8.24 moetsi::ssp::KinectReader Class Reference

Inheritance diagram for moetsi::ssp::KinectReader:



Collaboration diagram for moetsi::ssp::KinectReader:



Public Member Functions

- **KinectReader** (uint8_t device_index, ExtendedAzureConfig device_config)
- void Reset ()
- bool HasNextFrame ()
- void NextFrame ()
- std::vector< std::shared_ptr< FrameStruct > > GetCurrentFrame ()
- unsigned int GetCurrentFrameId ()
- virtual void GoToFrame (unsigned int frame_id)
- unsigned int GetFps ()
- std::vector< FrameType > GetType ()

8.24.1 Member Function Documentation

8.24.1.1 GetCurrentFrame()

```
std::vector< std::shared_ptr< FrameStruct > > moetsi::ssp::KinectReader::GetCurrentFrame ( )
[virtual]
```

Get current frame data

Implements moetsi::ssp::IReader.

8.24.1.2 GetCurrentFrameId()

```
unsigned int moetsi::ssp::KinectReader::GetCurrentFrameId ( )  [virtual]
```

Get current frame number

Returns

current frame number.

Implements moetsi::ssp::IReader.

8.24.1.3 GetFps()

```
unsigned int moetsi::ssp::KinectReader::GetFps ( )  [virtual]
```

Get indicative FPS in frame per second.

Returns

the FPS number

Implements moetsi::ssp::IReader.

8.24.1.4 GetType()

```
std::vector< FrameType > moetsi::ssp::KinectReader::GetType ( ) [virtual]
```

Get frame types

Returns

a vector of FrameType, listing available data types

Implements moetsi::ssp::IReader.

8.24.1.5 GoToFrame()

```
void moetsi::ssp::KinectReader::GoToFrame (
            unsigned int frame_id ) [virtual]
```

Go to a given frame

Parameters

frame← _id	target frame number

Implements moetsi::ssp::IReader.

8.24.1.6 HasNextFrame()

```
bool moetsi::ssp::KinectReader::HasNextFrame ( ) [virtual]
```

Check if there is a next frame

Returns

true if there is a next frame

Implements moetsi::ssp::IReader.

8.24.1.7 NextFrame()

```
void moetsi::ssp::KinectReader::NextFrame ( ) [virtual]
```

Go to next frame

Implements moetsi::ssp::IReader.

8.24.1.8 Reset()

```
void moetsi::ssp::KinectReader::Reset ( )  [virtual]
```

Reset this reader

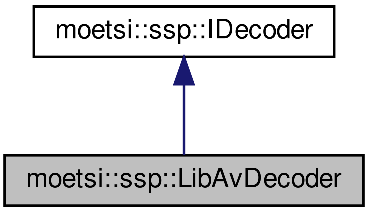
Implements moetsi::ssp::IReader.

The documentation for this class was generated from the following files:

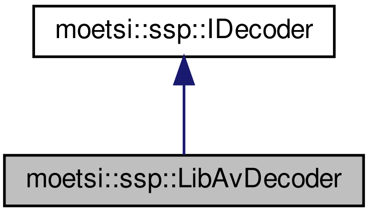
- kinect_reader.h
- kinect_reader.cc

8.25 moetsi::ssp::LibAvDecoder Class Reference

Inheritance diagram for moetsi::ssp::LibAvDecoder:



Collaboration diagram for moetsi::ssp::LibAvDecoder:



Public Member Functions

- void **Init** (AVCodecParameters ∗codec_parameters)
- cv::Mat Decode (FrameStruct &frame_struct)
- AVFrameSharedP **DecodeFrame** (FrameStruct &frame_struct)

8.25.1 Member Function Documentation

8.25.1.1 Decode()

```
cv::Mat moetsi::ssp::LibAvDecoder::Decode (
            FrameStruct & data )  [virtual]
```

Extract an opencv image from a FrameStruct data FrameStruct

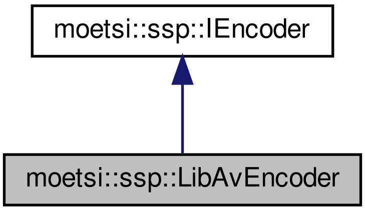
Implements moetsi::ssp::IDecoder.

The documentation for this class was generated from the following files:

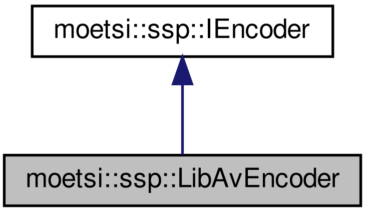
- libav_decoder.h
- libav_decoder.cc

8.26 moetsi::ssp::LibAvEncoder Class Reference

Inheritance diagram for moetsi::ssp::LibAvEncoder:



Collaboration diagram for moetsi::ssp::LibAvEncoder:



Public Member Functions

- **LibAvEncoder** (std::string codec_parameters_file, unsigned int fps)
- **LibAvEncoder** (YAML::Node &_codec_parameters, unsigned int fps)
- void AddFrameStruct (std::shared_ptr< FrameStruct > &fs)
- void NextPacket ()
- bool HasNextPacket ()
- std::shared_ptr< FrameStruct > CurrentFrameEncoded ()
- std::shared_ptr< FrameStruct > CurrentFrameOriginal ()
- std::shared_ptr< CodecParamsStruct > GetCodecParamsStruct ()
- unsigned int GetFps ()

8.26.1 Member Function Documentation

8.26.1.1 AddFrameStruct()

```
void moetsi::ssp::LibAvEncoder::AddFrameStruct (
            std::shared_ptr< FrameStruct > & frame_struct )  [virtual]
```

Add a frame struct

Parameters

frame_struct	FrameStruct to add

Implements moetsi::ssp::IEncoder.

8.26.1.2 CurrentFrameEncoded()

```
std::shared_ptr< FrameStruct > moetsi::ssp::LibAvEncoder::CurrentFrameEncoded ( )  [virtual]
```

Get current encoded frame

Returns

current encoded frame

Implements moetsi::ssp::IEncoder.

8.26.1.3 CurrentFrameOriginal()

```
std::shared_ptr< FrameStruct > moetsi::ssp::LibAvEncoder::CurrentFrameOriginal ( )  [virtual]
```

Get current frame in its original format

Returns

current frame in its original format

Implements moetsi::ssp::IEncoder.

8.26.1.4 GetCodecParamsStruct()

```
std::shared_ptr< CodecParamsStruct > moetsi::ssp::LibAvEncoder::GetCodecParamsStruct ( )
[virtual]
```

Get codec parameters

Returns

codec parameters

Implements moetsi::ssp::IEncoder.

8.26.1.5 GetFps()

```
unsigned int moetsi::ssp::LibAvEncoder::GetFps ( )  [virtual]
```

Get FPS

Returns

FPS in frame per second

Implements moetsi::ssp::IEncoder.

8.26.1.6 HasNextPacket()

```
bool moetsi::ssp::LibAvEncoder::HasNextPacket ( )  [virtual]
```

Check if there is a next packet

Returns

true if there is a next packet

Implements moetsi::ssp::IEncoder.

8.26.1.7 NextPacket()

```
void moetsi::ssp::LibAvEncoder::NextPacket ( )  [virtual]
```

Go to next packet

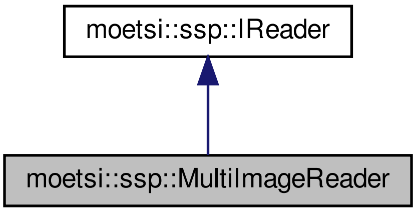
Implements moetsi::ssp::IEncoder.

The documentation for this class was generated from the following files:

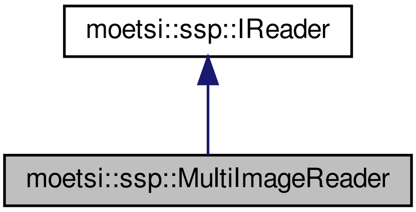
- libav_encoder.h
- libav_encoder.cc

8.27 moetsi::ssp::MultiImageReader Class Reference

Inheritance diagram for moetsi::ssp::MultiImageReader:



Collaboration diagram for moetsi::ssp::MultiImageReader:



Public Member Functions

- **MultiImageReader** (std::vector< std::string > filename)
- void Reset ()
- void GoToFrame (unsigned int frame_id)
- bool HasNextFrame ()
- void NextFrame ()
- std::vector< std::shared_ptr< FrameStruct > > GetCurrentFrame ()
- unsigned int GetCurrentFrameId ()
- std::vector< FrameType > GetType ()
- unsigned int GetFps ()

8.27.1 Member Function Documentation

8.27.1.1 GetCurrentFrame()

```
std::vector< std::shared_ptr< FrameStruct > > moetsi::ssp::MultiImageReader::GetCurrentFrame
( ) [virtual]
```

Get current frame data

Implements moetsi::ssp::IReader.

8.27.1.2 GetCurrentFrameId()

```
unsigned int moetsi::ssp::MultiImageReader::GetCurrentFrameId ( ) [virtual]
```

Get current frame number

Returns

current frame number.

Implements moetsi::ssp::IReader.

8.27.1.3 GetFps()

```
unsigned int moetsi::ssp::MultiImageReader::GetFps ( ) [virtual]
```

Get indicative FPS in frame per second.

Returns

the FPS number

Implements moetsi::ssp::IReader.

8.27.1.4 GetType()

```
std::vector< FrameType > moetsi::ssp::MultiImageReader::GetType ( )  [virtual]
```

Get frame types

Returns

a vector of FrameType, listing available data types

Implements moetsi::ssp::IReader.

8.27.1.5 GoToFrame()

```
void moetsi::ssp::MultiImageReader::GoToFrame (
             unsigned int frame_id )  [virtual]
```

Go to a given frame

Parameters

frame←_id	target frame number

Implements moetsi::ssp::IReader.

8.27.1.6 HasNextFrame()

```
bool moetsi::ssp::MultiImageReader::HasNextFrame ( )  [virtual]
```

Check if there is a next frame

Returns

true if there is a next frame

Implements moetsi::ssp::IReader.

8.27.1.7 NextFrame()

```
void moetsi::ssp::MultiImageReader::NextFrame ( )  [virtual]
```

Go to next frame

Implements moetsi::ssp::IReader.

8.27.1.8 Reset()

```
void moetsi::ssp::MultiImageReader::Reset ( )  [virtual]
```

Reset this reader

Implements moetsi::ssp::IReader.

The documentation for this class was generated from the following files:

- multi_image_reader.h
- multi_image_reader.cc

8.28 moetsi::ssp::NetworkReader Class Reference

Public Member Functions

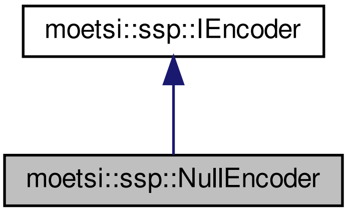
- **NetworkReader** (int port)
- void **init** ()
- bool **HasNextFrame** ()
- void **NextFrame** ()
- std::vector< FrameStruct > **GetCurrentFrame** ()
- unsigned int **GetCurrentFrameId** ()

The documentation for this class was generated from the following files:

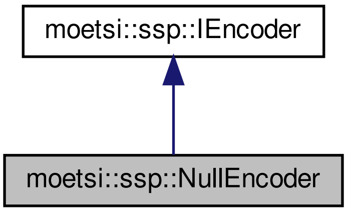
- network_reader.h
- network_reader.cc

8.29 moetsi::ssp::NullEncoder Class Reference

Inheritance diagram for moetsi::ssp::NullEncoder:



Collaboration diagram for moetsi::ssp::NullEncoder:



Public Member Functions

- **NullEncoder** (int _fps)
- void AddFrameStruct (std::shared_ptr< FrameStruct > &fs)
- void NextPacket ()
- bool HasNextPacket ()
- std::shared_ptr< FrameStruct > CurrentFrameEncoded ()
- std::shared_ptr< FrameStruct > CurrentFrameOriginal ()
- std::shared_ptr< CodecParamsStruct > GetCodecParamsStruct ()
- unsigned int GetFps ()

8.29.1 Member Function Documentation

8.29.1.1 AddFrameStruct()

```
void moetsi::ssp::NullEncoder::AddFrameStruct (
            std::shared_ptr< FrameStruct > & frame_struct )  [virtual]
```

Add a frame struct

Parameters

frame_struct	FrameStruct to add

Implements moetsi::ssp::IEncoder.

8.29.1.2 CurrentFrameEncoded()

```
std::shared_ptr< FrameStruct > moetsi::ssp::NullEncoder::CurrentFrameEncoded ( )  [virtual]
```

Get current encoded frame

Returns

current encoded frame

Implements moetsi::ssp::IEncoder.

8.29.1.3 CurrentFrameOriginal()

```
std::shared_ptr< FrameStruct > moetsi::ssp::NullEncoder::CurrentFrameOriginal ( )  [virtual]
```

Get current frame in its original format

Returns

current frame in its original format

Implements moetsi::ssp::IEncoder.

8.29.1.4 GetCodecParamsStruct()

```
std::shared_ptr< CodecParamsStruct > moetsi::ssp::NullEncoder::GetCodecParamsStruct ( )  [virtual]
```

Get codec parameters

Returns

codec parameters

Implements moetsi::ssp::IEncoder.

8.29.1.5 GetFps()

```
unsigned int moetsi::ssp::NullEncoder::GetFps ( )  [virtual]
```

Get FPS

Returns

FPS in frame per second

Implements moetsi::ssp::IEncoder.

8.29.1.6 HasNextPacket()

```
bool moetsi::ssp::NullEncoder::HasNextPacket ( )  [virtual]
```

Check if there is a next packet

Returns

true if there is a next packet

Implements moetsi::ssp::IEncoder.

8.29.1.7 NextPacket()

```
void moetsi::ssp::NullEncoder::NextPacket ( )  [virtual]
```

Go to next packet

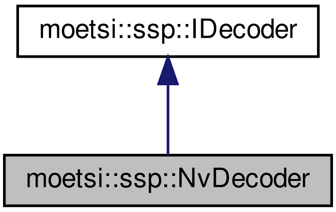
Implements moetsi::ssp::IEncoder.

The documentation for this class was generated from the following files:

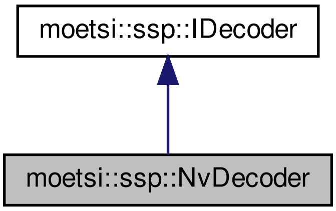
- null_encoder.h
- null_encoder.cc

8.30 moetsi::ssp::NvDecoder Class Reference

Inheritance diagram for moetsi::ssp::NvDecoder:



Collaboration diagram for moetsi::ssp::NvDecoder:



Public Member Functions

- void **Init** (std::vector< unsigned char > parameter_data)
- cv::Mat Decode (FrameStruct &frame)

8.30.1 Member Function Documentation

8.30.1.1 Decode()

```
cv::Mat moetsi::ssp::NvDecoder::Decode (
            FrameStruct & data )  [virtual]
```

Extract an opencv image from a FrameStruct data FrameStruct

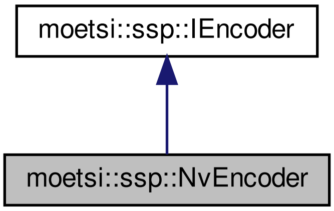
Implements moetsi::ssp::IDecoder.

The documentation for this class was generated from the following files:

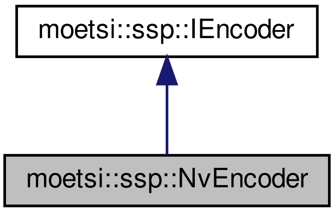
- nv_decoder.h
- nv_decoder.cc

8.31 moetsi::ssp::NvEncoder Class Reference

Inheritance diagram for moetsi::ssp::NvEncoder:



Collaboration diagram for moetsi::ssp::NvEncoder:



Public Member Functions

- **NvEncoder** (YAML::Node _codec_parameters, unsigned int _fps)
- void AddFrameStruct (std::shared_ptr< FrameStruct > &fs)
- void NextPacket ()
- bool HasNextPacket ()
- std::shared_ptr< FrameStruct > CurrentFrameEncoded ()
- std::shared_ptr< FrameStruct > CurrentFrameOriginal ()
- std::shared_ptr< CodecParamsStruct > GetCodecParamsStruct ()
- unsigned int GetFps ()

8.31.1 Member Function Documentation

8.31.1.1 AddFrameStruct()

```
void moetsi::ssp::NvEncoder::AddFrameStruct (
            std::shared_ptr< FrameStruct > & frame_struct )  [virtual]
```

Add a frame struct

Parameters

frame_struct	FrameStruct to add

Implements moetsi::ssp::IEncoder.

8.31.1.2 CurrentFrameEncoded()

```
std::shared_ptr< FrameStruct > moetsi::ssp::NvEncoder::CurrentFrameEncoded ( ) [virtual]
```

Get current encoded frame

Returns

current encoded frame

Implements moetsi::ssp::IEncoder.

8.31.1.3 CurrentFrameOriginal()

```
std::shared_ptr< FrameStruct > moetsi::ssp::NvEncoder::CurrentFrameOriginal ( ) [virtual]
```

Get current frame in its original format

Returns

current frame in its original format

Implements moetsi::ssp::IEncoder.

8.31.1.4 GetCodecParamsStruct()

```
std::shared_ptr< CodecParamsStruct > moetsi::ssp::NvEncoder::GetCodecParamsStruct ( ) [virtual]
```

Get codec parameters

Returns

codec parameters

Implements moetsi::ssp::IEncoder.

8.31.1.5 GetFps()

```
unsigned int moetsi::ssp::NvEncoder::GetFps ( )  [virtual]
```

Get FPS

Returns

FPS in frame per second

Implements moetsi::ssp::IEncoder.

8.31.1.6 HasNextPacket()

```
bool moetsi::ssp::NvEncoder::HasNextPacket ( )  [virtual]
```

Check if there is a next packet

Returns

true if there is a next packet

Implements moetsi::ssp::IEncoder.

8.31.1.7 NextPacket()

```
void moetsi::ssp::NvEncoder::NextPacket ( )  [virtual]
```

Go to next packet

Implements moetsi::ssp::IEncoder.

The documentation for this class was generated from the following files:

- nv_encoder.h
- nv_encoder.cc

8.32 moetsi::ssp::NVPipeDeleter Struct Reference

Public Member Functions

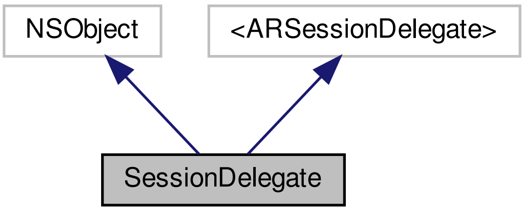
- void **operator()** (NvPipe ∗ptr) const

The documentation for this struct was generated from the following file:

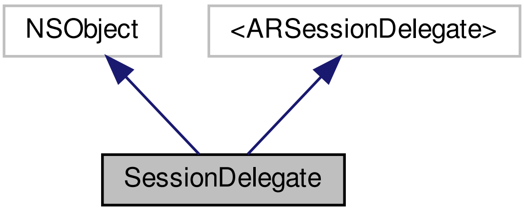
- nvpipe_types.h

8.33 SessionDelegate Class Reference

Inheritance diagram for SessionDelegate:



Collaboration diagram for SessionDelegate:



Public Attributes

- semaphore_t **_semaphore**
- pthread_mutex_t **_mutex**
- CVPixelBufferRef **_pixelBuffer**
- CVPixelBufferRef **_depthBuffer**
- CVPixelBufferRef **_confidenceBuffer**
- unsigned long **_timestamp**

The documentation for this class was generated from the following file:

- iphone_reader.mm

8.34 moetsi::ssp::SwsContextDeleter Struct Reference

Public Member Functions

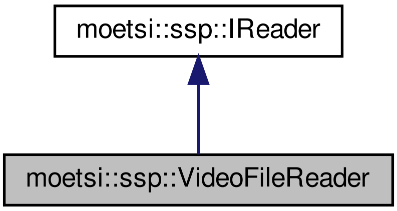
- void **operator()** (SwsContext ∗ptr) const

The documentation for this struct was generated from the following file:

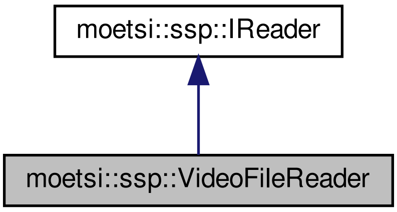
- libav_types.h

8.35 moetsi::ssp::VideoFileReader Class Reference

Inheritance diagram for moetsi::ssp::VideoFileReader:



Collaboration diagram for moetsi::ssp::VideoFileReader:



Public Member Functions

- **VideoFileReader** (std::string &filename)
- **VideoFileReader** (std::string &filename, std::vector< unsigned int > &video_stream_indexes)
- void Reset ()
- void GoToFrame (unsigned int frame_id)
- bool HasNextFrame ()
- void NextFrame ()
- std::vector< FrameType > GetType ()
- std::vector< std::shared_ptr< FrameStruct > > GetCurrentFrame ()
- unsigned int GetCurrentFrameId ()
- unsigned int GetFps ()

8.35.1 Member Function Documentation

8.35.1.1 GetCurrentFrame()

```
std::vector< std::shared_ptr< FrameStruct > > moetsi::ssp::VideoFileReader::GetCurrentFrame (
) [virtual]
```

Get current frame data

Implements moetsi::ssp::IReader.

8.35.1.2 GetCurrentFrameId()

```
unsigned int moetsi::ssp::VideoFileReader::GetCurrentFrameId ( ) [virtual]
```

Get current frame number

Returns

current frame number.

Implements moetsi::ssp::IReader.

8.35.1.3 GetFps()

```
unsigned int moetsi::ssp::VideoFileReader::GetFps ( ) [virtual]
```

Get indicative FPS in frame per second.

Returns

the FPS number

Implements moetsi::ssp::IReader.

8.35.1.4 GetType()

```
std::vector< FrameType > moetsi::ssp::VideoFileReader::GetType ( )  [virtual]
```

Get frame types

Returns

a vector of FrameType, listing available data types

Implements moetsi::ssp::IReader.

8.35.1.5 GoToFrame()

```
void moetsi::ssp::VideoFileReader::GoToFrame (
            unsigned int frame_id )  [virtual]
```

Go to a given frame

Parameters

frame← _id	target frame number

Implements moetsi::ssp::IReader.

8.35.1.6 HasNextFrame()

```
bool moetsi::ssp::VideoFileReader::HasNextFrame ( )  [virtual]
```

Check if there is a next frame

Returns

true if there is a next frame

Implements moetsi::ssp::IReader.

8.35.1.7 NextFrame()

```
void moetsi::ssp::VideoFileReader::NextFrame ( )  [virtual]
```

Go to next frame

Implements moetsi::ssp::IReader.

8.35.1.8 Reset()

```
void moetsi::ssp::VideoFileReader::Reset ( )  [virtual]
```

Reset this reader

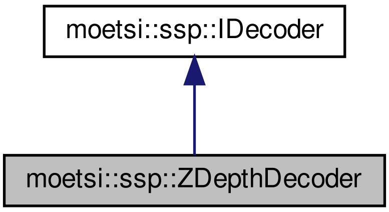
Implements moetsi::ssp::IReader.

The documentation for this class was generated from the following files:

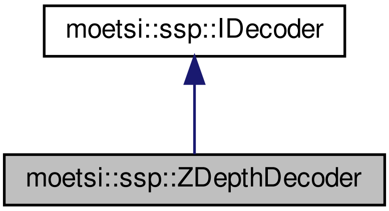
- video_file_reader.h
- video_file_reader.cc

8.36 moetsi::ssp::ZDepthDecoder Class Reference

Inheritance diagram for moetsi::ssp::ZDepthDecoder:



Collaboration diagram for moetsi::ssp::ZDepthDecoder:



Public Member Functions

- void **Init** (std::vector< unsigned char > parameter_data)
- cv::Mat Decode (FrameStruct &frame)

8.36.1 Member Function Documentation

8.36.1.1 Decode()

```
cv::Mat moetsi::ssp::ZDepthDecoder::Decode (
            FrameStruct & data )  [virtual]
```

Extract an opencv image from a FrameStruct data FrameStruct

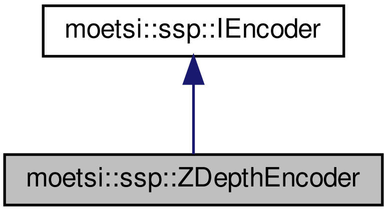
Implements moetsi::ssp::IDecoder.

The documentation for this class was generated from the following files:

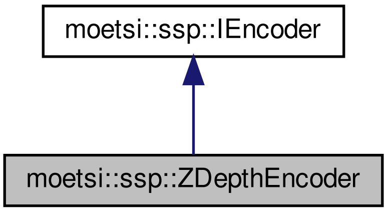
- zdepth_decoder.h
- zdepth_decoder.cc

8.37 moetsi::ssp::ZDepthEncoder Class Reference

Inheritance diagram for moetsi::ssp::ZDepthEncoder:



Collaboration diagram for moetsi::ssp::ZDepthEncoder:



Public Member Functions

- **ZDepthEncoder** (YAML::Node &_codec_parameters, int _fps)
- void AddFrameStruct (std::shared_ptr< FrameStruct > &fs)
- void NextPacket ()
- bool HasNextPacket ()
- std::shared_ptr< FrameStruct > CurrentFrameEncoded ()
- std::shared_ptr< FrameStruct > CurrentFrameOriginal ()
- std::shared_ptr< CodecParamsStruct > GetCodecParamsStruct ()
- unsigned int GetFps ()

8.37.1 Member Function Documentation

8.37.1.1 AddFrameStruct()

```
void moetsi::ssp::ZDepthEncoder::AddFrameStruct (
            std::shared_ptr< FrameStruct > & frame_struct )  [virtual]
```

Add a frame struct

Parameters

frame_struct	FrameStruct to add

Implements moetsi::ssp::IEncoder.

8.37.1.2 CurrentFrameEncoded()

```
std::shared_ptr< FrameStruct > moetsi::ssp::ZDepthEncoder::CurrentFrameEncoded ( )  [virtual]
```

Get current encoded frame

Returns

current encoded frame

Implements moetsi::ssp::IEncoder.

8.37.1.3 CurrentFrameOriginal()

```
std::shared_ptr< FrameStruct > moetsi::ssp::ZDepthEncoder::CurrentFrameOriginal ( )  [virtual]
```

Get current frame in its original format

Returns

current frame in its original format

Implements moetsi::ssp::IEncoder.

8.37.1.4 GetCodecParamsStruct()

```
std::shared_ptr< CodecParamsStruct > moetsi::ssp::ZDepthEncoder::GetCodecParamsStruct ( )
[virtual]
```

Get codec parameters

Returns

codec parameters

Implements moetsi::ssp::IEncoder.

8.37.1.5 GetFps()

```
unsigned int moetsi::ssp::ZDepthEncoder::GetFps ( ) [virtual]
```

Get FPS

Returns

FPS in frame per second

Implements moetsi::ssp::IEncoder.

8.37.1.6 HasNextPacket()

```
bool moetsi::ssp::ZDepthEncoder::HasNextPacket ( ) [virtual]
```

Check if there is a next packet

Returns

true if there is a next packet

Implements moetsi::ssp::IEncoder.

8.37.1.7 NextPacket()

```
void moetsi::ssp::ZDepthEncoder::NextPacket ( ) [virtual]
```

Go to next packet

Implements moetsi::ssp::IEncoder.

The documentation for this class was generated from the following files:

- zdepth_encoder.h
- zdepth_encoder.cc

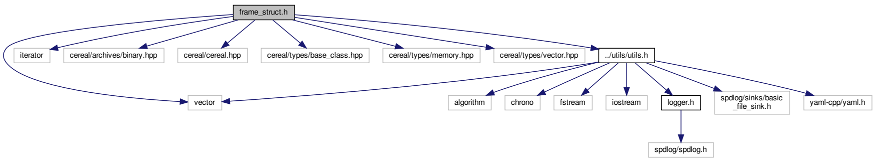
Chapter 9

File Documentation

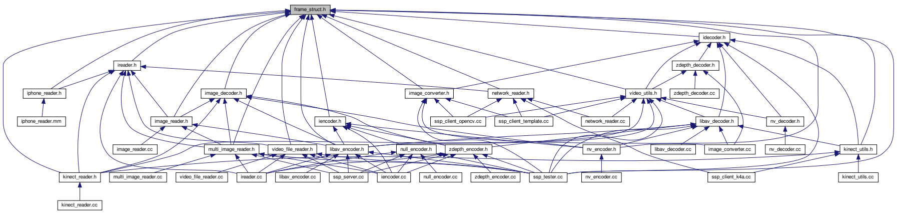
9.1 frame_struct.h File Reference

```
#include <vector>
#include <iterator>
#include <cereal/archives/binary.hpp>
#include <cereal/cereal.hpp>
#include <cereal/types/base_class.hpp>
#include <cereal/types/memory.hpp>
#include <cereal/types/vector.hpp>
#include "../utils/utils.h"
```

Include dependency graph for frame_struct.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct moetsi::ssp::CameraCalibrationStruct
- struct moetsi::ssp::CodecParamsStruct
- struct moetsi::ssp::FrameStruct

Namespaces

- moetsi::ssp

  MOETSI_RAAS

Enumerations

- enum moetsi::ssp::CameraCalibrationType : short { moetsi::ssp::CameraCalibrationType::CameraCalibrationTypeDefault = -1, moetsi::ssp::CameraCalibrationType::CameraCalibrationTypeKinect = 0 }
- enum moetsi::ssp::FrameType : short { moetsi::ssp::FrameType::FrameTypeColor = 0, moetsi::ssp::FrameType::FrameTypeDepth = 1, moetsi::ssp::FrameType::FrameTypeIR = 2, moetsi::ssp::FrameType::FrameTypeConfidence = 3 }
- enum moetsi::ssp::CodecParamsType : short { moetsi::ssp::CodecParamsType::CodecParamsTypeDefault = -1, moetsi::ssp::CodecParamsType::CodecParamsTypeAv = 0, moetsi::ssp::CodecParamsType::CodecParamsTypeNvPipe = 1, moetsi::ssp::CodecParamsType::CodecParamsTypeZDepth = 2 }
- enum moetsi::ssp::SSPMessageType : short { moetsi::ssp::SSPMessageType::MessageTypeDefault = 0 }
- enum moetsi::ssp::FrameDataType : short {
  moetsi::ssp::FrameDataType::FrameDataTypeImageFrame = 0, moetsi::ssp::FrameDataType::FrameDataTypeLibavPackets = 1, moetsi::ssp::FrameDataType::FrameDataTypeRawRGBA = 2, moetsi::ssp::FrameDataType::FrameDataTypeGRAY16LE = 3,
  moetsi::ssp::FrameDataType::FrameDataTypeNvPipePacket = 4, moetsi::ssp::FrameDataType::FrameDataTypeRaw32FC1 = 5, moetsi::ssp::FrameDataType::FrameDataTypeYUV = 6, moetsi::ssp::FrameDataType::FrameDataTypeU8C1 = 7 }
- enum moetsi::ssp::SensorType : short { moetsi::ssp::SensorType::SensorTypeColor = 0, moetsi::ssp::SensorType::SensorTypeDepth = 1, moetsi::ssp::SensorType::SensorTypeIR = 2, moetsi::ssp::SensorType::SensorTypeConfidence = 3 }

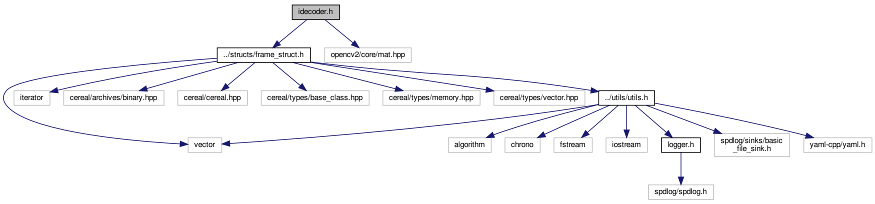
9.1.1 Detailed Description

Frame struct definition. "Universal" frame data type.

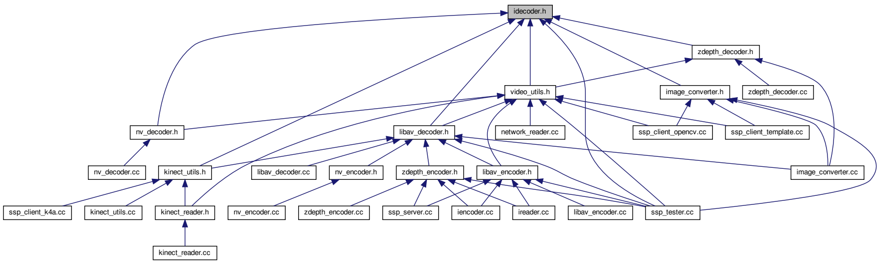
9.2 idecoder.h File Reference

```
#include "../structs/frame_struct.h"
#include <opencv2/core/mat.hpp>
```

Include dependency graph for idecoder.h:



This graph shows which files directly or indirectly include this file:



Classes

- class moetsi::ssp::IDecoder

Namespaces

- moetsi::ssp

  MOETSI_RAAS

Functions

- std::shared_ptr< IDecoder > moetsi::ssp::IDecoderFactory (const std::string &config)

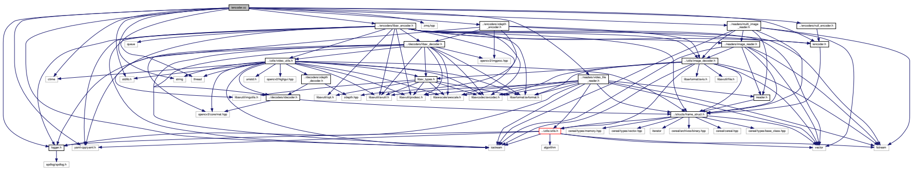
9.2.1 Detailed Description

Frame decoder interface

9.3 iencoder.cc File Reference

```
#include "iencoder.h"
#include "../utils/logger.h"
#include <ctime>
#include <iostream>
#include <stdlib.h>
#include <string>
#include <thread>
#include <yaml-cpp/yaml.h>
#include <zmq.hpp>
#include "../encoders/libav_encoder.h"
#include "../encoders/null_encoder.h"
#include "../encoders/zdepth_encoder.h"
#include "../readers/video_file_reader.h"
#include "../readers/multi_image_reader.h"
```

Include dependency graph for iencoder.cc:



Namespaces

- moetsi::ssp

    MOETSI_RAAS

Functions

- std::shared_ptr< IEncoder > moetsi::ssp::IEncoderFactory (const std::string &config)

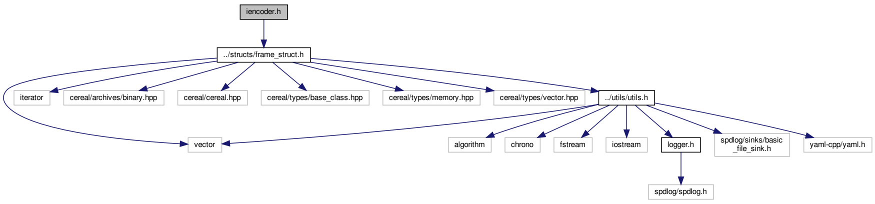
9.3.1 Detailed Description

IEncoder factory

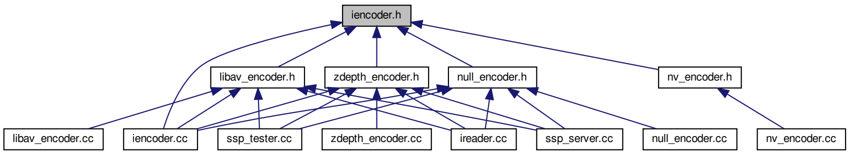
9.4 iencoder.h File Reference

```
#include "../structs/frame_struct.h"
```

Include dependency graph for iencoder.h:



This graph shows which files directly or indirectly include this file:



Classes

- class moetsi::ssp::IEncoder

Namespaces

- moetsi::ssp

    MOETSI_RAAS

Functions

- std::shared_ptr< IEncoder > moetsi::ssp::IEncoderFactory (const std::string &config)

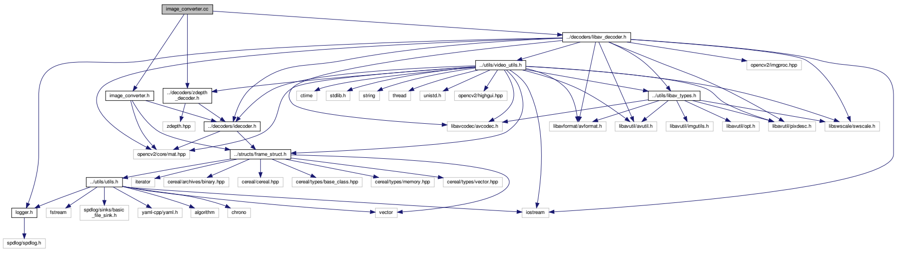
9.4.1 Detailed Description

IEncoder definition: frame encoder

9.5 image_converter.cc File Reference

```
#include "image_converter.h"
#include "../decoders/libav_decoder.h"
#include "../decoders/zdepth_decoder.h"
```

Include dependency graph for image_converter.cc:



Namespaces

- moetsi::ssp

  MOETSI_RAAS

Functions

- bool moetsi::ssp::FrameStructToMat (FrameStruct &f, cv::Mat &img, std::unordered_map< std::string, std↩ ::shared_ptr< IDecoder >> &decoders)

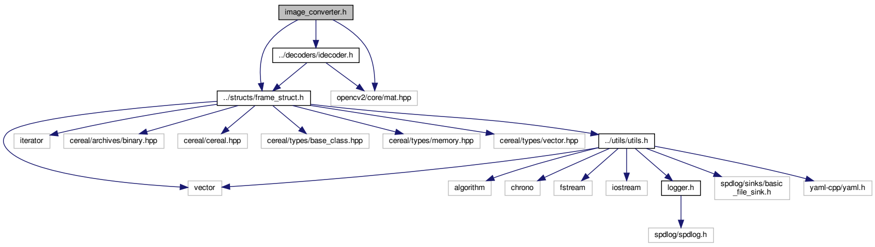
9.5.1 Detailed Description

Image converter from frame struct to opencv image

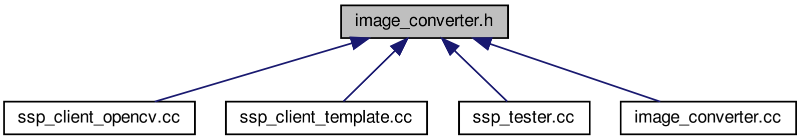
9.6 image_converter.h File Reference

```
#include "../decoders/idecoder.h"
#include "../structs/frame_struct.h"
#include <opencv2/core/mat.hpp>
```

Include dependency graph for image_converter.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- moetsi::ssp

  MOETSI_RAAS

Functions

- bool moetsi::ssp::FrameStructToMat (FrameStruct &f, cv::Mat &img, std::unordered_map< std::string, std::shared_ptr< IDecoder >> &decoders)

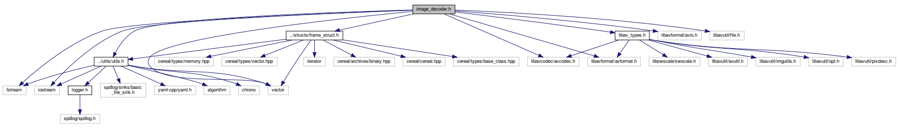
9.6.1 Detailed Description

Image converter from frame struct to opencv

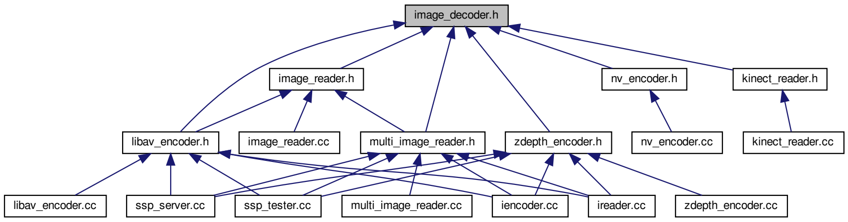
9.7 image_decoder.h File Reference

```
#include <fstream>
#include <iostream>
#include <vector>
#include <libavcodec/avcodec.h>
#include <libavformat/avformat.h>
#include <libavformat/avio.h>
#include <libavutil/file.h>
#include "../structs/frame_struct.h"
#include "libav_types.h"
#include "utils.h"
```

Include dependency graph for image_decoder.h:



This graph shows which files directly or indirectly include this file:



Classes

- class moetsi::ssp::ImageDecoder

Namespaces

- moetsi::ssp

  MOETSI_RAAS

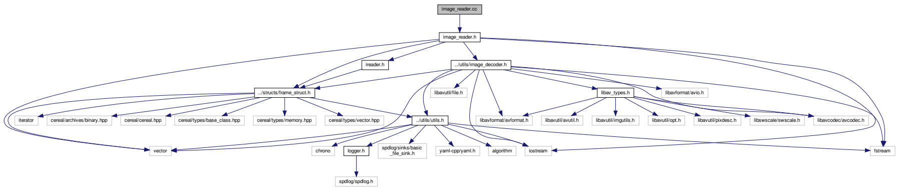
9.7.1 Detailed Description

AV Image decoder

9.8 image_reader.cc File Reference

```
#include "image_reader.h"
```

Include dependency graph for image_reader.cc:



Namespaces

- moetsi::ssp

  MOETSI_RAAS

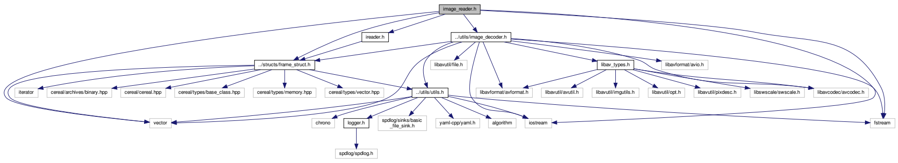
9.8.1 Detailed Description

Image reader

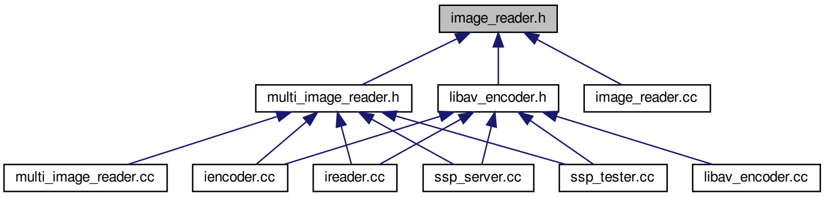
9.9 image_reader.h File Reference

```
#include <fstream>
#include <iostream>
#include <vector>
#include "../structs/frame_struct.h"
#include "../utils/image_decoder.h"
#include "ireader.h"
```

Include dependency graph for image_reader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class moetsi::ssp::ImageReader

Namespaces

- moetsi::ssp

  MOETSI_RAAS

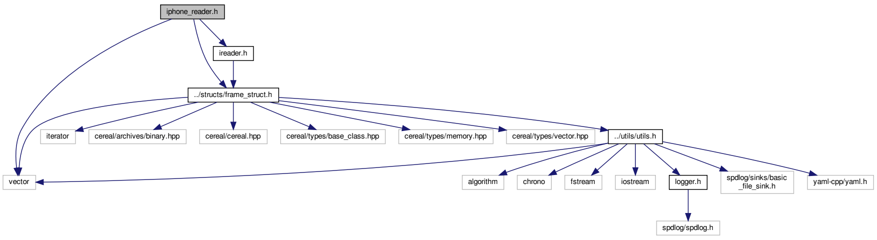
9.9.1 Detailed Description

Image reader

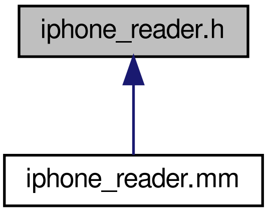
9.10 iphone_reader.h File Reference

```
#include <vector>
#include "../structs/frame_struct.h"
#include "ireader.h"
```

Include dependency graph for iphone_reader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class moetsi::ssp::iPhoneReader

Namespaces

- moetsi::ssp

  MOETSI_RAAS

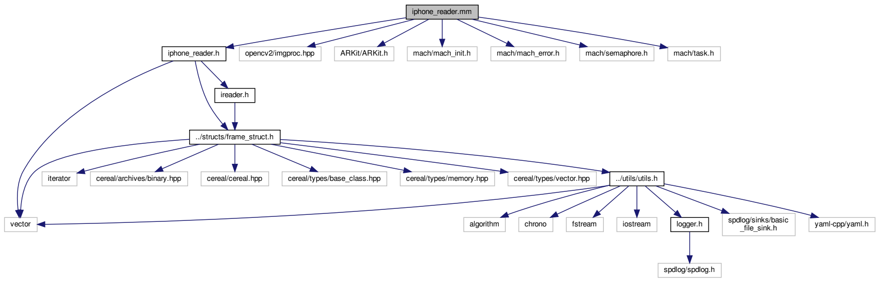
9.10.1 Detailed Description

iPhone driver

9.11 iphone_reader.mm File Reference

```
#include "iphone_reader.h"
#include <opencv2/imgproc.hpp>
#import <ARKit/ARKit.h>
#include <mach/mach_init.h>
#include <mach/mach_error.h>
#include <mach/semaphore.h>
#include <mach/task.h>
```

Include dependency graph for iphone_reader.mm:



Classes

- class SessionDelegate
- class moetsi::ssp::iPhoneReaderImpl

Namespaces

- moetsi::ssp

  MOETSI_RAAS

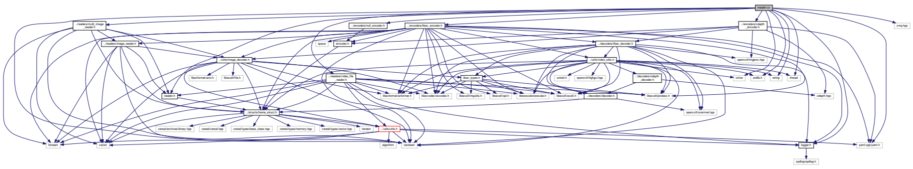
9.11.1 Detailed Description

iPhone driver

9.12 ireader.cc File Reference

```
#include "ireader.h"
#include "../utils/logger.h"
#include <ctime>
#include <iostream>
#include <stdlib.h>
#include <string>
#include <thread>
#include <yaml-cpp/yaml.h>
#include <zmq.hpp>
#include "../encoders/libav_encoder.h"
#include "../encoders/null_encoder.h"
#include "../encoders/zdepth_encoder.h"
#include "../readers/video_file_reader.h"
#include "../readers/multi_image_reader.h"
```

Include dependency graph for ireader.cc:



Namespaces

- moetsi::ssp

  MOETSI_RAAS

Functions

- std::shared_ptr< IReader > moetsi::ssp::IReaderFactory (const std::string &config)

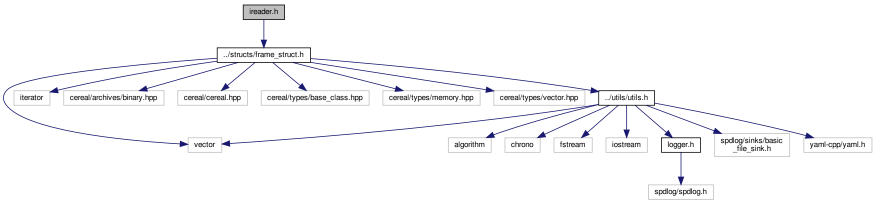
9.12.1 Detailed Description

IReader factory

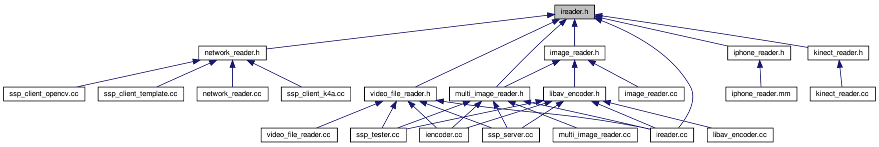
9.13 ireader.h File Reference

```
#include "../structs/frame_struct.h"
```

Include dependency graph for ireader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class moetsi::ssp::IReader

Namespaces

- moetsi::ssp

  MOETSI_RAAS

Functions

- void moetsi::ssp::SetupLogging (std::string &level, std::string &file)
- std::shared_ptr< IReader > moetsi::ssp::IReaderFactory (const std::string &config)

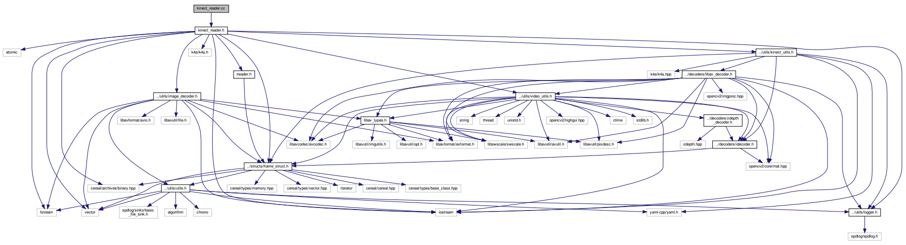
9.13.1 Detailed Description

Reader interface to SSP

9.14 kinect_reader.cc File Reference

`#include "kinect_reader.h"`

Include dependency graph for kinect_reader.cc:



Namespaces

- moetsi::ssp

  MOETSI_RAAS

Functions

- std::atomic_bool **moetsi::ssp::exiting** (false)

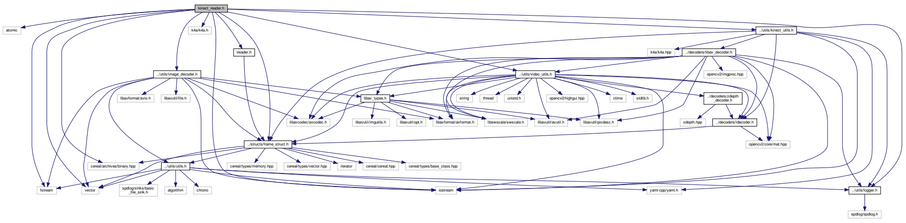
9.14.1 Detailed Description

Kinect driver

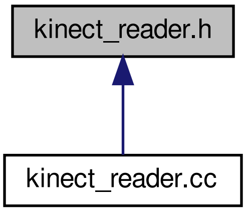
9.15 kinect_reader.h File Reference

```
#include <atomic>
#include <fstream>
#include <iostream>
#include <vector>
#include "../utils/logger.h"
#include <k4a/k4a.h>
#include <cereal/archives/binary.hpp>
#include "../structs/frame_struct.h"
#include "../utils/image_decoder.h"
#include "../utils/kinect_utils.h"
#include "../utils/video_utils.h"
#include "ireader.h"
```

Include dependency graph for kinect_reader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class moetsi::ssp::KinectReader

Namespaces

- moetsi::ssp

  MOETSI_RAAS

Macros

- #define **CHECK**(x, device)

Variables

- std::atomic_bool **moetsi::ssp::exiting**

9.15.1 Detailed Description

Kinect driver

9.15.2 Macro Definition Documentation

9.15.2.1 CHECK

```
#define CHECK(
               x,
               device )
```

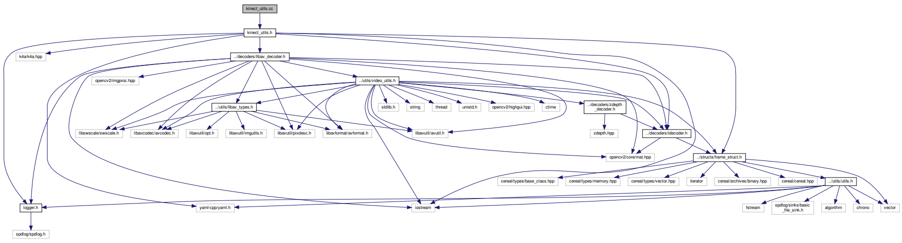
Value:

```
{                                                                              \
    auto retval = (x);                                                           \
    if (retval) {                                                                \
      spdlog::error("\"Runtime error: {} returned {} ", #x, retval);             \
      k4a_device_close(device);                                                  \
      exit(1);                                                                   \
    }                                                                            \
  }
```

9.16 kinect_utils.cc File Reference

```
#include "kinect_utils.h"
```

Include dependency graph for kinect_utils.cc:



Namespaces

- moetsi::ssp

  MOETSI_RAAS

Functions

- ExtendedAzureConfig moetsi::ssp::BuildKinectConfigFromYAML (YAML::Node config)
- void moetsi::ssp::FrameStructToK4A (std::vector< FrameStruct > &fs, k4a::capture &sensor_capture, std↩ ::unordered_map< std::string, std::shared_ptr< IDecoder >> &decoders)

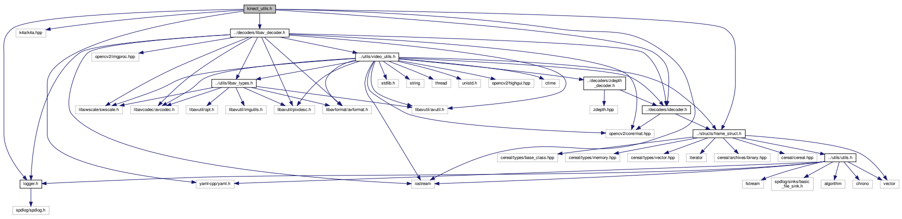
9.16.1 Detailed Description

Utils for Kinect RT integration

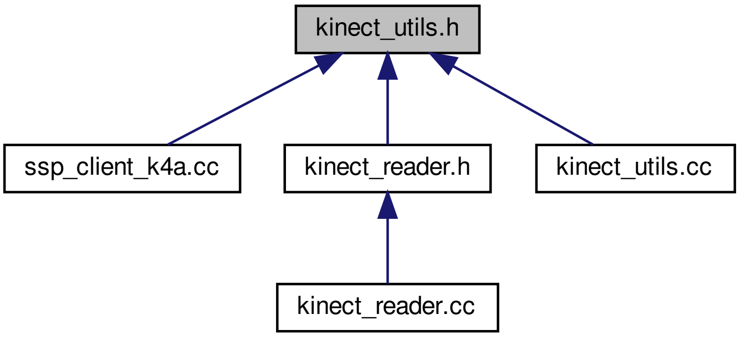
9.17 kinect_utils.h File Reference

```
#include <iostream>
#include <k4a/k4a.hpp>
#include <yaml-cpp/yaml.h>
#include "../decoders/idecoder.h"
#include "../decoders/libav_decoder.h"
#include "../structs/frame_struct.h"
#include "logger.h"
```

Include dependency graph for kinect_utils.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct moetsi::ssp::ExtendedAzureConfig

Namespaces

- moetsi::ssp

  MOETSI_RAAS

Functions

- ExtendedAzureConfig moetsi::ssp::BuildKinectConfigFromYAML (YAML::Node config)
- void moetsi::ssp::FrameStructToK4A (std::vector< FrameStruct > &fs, k4a::capture &sensor_capture, std::unordered_map< std::string, std::shared_ptr< IDecoder >> &decoders)

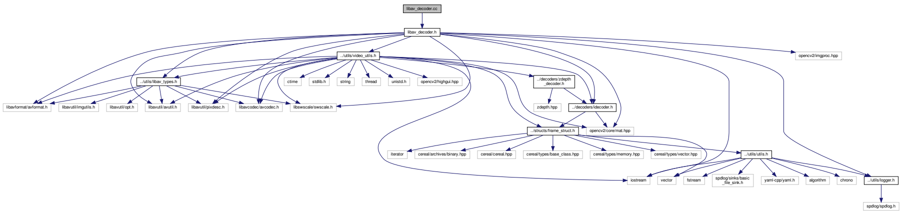
9.17.1 Detailed Description

Utils for Kinect RT integration

9.18 libav_decoder.cc File Reference

```
#include "libav_decoder.h"
```

Include dependency graph for libav_decoder.cc:



Namespaces

- moetsi::ssp

  MOETSI_RAAS

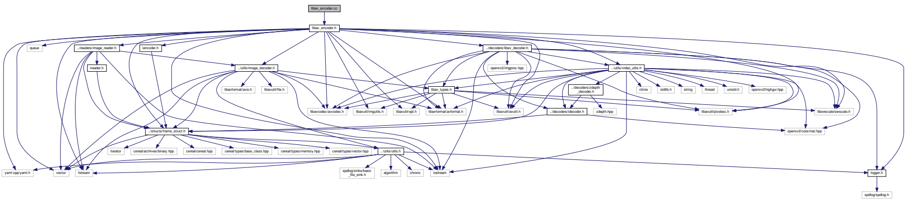
9.18.1 Detailed Description

Jpeg/Mpeg decoder

9.19 libav_encoder.cc File Reference

```
#include "libav_encoder.h"
```

Include dependency graph for libav_encoder.cc:



Namespaces

- moetsi::ssp

  MOETSI_RAAS

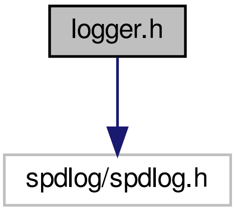
9.19.1 Detailed Description

Jpef/Mpeg encoder

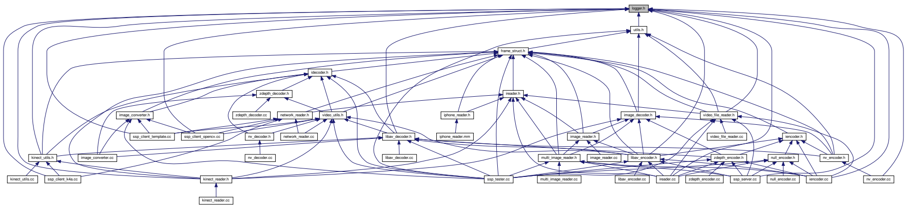
9.20 logger.h File Reference

```
#include <spdlog/spdlog.h>
```

Include dependency graph for logger.h:



This graph shows which files directly or indirectly include this file:



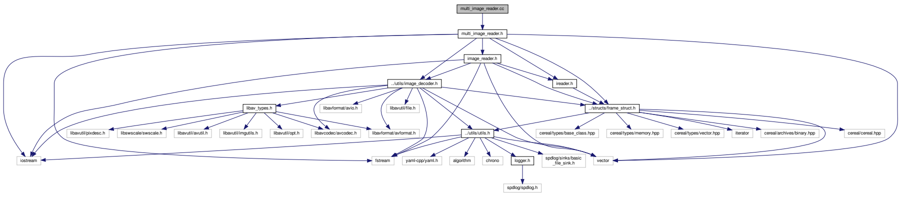
9.20.1 Detailed Description

Logger header

9.21 multi_image_reader.cc File Reference

```
#include "multi_image_reader.h"
```

Include dependency graph for multi_image_reader.cc:



Namespaces

- moetsi::ssp

  MOETSI_RAAS

9.21.1 Detailed Description

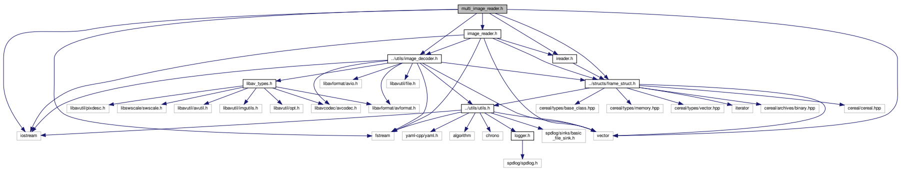
Multi image reader

9.22 multi_image_reader.h File Reference

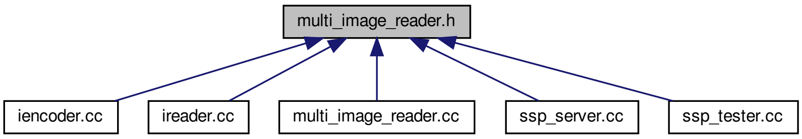
```
#include <fstream>
#include <iostream>
#include <vector>
#include "../structs/frame_struct.h"
#include "../utils/image_decoder.h"
#include "image_reader.h"
```

```
#include "ireader.h"
```

Include dependency graph for multi_image_reader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class moetsi::ssp::MultiImageReader

Namespaces

- moetsi::ssp

MOETSI_RAAS

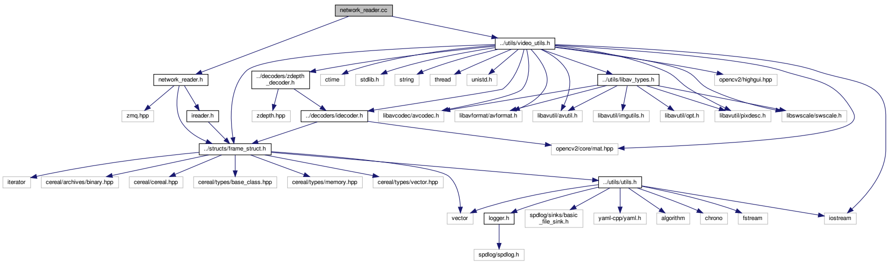
9.22.1 Detailed Description

Multi image reader

9.23 network_reader.cc File Reference

```
#include "network_reader.h"
#include "../utils/video_utils.h"
```

Include dependency graph for network_reader.cc:



Namespaces

- moetsi::ssp

  MOETSI_RAAS

Functions

- unsigned long **moetsi::ssp::elapsed** (unsigned long start, unsigned long end)

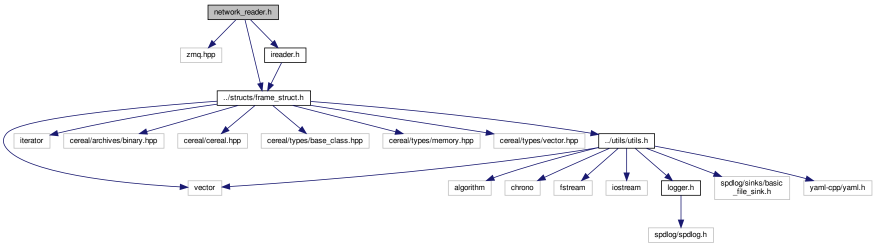
9.23.1 Detailed Description

Network reader

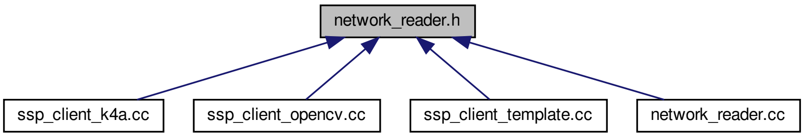
9.24 network_reader.h File Reference

```
#include <zmq.hpp>
#include "../structs/frame_struct.h"
#include "ireader.h"
```

Include dependency graph for network_reader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class moetsi::ssp::NetworkReader

Namespaces

- moetsi::ssp

    MOETSI_RAAS

Macros

- #define **POLL_TIMEOUT_MS** 500

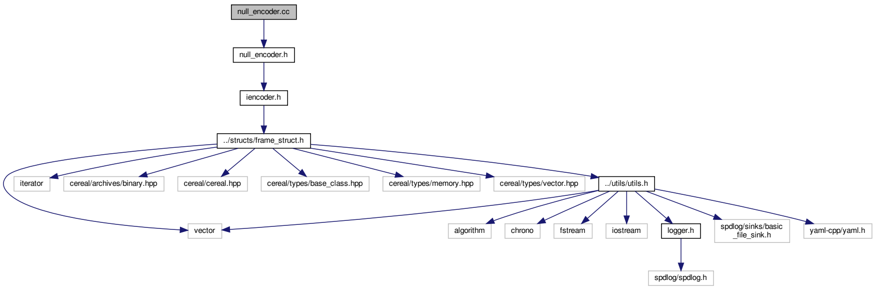
9.24.1 Detailed Description

Network reader

9.25 null_encoder.cc File Reference

```
#include "null_encoder.h"
```

Include dependency graph for null_encoder.cc:



Namespaces

- moetsi::ssp

    MOETSI_RAAS

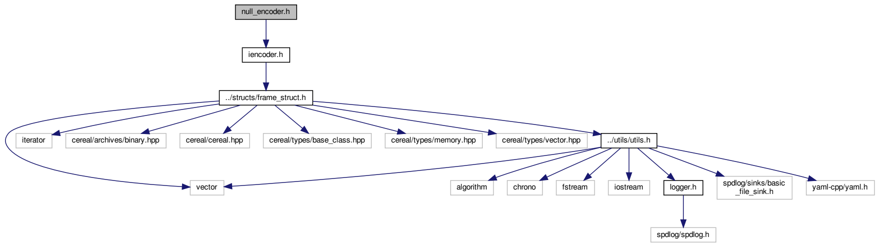
9.25.1 Detailed Description

Straight pipe encoder

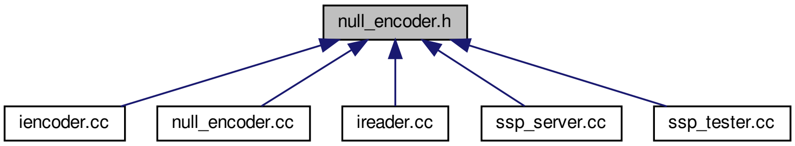
9.26 null_encoder.h File Reference

```
#include "iencoder.h"
```

Include dependency graph for null_encoder.h:



This graph shows which files directly or indirectly include this file:



Classes

- class moetsi::ssp::NullEncoder

Namespaces

- moetsi::ssp

  MOETSI_RAAS

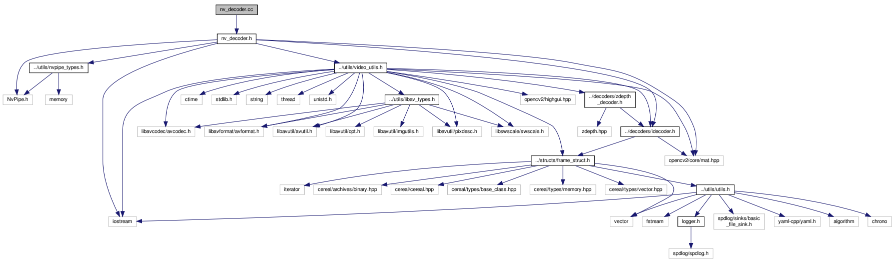
9.26.1 Detailed Description

Straight pipe encoder

9.27 nv_decoder.cc File Reference

```
#include "nv_decoder.h"
```

Include dependency graph for nv_decoder.cc:



Namespaces

- moetsi::ssp

  MOETSI_RAAS

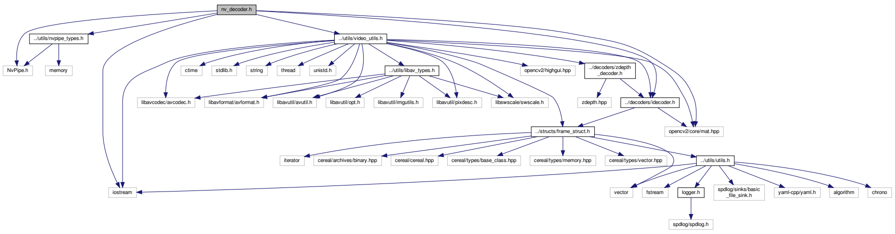
9.27.1 Detailed Description

NvPipe decoder

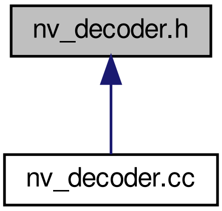
9.28 nv_decoder.h File Reference

```
#include <NvPipe.h>
#include <iostream>
#include <opencv2/core/mat.hpp>
#include "../utils/nvpipe_types.h"
#include "../utils/video_utils.h"
#include "idecoder.h"
```

Include dependency graph for nv_decoder.h:



This graph shows which files directly or indirectly include this file:



Classes

- class moetsi::ssp::NvDecoder

Namespaces

- moetsi::ssp

  MOETSI_RAAS

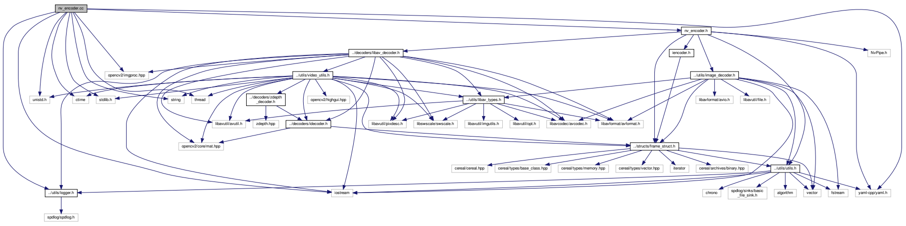
9.28.1 Detailed Description

NvPipe decoder

9.29 nv_encoder.cc File Reference

```
#include <unistd.h>
#include "../utils/logger.h"
#include <ctime>
#include <iostream>
#include <stdlib.h>
#include <string>
#include <thread>
#include <opencv2/imgproc.hpp>
#include <yaml-cpp/yaml.h>
#include "nv_encoder.h"
```

Include dependency graph for nv_encoder.cc:



Namespaces

- moetsi::ssp

    MOETSI_RAAS

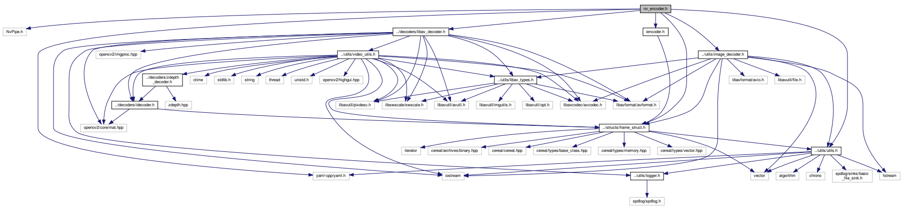
9.29.1 Detailed Description

NvPipe encoder

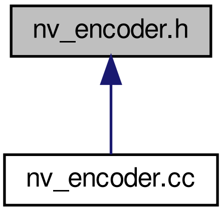
9.30 nv_encoder.h File Reference

```
#include <NvPipe.h>
#include <yaml-cpp/yaml.h>
#include "../decoders/libav_decoder.h"
#include "../utils/image_decoder.h"
#include "iencoder.h"
#include "../structs/frame_struct.h"
#include "../utils/utils.h"
```

Include dependency graph for nv_encoder.h:



This graph shows which files directly or indirectly include this file:



Classes

- class moetsi::ssp::NvEncoder

Namespaces

- moetsi::ssp

  MOETSI_RAAS

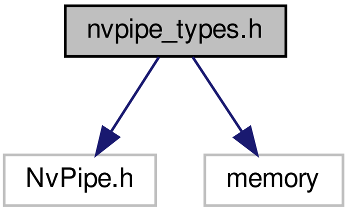
9.30.1 Detailed Description

NvPipe encoder

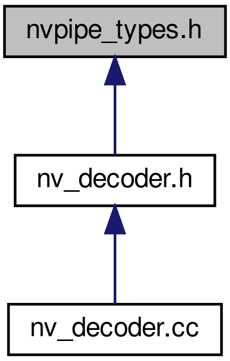
9.31 nvpipe_types.h File Reference

```
#include <NvPipe.h>
#include <memory>
```

Include dependency graph for nvpipe_types.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct moetsi::ssp::NVPipeDeleter

Namespaces

- moetsi::ssp

  MOETSI_RAAS

Typedefs

- typedef std::unique_ptr< NvPipe, NVPipeDeleter > **moetsi::ssp::NvPipeSafeP**

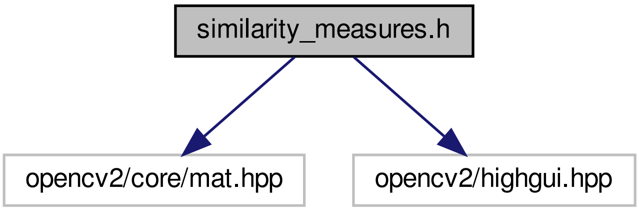
9.31.1 Detailed Description

Type for NVPipe support

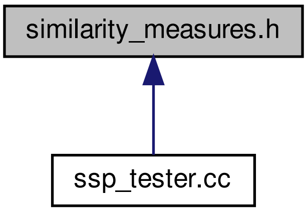
9.32 similarity_measures.h File Reference

```
#include <opencv2/core/mat.hpp>
#include <opencv2/highgui.hpp>
```

Include dependency graph for similarity_measures.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- moetsi::ssp

  MOETSI_RAAS

Functions

- double moetsi::ssp::GetPSNR (const Mat &I1, const Mat &I2, double max_value)
- double moetsi::ssp::GetMSE (const Mat &I1, const Mat &I2)
- Scalar moetsi::ssp::GetMSSIM (const Mat &i1, const Mat &i2)

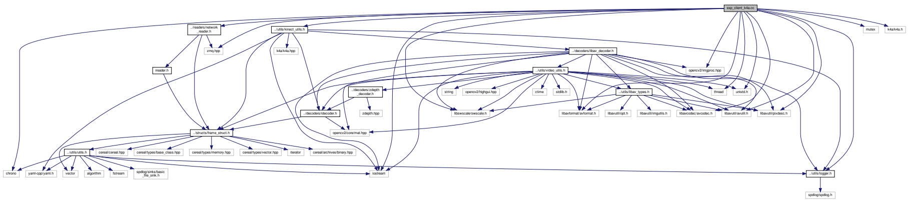
9.32.1 Detailed Description

Similarity measures

9.33 ssp_client_k4a.cc File Reference

```
#include <chrono>
#include <iostream>
#include <mutex>
#include <thread>
#include <unistd.h>
#include <k4a/k4a.h>
#include <opencv2/imgproc.hpp>
#include <zmq.hpp>
#include <libavcodec/avcodec.h>
#include <libavformat/avformat.h>
#include <libavutil/avutil.h>
#include <libavutil/pixdesc.h>
#include <libswscale/swscale.h>
#include "../utils/logger.h"
#include "../readers/network_reader.h"
#include "../utils/kinect_utils.h"
```

Include dependency graph for ssp_client_k4a.cc:



Classes

- struct _custom_k4abt_body_t
- class BodyTracker

Typedefs

- typedef struct _custom_k4abt_body_t **custom_k4abt_body_t**

Functions

- SSP_EXPORT int **open_k4a** (int port)
- SSP_EXPORT int **close_k4a** ()
- void **update** ()
- SSP_EXPORT int **start_k4a** (int port)
- SSP_EXPORT int **stop_k4a** ()
- SSP_EXPORT int **update_k4a** ()
- SSP_EXPORT int **getBodyCount** ()
- SSP_EXPORT int **getBodiesStruct** (k4abt_body_t ∗pBodies, int n)
- SSP_EXPORT custom_k4abt_body_t **getCustomBodiesStruct** (int n)
- SSP_EXPORT int **getBodies** (k4abt_skeleton_t ∗pSkeletons, int ∗pIds, int n)
- void **PrintBodyInformation** (k4abt_body_t body)
- void **PrintBodyIndexMapMiddleLine** (k4a::image body_index_map)
- int **main** (int argc, char ∗argv[ ])

Variables

- BodyTracker ∗ **gTracker** = NULL
- std::thread **gUpdateThread**
- bool **gStop** = false

9.33.1 Detailed Description

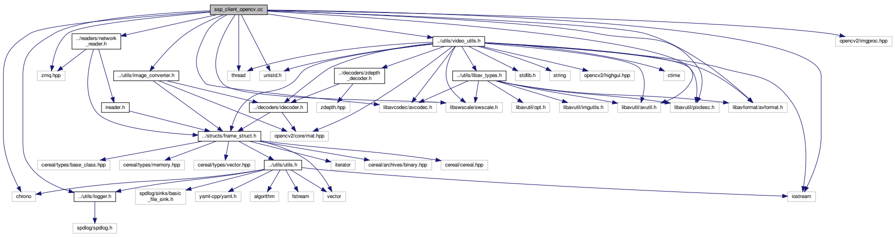
SSP client with lib k4a

9.34 ssp_client_opencv.cc File Reference

```
#include <chrono>
#include <iostream>
#include <thread>
#include <unistd.h>
#include <opencv2/imgproc.hpp>
#include <zmq.hpp>
#include <libavcodec/avcodec.h>
#include <libavformat/avformat.h>
#include <libavutil/avutil.h>
#include <libavutil/pixdesc.h>
#include <libswscale/swscale.h>
#include "../utils/logger.h"
#include "../readers/network_reader.h"
#include "../utils/video_utils.h"
```

```
#include "../utils/image_converter.h"
```

Include dependency graph for ssp_client_opencv.cc:



Macros

- #define **SSP_EXPORT**
- #define **HAS_IMSHOW** 1

Functions

- SSP_EXPORT int **ssp_client_opencv** (int port)
- int **main** (int argc, char ∗argv[ ])

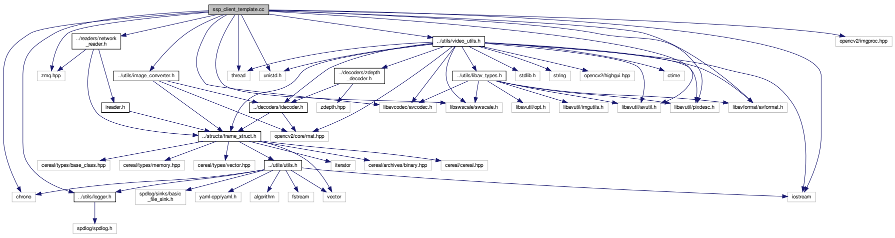
9.34.1 Detailed Description

OpenCV based ssp client client

9.35 ssp_client_template.cc File Reference

```
#include <chrono>
#include <iostream>
#include <thread>
#include <unistd.h>
#include <opencv2/imgproc.hpp>
#include <zmq.hpp>
#include <libavcodec/avcodec.h>
#include <libavformat/avformat.h>
#include <libavutil/avutil.h>
#include <libavutil/pixdesc.h>
#include <libswscale/swscale.h>
#include "../utils/logger.h"
#include "../readers/network_reader.h"
#include "../utils/video_utils.h"
#include "../utils/image_converter.h"
```

Include dependency graph for ssp_client_template.cc:



Functions

- SSP_EXPORT int **ssp_client_template** (int port)
- int **main** (int argc, char ∗argv[ ])

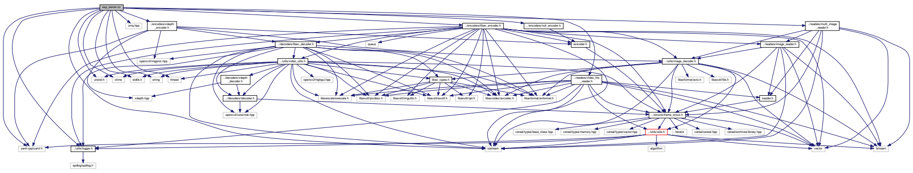
9.35.1 Detailed Description

Template for an SSP client

9.36 ssp_server.cc File Reference

```
#include <unistd.h>
#include "../utils/logger.h"
#include <ctime>
#include <iostream>
#include <stdlib.h>
#include <string>
#include <thread>
#include <yaml-cpp/yaml.h>
#include <zmq.hpp>
#include "../encoders/libav_encoder.h"
#include "../encoders/null_encoder.h"
#include "../encoders/zdepth_encoder.h"
#include "../readers/video_file_reader.h"
#include "../readers/multi_image_reader.h"
```

Include dependency graph for ssp_server.cc:



Functions

- SSP_EXPORT int **ssp_server** (const char ∗filename)
- int **main** (int argc, char ∗argv[ ])

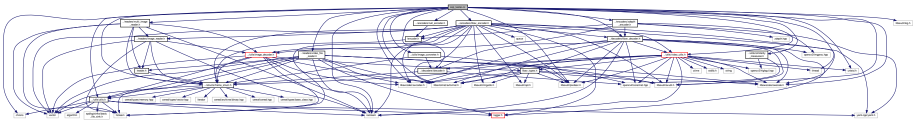
9.36.1 Detailed Description

SSP, server side.

9.37 ssp_tester.cc File Reference

```
#include <chrono>
#include <iostream>
#include <thread>
#include <unistd.h>
#include <libavcodec/avcodec.h>
#include <libavformat/avformat.h>
#include <libavutil/avutil.h>
#include <libavutil/log.h>
#include <libavutil/pixdesc.h>
#include <libswscale/swscale.h>
#include "../encoders/libav_encoder.h"
#include "../structs/frame_struct.h"
#include "../decoders/idecoder.h"
#include "../decoders/libav_decoder.h"
#include "../encoders/null_encoder.h"
#include "../encoders/zdepth_encoder.h"
#include "../readers/video_file_reader.h"
#include "../readers/multi_image_reader.h"
#include "../utils/image_converter.h"
#include "../utils/similarity_measures.h"
#include "../utils/utils.h"
#include "../utils/video_utils.h"
```

Include dependency graph for ssp_tester.cc:



Functions

- int **main** (int argc, char ∗argv[ ])

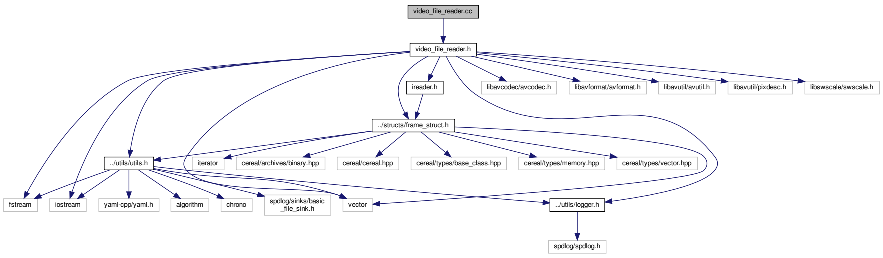
9.37.1 Detailed Description

SSP test program

9.38 video_file_reader.cc File Reference

```
#include "video_file_reader.h"
```

Include dependency graph for video_file_reader.cc:



Namespaces

- moetsi::ssp

  MOETSI_RAAS

Enumerations

- enum moetsi::ssp::video_reader_k4a_depth_mode_t {
  moetsi::ssp::VIDEO_READER_K4A_DEPTH_MODE_OFF, moetsi::ssp::VIDEO_READER_K4A_DEPTH_MODE_NFOV_2X2BINNED, moetsi::ssp::VIDEO_READER_K4A_DEPTH_MODE_NFOV_UNBINNED,
  moetsi::ssp::VIDEO_READER_K4A_DEPTH_MODE_WFOV_2X2BINNED,
  moetsi::ssp::VIDEO_READER_K4A_DEPTH_MODE_WFOV_UNBINNED, moetsi::ssp::VIDEO_READER_K4A_DEPTH_MODE_PASSIVE_IR }
- enum moetsi::ssp::video_reader_k4a_color_resolution_t {
  moetsi::ssp::VIDEO_READER_K4A_COLOR_RESOLUTION_OFF, moetsi::ssp::VIDEO_READER_K4A_COLOR_RESOLUTION_720P, moetsi::ssp::VIDEO_READER_K4A_COLOR_RESOLUTION_1080P,
  moetsi::ssp::VIDEO_READER_K4A_COLOR_RESOLUTION_1440P,
  moetsi::ssp::VIDEO_READER_K4A_COLOR_RESOLUTION_1536P, moetsi::ssp::VIDEO_READER_K4A_COLOR_RESOLUTION_2160P, moetsi::ssp::VIDEO_READER_K4A_COLOR_RESOLUTION_3072P
  }

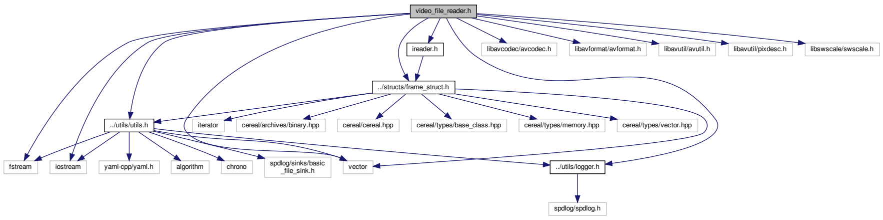
9.38.1 Detailed Description

Video file reader

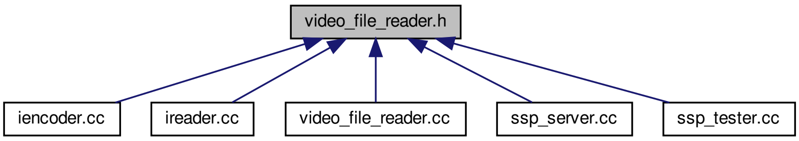
9.39 video_file_reader.h File Reference

```
#include <fstream>
#include <iostream>
#include <vector>
#include "../utils/logger.h"
#include <libavcodec/avcodec.h>
#include <libavformat/avformat.h>
#include <libavutil/avutil.h>
#include <libavutil/pixdesc.h>
#include <libswscale/swscale.h>
#include "../structs/frame_struct.h"
#include "../utils/utils.h"
#include "ireader.h"
```

Include dependency graph for video_file_reader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class moetsi::ssp::VideoFileReader

Namespaces

- moetsi::ssp

  MOETSI_RAAS

9.39.1 Detailed Description

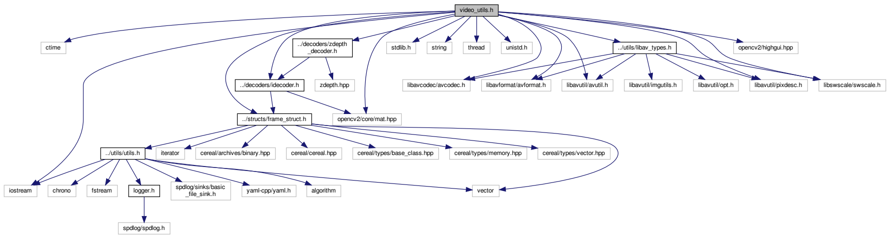
Video file reader support

9.40 video_utils.h File Reference

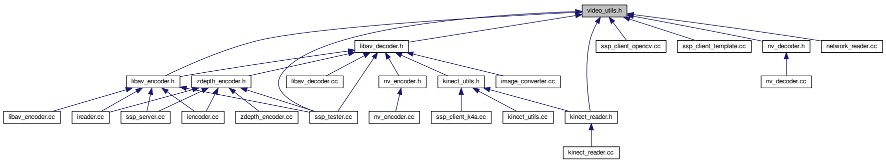
```
#include <ctime>
#include <iostream>
#include <stdlib.h>
#include <string>
#include <thread>
#include <unistd.h>
#include <libavcodec/avcodec.h>
#include <libavformat/avformat.h>
#include <libavutil/avutil.h>
#include <libavutil/pixdesc.h>
#include <libswscale/swscale.h>
#include <opencv2/core/mat.hpp>
#include <opencv2/highgui.hpp>
#include "../decoders/idecoder.h"
#include "../decoders/zdepth_decoder.h"
#include "../structs/frame_struct.h"
```

```
#include "../utils/libav_types.h"
```

Include dependency graph for video_utils.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- moetsi::ssp

  MOETSI_RAAS

Macros

- #define **MAX_DEPTH_VALUE_16_BITS** 65536
- #define **MAX_DEPTH_VALUE_14_BITS** 16384
- #define **MAX_DEPTH_VALUE_13_BITS** 8192
- #define **MAX_DEPTH_VALUE_12_BITS** 4096
- #define **MAX_DEPTH_VALUE_11_BITS** 2048
- #define **MAX_DEPTH_VALUE_8_BITS** 256

Functions

- void moetsi::ssp::AVFrameToMatYUV (AVFrameSharedP &frame, cv::Mat &image)
- void moetsi::ssp::AVFrameToMatGray (AVFrameSharedP &frame, cv::Mat &image)
- AVCodecParameters $*$ moetsi::ssp::getParams (FrameStruct &frame_struct)
- template$<$typename T $>$
  void **moetsi::ssp::MinMaxFilter** (cv::Mat &in_mat, cv::Mat &out_mat, double min, double max)

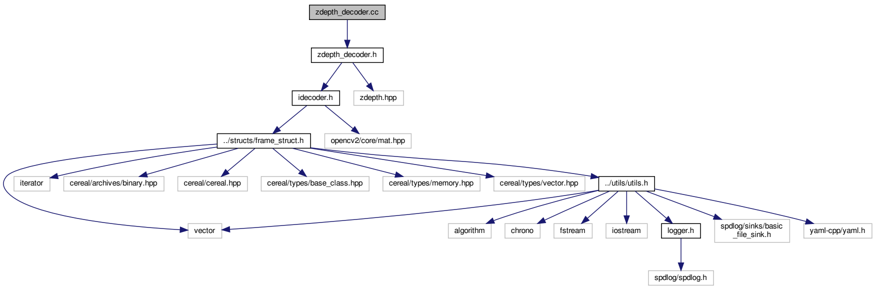
9.40.1 Detailed Description

Video utilities

9.41 zdepth_decoder.cc File Reference

```
#include "zdepth_decoder.h"
```

Include dependency graph for zdepth_decoder.cc:



Namespaces

- moetsi::ssp

  MOETSI_RAAS

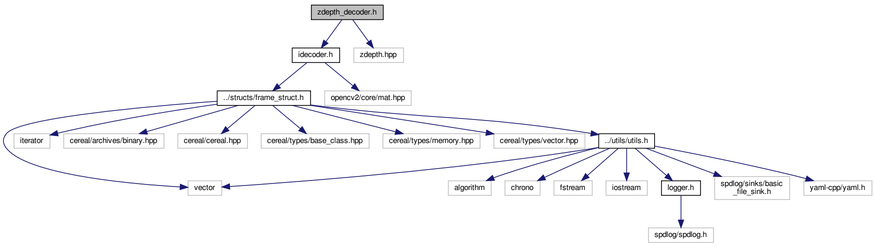
9.41.1 Detailed Description

ZDepth decoder

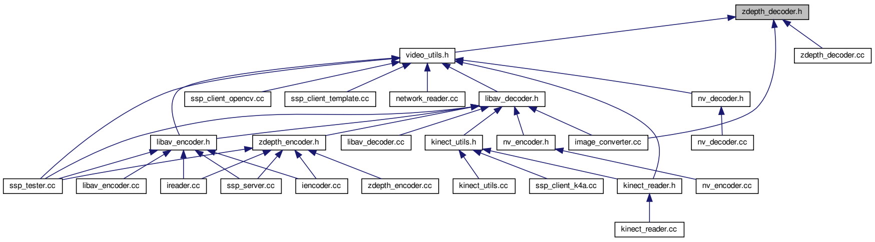
9.42 zdepth_decoder.h File Reference

```
#include "idecoder.h"
#include "zdepth.hpp"
```

Include dependency graph for zdepth_decoder.h:



This graph shows which files directly or indirectly include this file:



Classes

- class moetsi::ssp::ZDepthDecoder

Namespaces

- moetsi::ssp

    MOETSI_RAAS

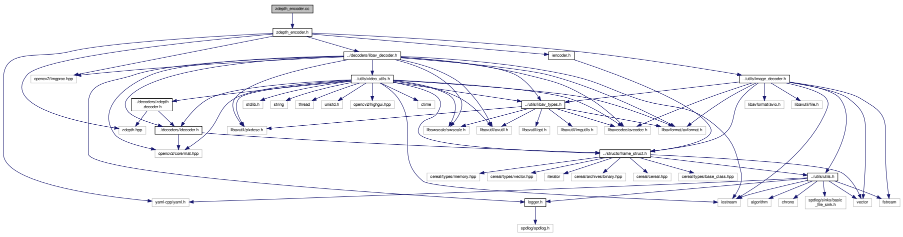
9.42.1 Detailed Description

ZDepth decoder

9.43 zdepth_encoder.cc File Reference

```
#include "zdepth_encoder.h"
```

Include dependency graph for zdepth_encoder.cc:



Namespaces

- moetsi::ssp

    MOETSI_RAAS

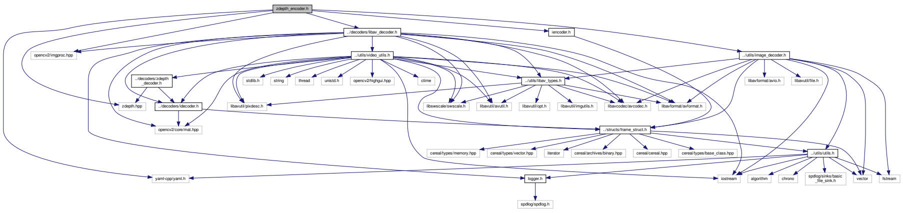
9.43.1 Detailed Description

ZDepth encoder

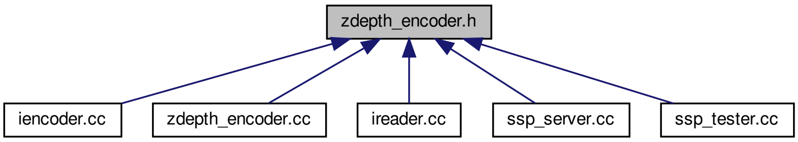
9.44 zdepth_encoder.h File Reference

```
#include "zdepth.hpp"
#include <yaml-cpp/yaml.h>
#include <opencv2/imgproc.hpp>
#include "iencoder.h"
#include "../decoders/libav_decoder.h"
#include "../utils/image_decoder.h"
```

Include dependency graph for zdepth_encoder.h:



This graph shows which files directly or indirectly include this file:



Classes

- class moetsi::ssp::ZDepthEncoder

Namespaces

- moetsi::ssp

  MOETSI_RAAS

9.44.1 Detailed Description

ZDepth encoder