SC Analyzer

Generated by Doxygen 1.8.14

Contents

Chapter 1

Namespace Index

1.1 Namespace List

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

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

Chapter 4

Namespace Documentation

4.1 dto Namespace Reference

The dto namespace contains all DTOs.

Namespaces

- Configuration

  This namepsace contains all configuration options.

Classes

- struct Camera

  The camera dto cotnains all camera properties.
- struct Image

  The image dto struct cotnains all image properties.
- struct Person

  The person dto struct cotnains all person properties.
- struct Region

  The region dto struct cotnains all region properties.
- class SQLHelper

  The SQLHelper class contains all helper functions and manages the connection state.
- struct Track

  The track dto struct cotnains all track properties.

4.1.1 Detailed Description

The dto namespace contains all DTOs.

4.2 dto::Configuration Namespace Reference

This namepsace contains all configuration options.

Variables

- const bool LOAD_AF_IMAGE = false

  Configure if ArrayFire image should be loaded.
- const bool CREATE_DISTORTED_IMAGE = false

  Configure if distorted images should be calculated.
- const bool SHOW_DISTORTED_IMAGE = false

  Configure if distorted images should be displayed.
- const bool SAVE_DISTORTED_IMAGE = false

  Configure if distorted images should be saved.
- const std::string DISTORTED_IMAGES_DIRECTORY = "C:\\Temp\\output\\"

  Save location for distorted images.
- const bool SHOW_ORIGINAL_IMAGES = false

  Configure if original images should be displayed.
- const bool SAVE_ORIGINAL_IMAGES = true

  Configure if original images should be saved.
- const std::string ORIGINAL_IMAGES_DIRECTORY = "C:\\Temp\\output\\"

  Original images save location.
- const bool SHOW_FG_IMAGES = true

  Configure if foreground images should be displayed.
- const bool SAVE_FG_IMAGES = true

  Configure if foreground images should be saved.
- const std::string FG_IMAGES_DIRECTORY = "C:\\Temp\\output\\"

  Original images save location.
- const bool SHOW_FG_MASKS = false

  Configure if foreground mask should be displayed.
- const bool SAVE_FG_MASK = false

  Configure if foreground mask should be saved.
- const std::string FG_MASKS_DIRECTORY = "C:\\Temp\\output\\"

  Foreground mask save location.
- const bool SHOW_ALL_CONTOURS = false

  Configure if all countours should be displayed.
- const bool SAVE_ALL_CONTOURS = true

  Configure if all contours should be saved.
- const std::string ALL_CONTOURS_DIRECTORY = "C:\\Temp\\output\\"

  All contours save directory.
- const bool SHOW_CONTOUR_IMAGES = false

  Configure if contour images should be displayed.
- const bool SAVE_CONTOUR_IMAGES = true

  Configure if contour images should be saved.
- const std::string CONTOUR_IMAGES_DIRECTORY = "C:\\Temp\\output\\"

  Contour images save directory.
- const bool SHOW_TRACK_IMAGES = false

  Configure if track images should be displayed.
- const bool SAVE_TRACK_IMAGES = true

  Configure if track images should be saved.
- const std::string TRACK_IMAGES_DIRECTORY = "C:\\Temp\\output\\"

  Track images save directory.
- const int MAX_NUMBER_OF_MISSING_FRAMES_IN_TRACK = 8

  Configure the maximum number of missing frames in a track.
- const int MIN_NUMBERS_OF_FRAMES_IN_TRACK = 8

Configure the minimum number of frames in a track.

- const std::string yoloConfig = "yolo.cfg.txt"

  Name of the yolo config file.
- const std::string yoloWeights = "yolo.weights"

  Name of the yolo weights file.
- const int yoloPersonObjectId = 0

  Yolo person object id.
- const bool PRINT_YOLO_PERSONS = false

  Configure if Yolo BBs should be printed on console.
- const bool SHOW_YOLO_PERSONS_IMAGES = true

  Configure if Yolo BBs should be displayed.
- const bool SAVE_YOLO_PERSONS_IMAGES = true

  Configure if Yolo BBs should be saved.
- const bool USE_HIGH_CONTRAST_IMAGE_FOR_YOLO = false

  Configure if local contrast correction should be applied on input image.
- const std::string YOLO_PERSONS_IMAGES_DIRECTORY = "C:\\Temp\\output\\"

  Yolo BB Images save directory.
- const int YOLO_GPU_ID = 0

  GPU ID for Yolo inferencing.
- const bool USE_YOLO_FOR_TRACKING = true

  Configure if Yolo images should be used for tracking.
- const bool USE_YOLO_FOR_FRAMESELECTION = true

  Configure if Yolo images should be used for frame selection.
- const bool USE_FG_IMAGE_FOR_YOLO = false

  Configure if FG images should be used as input for yolo.
- const std::string PERSON_COUNTER_0_PUBLISH_URL = "https://sc-analyzer-reporter.firebaseio.↩ com/person_counter_0.json"

  Configure Publish URL for person counter.
- const float IMAGE_BORDER_PERCENTATGE = 0.03f

  Configure percentage of border. This is used for entry side detection.
- const bool SAVE_TRACK_STATISTICS = true

  Configure if statistics file for a track should be saved.
- const std::string STATISTICS_DIRECTORY = "C:\\Temp\\output\\"

  Statistics save directory.
- const double MAX_TRACKING_DISTANCE = 50

  Configure maximum number of pixels a bounding box can move between two frames.
- const bool USE_POINT_FOR_OPTIMAL_TRACK = true

  Configure if optimal track position should use a point.
- const long MAX_OPTIMAL_DISTANCE = 50

  Configure max distance from optimal point to bounding box.
- const bool PRINT_FRAME_SELECTION_STEPS = false

  Configure if frame selection steps should be printed to console.
- const bool SAVE_OPTIMAL_TRACK_IMAGE = true

  Configure if optimal track image should be saved.
- const bool SAVE_OPTIMAL_TRACK_IMAGE_CUT = true

  Configure if cut optimal track image should be saved.
- const std::string OPTIMAL_TRACK_DIRECTORY = "C:\\Temp\\output\\"

  Optimal track image save directory.
- const bool SAVE_BODY_PARTS_IMAGES = true

  Configure if body part images should be saved.
- const bool SAVE_HUE_IMAGE = true

Configure if generated image with hue only should be saved.

- const bool PRINT_HSV_VALUES = false

  Configure if HSV values should be printed on console.

- const bool SAVE_FEATURE_POINT_IMAGES = true

  Configure if feature point images should be saved.

- const bool USE_FG_IMAGE_FOR_FEATURE_POINTS = true

  Configure if foreground image should be used for feature point extraction.

- const bool STORE_TRACK_RESULTS_IN_DB = false

  Configure if tracks should be persisted in the MSSQL database.

- const nanodbc::string DATABASE_ODBC_NAME = "sc_analyzer"

  ODBC connection name. Must be configured on the system.

- const nanodbc::string DATABASE_USER = "db_user"

  ODBC user name to connect to the database.

- const nanodbc::string DATABASE_PASSWORD = "Rtchir3ORJe2"

  ODBC user password to connect to the database.

- const float surf_keypoint_suggestion_weight = 0.4f

  Weight of SURF keypoints.

- const float sift_keypoint_suggestion_weight = 0.0f

  Weigth of sift keypoints.

- const float size_width_suggestion_weight = 0.15f

  Weight of the estimtated width.

- const float size_height_suggestion_weight = 0.15f

  Weight of the estimated height.

- const float upper_body_color_suggestion_weight = 0.15f

  Weight of the upper body color.

- const float lower_body_color_suggestion_weight = 0.15f

  Weight of the lower body color.

- const bool SAVE_PERSON_IDENTIFICATION = true

  Configure if person identification results should be saved.

- const float ALWAYS_MATCH_LIKELIHOOD = 0.5f

  Configure minimum likelihood between two persons.

- const bool PRINT_PERSON_SELECTOR_STEPS = false

  Configure if person selection steps should be printed on console.

4.2.1 Detailed Description

This namepsace contains all configuration options.

4.3 feature_extraction Namespace Reference

The feature_extraction namespace contains all classes used for feature extraction.

Classes

- class BodyPartExtractor

  This class is used to extract body parts from an input image.
- class ColorExtractor

  This class is used to extract the primary body colors.
- class Controller

  This class is used to control all feature extraction steps.
- class DirectionExtractor

  This class is used to estimate the walking direction of a person.
- class FeaturePointExtractor

  This class is used to generate sift and surf feature point descriptors.
- class FrameSelector

  This class is used to select an optimal frame in a track.
- class SizeExtractor

  This class is used to estimate body height and width.
- class TrackPersistor

  This class is used to persist tracks on the sql database.

4.3.1 Detailed Description

The feature_extraction namespace contains all classes used for feature extraction.

4.4 identification Namespace Reference

The identification namespace contains all classes used for identification.

Classes

- class ColorMatcher

  This class is used to compare the color of two tracks.
- class FeaturePointMatcher

  This class is used to compare feature point descriptors.
- class LikelihoodCalculator

  This class is used to calculate the overall likelihood two tracks match.
- class PersonAssigner

  This class is used to assign a person to each track.
- class SizeMatcher

  This class is used to compare sizes.

4.4.1 Detailed Description

The identification namespace contains all classes used for identification.

4.5 image_acquisition Namespace Reference

The image_acquisition namespace contains all classes used for image acquisition.

Classes

- class FileLoader

  This class is used to stream files in a local folder.
- class MKVFileLoader

  This class is used to streams frames from a local MKV file.
- class RTSPImageCapture

  This class is used to stream frames from a RTSP source.
- class URLImageLoader

  This class is used to stream frames from an WebURL.

4.5.1 Detailed Description

The image_acquisition namespace contains all classes used for image acquisition.

4.6 image_segmentation Namespace Reference

The image_segmentation namespace contains all classes used for image segmentation.

Classes

- class BackgroundRemover

  This class is used to remove image background.
- class Controller

  This class is used to controll all image segmentation steps.
- class PersonDetector

  This class is used to detect persons on an image.
- class PersonDetectorHog

  This class is used to detect persons on an image.

4.6.1 Detailed Description

The image_segmentation namespace contains all classes used for image segmentation.

4.7 image_tracking Namespace Reference

The image_tracking namespace contains all classes used for image tracking.

Classes

- class Controller

  This class is used to controll image tracking steps.
- class ObjectTracker

  This class is used to track objects.
- class ObjectTrackerYolo

  This class is used to track objects.

4.7.1 Detailed Description

The image_tracking namespace contains all classes used for image tracking.

Chapter 5

Class Documentation

5.1 image_segmentation::BackgroundRemover Class Reference

This class is used to remove image background.

`#include <BackgroundRemover.h>`

Public Member Functions

- BackgroundRemover (dto::Camera &camera)
  Constructor.
- void removeBackground (dto::Image &image, dto::Camera &camera) const
  Remove background from image.

5.1.1 Detailed Description

This class is used to remove image background.

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

- src/image_segmentation/BackgroundRemover.h
- src/image_segmentation/BackgroundRemover.cpp

5.2 feature_extraction::BodyPartExtractor Class Reference

This class is used to extract body parts from an input image.

`#include <BodyPartExtractor.h>`

Public Member Functions

- BodyPartExtractor ()

  Default constructor.
- ~BodyPartExtractor ()

  Default destructor.

Static Public Member Functions

- static void extractBodyParts (dto::Track &track, dto::Camera &camera)

  Extracts body parts from an input image and saves the result in the track dto.

5.2.1 Detailed Description

This class is used to extract body parts from an input image.

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

- src/feature_extraction/BodyPartExtractor.h
- src/feature_extraction/BodyPartExtractor.cpp

5.3 dto::Camera Struct Reference

The camera dto cotnains all camera properties.

```
#include <Camera.h>
```

Public Types

- enum entrySide {
  entry_left, entry_right, entry_top, entry_bottom,
  none }

  This enum lists the available options for the entry side.
- enum gateMode { minLeft, minRight, minTop, minBottom }

  The mode a gate operates in.
- enum personCountUpWhen { in_to_entry, entry_to_in }

  Configure how persons should be counted.

Public Attributes

- std::string directory

  Directory where the input files are.
- std::string prefix

  Prefix of the input files.
- int scene

  Unique scene identifier.
- entrySide entry_side

  Entry side where persons enter or leave the room.
- personCountUpWhen personCountMode

  How to count persons.
- int width

  Image width.
- int height

  Image height.
- int fps

  Frames per second of the camera.
- gateMode gateMode

  Configured gate mode.
- enum gateMode secondGateMode

  Configured second gate mode.
- int gateValue

  Number of pixels where the gate is.
- int secondGateValue

  Number of pixels where the second gate is.
- cv::Point optimalPersonLocation

  Optimal person location.
- double backgroundThreshold

  Threshold to subtract the background.
- cv::Mat cameraMatrix

  Camera matrix for calibration.
- cv::Mat distCoeffs

  Distortion Coefficients for calibration.
- double pixelToCentimeterRatio

  Pixel to Centimeter ratio.
- std::string rtspConnectionString

  If RTPS is used, the connection string.
- std::string urlConnectionString

  If Image URL is used, the url to the image.
- std::string urlUsername

  If Image URL is used, the username for authentication.
- std::string urlPassword

  If Image URL is used, the password for authentication.
- std::string videoFilePath

  If video file is used, the path to the video file.

5.3.1 Detailed Description

The camera dto cotnains all camera properties.

5.3.2 Member Enumeration Documentation

5.3.2.1 entrySide

`enum dto::Camera::entrySide`

This enum lists the available options for the entry side.

Enumerator

entry_left	The person enters the room from the left.
entry_right	The person enters the room from the right.
entry_top	The person enters the room from top.
entry_bottom	The person enters the room from bottom.
none	There is no entry side.

5.3.2.2 gateMode

`enum dto::Camera::gateMode`

The mode a gate operates in.

Enumerator

minLeft	Measure based on distance from the left side.
minRight	Measure based on distance from the rigth side.
minTop	Measure based on distance from top.
minBottom	Measure based on distance from bottom.

5.3.2.3 personCountUpWhen

`enum dto::Camera::personCountUpWhen`

Configure how persons should be counted.

Enumerator

in_to_entry	Count persons up if they are recognized in the room and go to entry side.
entry_to_in	Count persons up if they enter thorugh entry side and go in the room.

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

- src/dto/Camera.h

5.4 feature_extraction::ColorExtractor Class Reference

This class is used to extract the primary body colors.

`#include <ColorExtractor.h>`

Public Member Functions

- ColorExtractor ()
  Default Constructor.
- ~ColorExtractor ()
  Default Destructor.
- void extractMaxHue (dto::Track &track, dto::Camera &camera, cv::Mat &hsv_image, int &maxBucketId, bool isUpperBody) const
  Extract primary color and save it in maxBucketId.
- void extractPrimaryColors (dto::Track &track, dto::Camera &camera) const
  Extract primary colors of all body parts.

5.4.1 Detailed Description

This class is used to extract the primary body colors.

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

- src/feature_extraction/ColorExtractor.h
- src/feature_extraction/ColorExtractor.cpp

5.5 identification::ColorMatcher Class Reference

This class is used to compare the color of two tracks.

`#include <ColorMatcher.h>`

Public Member Functions

- ColorMatcher ()
  Default Constructor.
- ~ColorMatcher ()
  Default Destructor.
- void matchAllColors (std::vector< dto::Track > &tracks) const
  Compare colors between all tracks.

5.5.1 Detailed Description

This class is used to compare the color of two tracks.

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

- src/identification/ColorMatcher.h
- src/identification/ColorMatcher.cpp

5.6 image_tracking::Controller Class Reference

This class is used to controll image tracking steps.

`#include <Controller.h>`

Public Member Functions

- Controller ()
  Default constructor.
- void ProcessImage (dto::Image &image, dto::Camera &camera)
  Process image.

5.6.1 Detailed Description

This class is used to controll image tracking steps.

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

- src/image_tracking/Controller.h
- src/image_tracking/Controller.cpp

5.7 feature_extraction::Controller Class Reference

This class is used to control all feature extraction steps.

`#include <Controller.h>`

Public Member Functions

- Controller ()
  Default Constructor.
- void processTrack (dto::Track &track, dto::Camera &camera)
  Start feature extraction of a track.

5.7.1 Detailed Description

This class is used to control all feature extraction steps.

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

- src/feature_extraction/Controller.h
- src/feature_extraction/Controller.cpp

5.8 image_segmentation::Controller Class Reference

This class is used to controll all image segmentation steps.

`#include <Controller.h>`

Public Member Functions

- Controller (dto::Camera &camera)
  Constructor.
- void ProcessImage (dto::Image &image)
  Process image.

Static Public Member Functions

- static void ProcessImage (SYSTEMTIME *time, af::array &image, std::string path, std::string filename)
  Process an image based on static configuration.

5.8.1 Detailed Description

This class is used to controll all image segmentation steps.

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

- src/image_segmentation/Controller.h
- src/image_segmentation/Controller.cpp

5.9 dto::Track::Cv_optimalPersonBodyParts Struct Reference

This struct contains OpenCV representation of the body parts.

`#include <Track.h>`

Public Attributes

- cv::Mat head

  OpenCV representation of the head.
- cv::Mat upperBody

  OpenCV representation of the upper body.
- cv::Mat lowerBody

  OpenCV representation of the lower body.

5.9.1 Detailed Description

This struct contains OpenCV representation of the body parts.

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

- src/dto/Track.h

5.10 feature_extraction::DirectionExtractor Class Reference

This class is used to estimate the walking direction of a person.

`#include <DirectionExtractor.h>`

Public Member Functions

- DirectionExtractor ()

  Default Constructor.

Static Public Member Functions

- static void extractDirection (dto::Track &track, dto::Camera &camera)

  Extract walking direction.

5.10.1 Detailed Description

This class is used to estimate the walking direction of a person.

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

- src/feature_extraction/DirectionExtractor.h
- src/feature_extraction/DirectionExtractor.cpp

5.11 feature_extraction::FeaturePointExtractor Class Reference

This class is used to generate sift and surf feature point descriptors.

`#include <FeaturePointExtractor.h>`

Public Member Functions

- FeaturePointExtractor ()
  Default Constructor.
- ~FeaturePointExtractor ()
  Default Destructor.
- void extractFeaturePoints (dto::Track &track, dto::Camera &camera) const
  Extract sift and surf feature points.

5.11.1 Detailed Description

This class is used to generate sift and surf feature point descriptors.

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

- src/feature_extraction/FeaturePointExtractor.h
- src/feature_extraction/FeaturePointExtractor.cpp

5.12 identification::FeaturePointMatcher Class Reference

This class is used to compare feature point descriptors.

`#include <FeaturePointMatcher.h>`

Public Member Functions

- FeaturePointMatcher ()
  Default Constructor.
- ~FeaturePointMatcher ()
  Default Destructor.
- void matchAllFeaturePoints (std::vector< dto::Track > &tracks)
  Compare feature points between all tracks.

5.12.1 Detailed Description

This class is used to compare feature point descriptors.

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

- src/identification/FeaturePointMatcher.h
- src/identification/FeaturePointMatcher.cpp

5.13 image_acquisition::FileLoader Class Reference

This class is used to stream files in a local folder.

`#include <JPGFileLoader.h>`

Public Member Functions

- void WatchDirectory () const
  Start watching a local director fro changes.
- void ProcessFiles () const
  Process all files in the configured directory.
- FileLoader (std::string directory, std::string prefix, image_segmentation::Controller *segmentation_controller)
  Constructor.
- FileLoader (dto::Camera &camera, image_segmentation::Controller *segmentation_controller)
  Constructor.
- void SetDirectory (std::string directory)
  Set directory for processing files.
- void SetImageSegmentationController (image_segmentation::Controller *segmentation_controller)
  Set a new image segmentation controller.
- ~FileLoader ()
  Default Destructor.

Static Public Member Functions

- static bool isPrefix (const char *s1, const char *s2)
  Ask if one string is prefix of another string.
- static std::string extract_filename (char const *path_c)
  Extract filename from file path.

5.13.1 Detailed Description

This class is used to stream files in a local folder.

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

- src/image_acquisition/JPGFileLoader.h
- src/image_acquisition/JPGFileLoader.cpp

5.14 feature_extraction::FrameSelector Class Reference

This class is used to select an optimal frame in a track.

`#include <FrameSelector.h>`

Public Member Functions

- FrameSelector ()
  Default Constructor.
- void SelectFrame (dto::Track &track, const dto::Camera &camera) const
  Select optimal frame in track.
- void SaveRegion (dto::Track &track, const dto::Camera &camera) const
  Select optimal region in track.

5.14.1 Detailed Description

This class is used to select an optimal frame in a track.

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

- src/feature_extraction/FrameSelector.h
- src/feature_extraction/FrameSelector.cpp

5.15 dto::Image Struct Reference

The image dto struct cotnains all image properties.

`#include <Image.h>`

Public Attributes

- std::string path
  The path to the original image.
- std::string filename
  The filename of the original image.
- af::array af_image_color
  The ArrayFire representation of the image.
- SYSTEMTIME filetime
  The time the image was recorded.
- cv::Mat cv_image_original
  The OpenCV representation of the original image.
- cv::Mat cv_image_distorted
  The OpenCV representation of the distorted image.
- cv::Mat cv_image_high_contrast
  The OpenCV representation of the image with local contrast correction,.
- cv::cuda::GpuMat cv_gpu_image
  The OpenCV cuda representation of the image.
- cv::Mat cv_fgmask
  The foreground mask.
- cv::cuda::GpuMat cv_gpu_fgmask
  The OpenCV cuda representation of the foreground mask.
- cv::Mat cv_fgimg
  The OpenCV representation of the foreground image.

- cv::cuda::GpuMat cv_gpu_fgimg

  The OpenCV cuda representation of the foreground image.
- cv::Mat cv_bgimg

  The OpenCV repesentation of background image.
- cv::cuda::GpuMat cv_gpu_bgimg

  The OpenCV cuda representation of the background image.
- std::vector< Region > regions

  The person regions found in that image.
- std::vector< bbox_t > yoloPersons

  The yolo bounding boxes found in that image.

5.15.1 Detailed Description

The image dto struct cotnains all image properties.

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

- src/dto/Image.h

5.16 identification::LikelihoodCalculator Class Reference

This class is used to calculate the overall likelihood two tracks match.

`#include <LikelihoodCalculator.h>`

Public Member Functions

- LikelihoodCalculator ()

  Default Constructor.
- ~LikelihoodCalculator ()

  Default Destructor.
- void calculateAllLikelihoods (std::vector< dto::Track > &tracks) const

  Calculate overall likelihood of tracks.

5.16.1 Detailed Description

This class is used to calculate the overall likelihood two tracks match.

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

- src/identification/LikelihoodCalculator.h
- src/identification/LikelihoodCalculator.cpp

5.17 image_acquisition::MKVFileLoader Class Reference

This class is used to streams frames from a local MKV file.

`#include <MKVFileLoader.h>`

Public Member Functions

- MKVFileLoader (dto::Camera &camera, image_segmentation::Controller *segmentation_controller)
  Constructor.
- ~MKVFileLoader ()
  Default destructor.
- void process_file () const
  Process the configured mkv file.

Static Public Member Functions

- static std::string extract_filename (char const *path_c)
  Extract file name from file path.

5.17.1 Detailed Description

This class is used to streams frames from a local MKV file.

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

- src/image_acquisition/MKVFileLoader.h
- src/image_acquisition/MKVFileLoader.cpp

5.18 image_tracking::ObjectTracker Class Reference

This class is used to track objects.

`#include <ObjectTracker.h>`

Public Member Functions

- ObjectTracker ()
  Default constructor.
- void apply (dto::Image &image)
  Remove background from image.
- bool hasFinishedTracks ()
  Asks if there are any finished tracks.
- dto::Track getFinishedTrack ()
  Retrieve a finished track.
- void SendFinishedTracksTo (feature_extraction::Controller &controller, dto::Camera &camera)
  Send all finished tracks to feature extraction controller.

5.18.1 Detailed Description

This class is used to track objects.

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

- src/image_tracking/ObjectTracker.h
- src/image_tracking/ObjectTracker.cpp

5.19 image_tracking::ObjectTrackerYolo Class Reference

This class is used to track objects.

`#include <ObjectTrackerYolo.h>`

Public Member Functions

- ObjectTrackerYolo ()
  Default constructor.
- void apply (dto::Image &image)
  Remove background from image.
- bool hasFinishedTracks ()
  Ask if there are any finished tracks.
- void SendFinishedTracksTo (feature_extraction::Controller &controller, dto::Camera &camera)
  Send finished tracks to feature extraction controller.

5.19.1 Detailed Description

This class is used to track objects.

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

- src/image_tracking/ObjectTrackerYolo.h
- src/image_tracking/ObjectTrackerYolo.cpp

5.20 dto::Person Struct Reference

The person dto struct cotnains all person properties.

`#include <Person.h>`

Public Attributes

- int person_id
  A unique person id.
- std::vector< Track > tracks
  All tracks that are assigned to this person.

5.20.1 Detailed Description

The person dto struct cotnains all person properties.

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

- src/dto/Person.h

5.21 identification::PersonAssigner Class Reference

This class is used to assign a person to each track.

`#include <PersonAssigner.h>`

Public Member Functions

- PersonAssigner ()
  Default Constructor.
- ~PersonAssigner ()
  Default Destructor.

Static Public Member Functions

- static void assignTracksToPerson (std::vector< dto::Track > &tracks, std::vector< dto::Person > &persons)
  Assign a person to each track.

5.21.1 Detailed Description

This class is used to assign a person to each track.

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

- src/identification/PersonAssigner.h
- src/identification/PersonAssigner.cpp

5.22 image_segmentation::PersonDetector Class Reference

This class is used to detect persons on an image.

`#include <PersonDetector.h>`

Public Member Functions

- PersonDetector ()
  Default constructor.
- void extractPersonContours (dto::Image &Image, dto::Camera &camera) const
  Extract the contours of all persons.
- void detectPersonsYolo (dto::Image &Image, dto::Camera &camera) const
  Extract person bounding boxes using Yolo.

5.22.1 Detailed Description

This class is used to detect persons on an image.

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

- src/image_segmentation/PersonDetector.h
- src/image_segmentation/PersonDetector.cpp

5.23 image_segmentation::PersonDetectorHog Class Reference

This class is used to detect persons on an image.

`#include <PersonDetectorHog.h>`

Public Member Functions

- PersonDetectorHog ()
  Default constructor.
- void detectPerson (cv::cuda::GpuMat image) const
  Detect a person on an image using Hog.

5.23.1 Detailed Description

This class is used to detect persons on an image.

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

- src/image_segmentation/PersonDetectorHog.h
- src/image_segmentation/PersonDetectorHog.cpp

5.24 dto::Track::personSize Struct Reference

This struct contains the recognized person sizes.

`#include <Track.h>`

Public Attributes

- float height

  The recognized height in cm.
- float width

  The recognized width in cm.

5.24.1 Detailed Description

This struct contains the recognized person sizes.

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

- src/dto/Track.h

5.25 dto::Track::primaryColorIds Struct Reference

This struct contains the ids of the extracted colors.

```
#include <Track.h>
```

Public Attributes

- int upperBody

  The recognized upper body color id.
- int lowerBody

  The recognized lower body color id.

5.25.1 Detailed Description

This struct contains the ids of the extracted colors.

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

- src/dto/Track.h

5.26 dto::Region Struct Reference

The region dto struct cotnains all region properties.

```
#include <Region.h>
```

Public Attributes

- std::vector< cv::Point > contour
  The contour of the object as a list of points.
- int minX
  Minimal x values of all points of the region.
- int minY
  Minimal y values of all points of the region.
- int maxX
  Maximum x values of all points of the region.
- int maxY
  Maximum y values of all points of the region.
- float ratio
  The side ratio of the region.

5.26.1 Detailed Description

The region dto struct cotnains all region properties.

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

- src/dto/Region.h

5.27 image_acquisition::RTSPImageCapture Class Reference

This class is used to stream frames from a RTSP source.

`#include <RTSPImageCapture.h>`

Public Member Functions

- void startCapturing ()
  Start reading images from rtsp stream.
- RTSPImageCapture (dto::Camera &camera, image_segmentation::Controller *segmentation_controller)
  Constructor.
- ~RTSPImageCapture ()
  Default destructor.

5.27.1 Detailed Description

This class is used to stream frames from a RTSP source.

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

- src/image_acquisition/RTSPImageCapture.h
- src/image_acquisition/RTSPImageCapture.cpp

5.28 feature_extraction::SizeExtractor Class Reference

This class is used to estimate body height and width.

`#include <SizeExtractor.h>`

Public Member Functions

- SizeExtractor ()
  Default constructor.
- ~SizeExtractor ()
  Default destructor.

Static Public Member Functions

- static void extractBodySizes (dto::Track &track, dto::Camera camera)
  Extract body sizes from an input track and stores the result in the track dto.

5.28.1 Detailed Description

This class is used to estimate body height and width.

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

- src/feature_extraction/SizeExtractor.h
- src/feature_extraction/SizeExtractor.cpp

5.29 identification::SizeMatcher Class Reference

This class is used to compare sizes.

`#include <SizeMatcher.h>`

Public Member Functions

- SizeMatcher ()
  Default Constructor.
- ~SizeMatcher ()
  Default Destructor.
- void matchAllSizes (std::vector< dto::Track > &tracks) const
  Compare sizes between all tracks.

5.29.1 Detailed Description

This class is used to compare sizes.

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

- src/identification/SizeMatcher.h
- src/identification/SizeMatcher.cpp

5.30 dto::SQLHelper Class Reference

The SQLHelper class contains all helper functions and manages the connection state.

`#include <SQLHelper.h>`

Public Member Functions

- SQLHelper ()
  Constructor of the SQLHelper class.
- ~SQLHelper ()
  Destructor of the SQLHelper class.
- void testSQLConnection () const
  Test function to check SQL server connectivity.
- void persist_camera (const Camera &camera) const
  Persists a camera in the database.
- void retrieve_camera (Camera &camera, const char *directory, const char *prefix, const char *scene) const
  Reads camera from the database.
- void persist_track (const Track &track, const Camera &camera) const
  Persist a track on the database.
- void persist_persons (std::vector< Person > &persons) const
  Persist a person on the database.
- void retrieve_camera (Camera &camera, int camera_id) const
  Retrieve a camera from the database.
- std::vector< Track > retrieve_all_tracks () const
  Retrieve all tracks from the database.
- void backup_database (char *destination) const
  Backup the database on the server.

5.30.1 Detailed Description

The SQLHelper class contains all helper functions and manages the connection state.

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

- src/dto/SQLHelper.h
- src/dto/SQLHelper.cpp

5.31 dto::Track::suggestion Struct Reference

This struct contains a suggestion to assign this track to another one.

`#include <Track.h>`

Public Attributes

- Track * track
  A pointer to the suggested track.
- float likelihood
  A likelihood between 0 and 1 how likely the tracks match.

5.31.1 Detailed Description

This struct contains a suggestion to assign this track to another one.

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

- src/dto/Track.h

5.32 dto::Track Struct Reference

The track dto struct cotnains all track properties.

`#include <Track.h>`

Classes

- struct Cv_optimalPersonBodyParts
  This struct contains OpenCV representation of the body parts.
- struct personSize
  This struct contains the recognized person sizes.
- struct primaryColorIds
  This struct contains the ids of the extracted colors.
- struct suggestion
  This struct contains a suggestion to assign this track to another one.

Public Types

- enum WalkingDirection { out_in, in_out, out_out, in_in }
  This enum lists all possible walking directions.

Public Attributes

- int trackId

  Unique track id per camera.
- std::vector< Region > regions

  List of detected regions of one person.
- std::vector< bbox_t > persons

  List of detected bounding boxes of one person.
- std::vector< Image > **images**
- WalkingDirection **walkingDirection**
- int **numImagesWithoutRegion**
- int **optimalPersonId**
- cv::Mat **cv_optimalPersonCut**
- cv::Mat **cv_optimalPersonCut_Full**
- Cv_optimalPersonBodyParts **cv_optimalPersonBodyParts**
- primaryColorIds **primary_color_ids**
- personSize **estimatedPersonSize**
- std::vector< cv::KeyPoint > **surf_keyPoints**
- cv::Mat **surf_descriptors**
- std::vector< cv::KeyPoint > **sift_keyPoints**
- cv::Mat **sift_descriptors**
- int **track_db_id**
- Camera **camera**
- std::vector< suggestion > **surf_keypoint_suggestion**
- std::vector< suggestion > **sift_keypoint_suggestion**
- std::vector< suggestion > **size_width_suggestion**
- std::vector< suggestion > **size_height_suggestion**
- std::vector< suggestion > **upper_body_color_suggestion**
- std::vector< suggestion > **lower_body_color_suggestion**
- std::vector< suggestion > **overall_suggestion**
- int **assignedPersonId**

5.32.1 Detailed Description

The track dto struct cotnains all track properties.

5.32.2 Member Enumeration Documentation

5.32.2.1 WalkingDirection

`enum dto::Track::WalkingDirection`

This enum lists all possible walking directions.

Enumerator

out_in	Person walks in the room.
in_out	Person walks out of the room.
out_out	Person walks in the room and leaves it again.
in_in	Person moves inside the room.

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

- src/dto/Track.h

5.33 feature_extraction::TrackPersistor Class Reference

This class is used to persist tracks on the sql database.

`#include <TrackPersistor.h>`

Public Member Functions

- TrackPersistor ()
  Default constructor.
- ~TrackPersistor ()
  Default destructor.
- void persistTrack (dto::Track &track, dto::Camera &camera) const
  Persist a track in the sql database.

5.33.1 Detailed Description

This class is used to persist tracks on the sql database.

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

- src/feature_extraction/TrackPersistor.h
- src/feature_extraction/TrackPersistor.cpp

5.34 image_acquisition::URLImageLoader Class Reference

This class is used to stream frames from an WebURL.

`#include <URLImageLoader.h>`

Public Member Functions

- void startCapturing ()
  Start loading images from URL.
- URLImageLoader (dto::Camera &camera, image_segmentation::Controller *segmentation_controller)
  Constructor.
- ~URLImageLoader ()
  Default Destructor.

Static Public Member Functions

- static void publishResults (int size)
  Publish Results.

5.34.1 Detailed Description

This class is used to stream frames from an WebURL.

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

- src/image_acquisition/URLImageLoader.h
- src/image_acquisition/URLImageLoader.cpp

Chapter 6

File Documentation

6.1 src/dto/Camera.h File Reference

Contains DTO for the camera.

```
#include <string>
#include "opencv2/core.hpp"
```

Classes

- struct dto::Camera

  The camera dto cotnains all camera properties.

Namespaces

- dto

  The dto namespace contains all DTOs.

6.1.1 Detailed Description

Contains DTO for the camera.

This file contains the camera dto with all required properties.

6.2 src/dto/Configuration.h File Reference

Contains compile time program options.

```
#include <string>
#include "../image_tracking/ObjectTracker.h"
```

Namespaces

- dto

  The dto namespace contains all DTOs.
- dto::Configuration

  This namepsace contains all configuration options.

Variables

- const bool dto::Configuration::LOAD_AF_IMAGE = false

  Configure if ArrayFire image should be loaded.
- const bool dto::Configuration::CREATE_DISTORTED_IMAGE = false

  Configure if distorted images should be calculated.
- const bool dto::Configuration::SHOW_DISTORTED_IMAGE = false

  Configure if distorted images should be displayed.
- const bool dto::Configuration::SAVE_DISTORTED_IMAGE = false

  Configure if distorted images should be saved.
- const std::string dto::Configuration::DISTORTED_IMAGES_DIRECTORY = "C:\\Temp\\output\\"

  Save location for distorted images.
- const bool dto::Configuration::SHOW_ORIGINAL_IMAGES = false

  Configure if original images should be displayed.
- const bool dto::Configuration::SAVE_ORIGINAL_IMAGES = true

  Configure if original images should be saved.
- const std::string dto::Configuration::ORIGINAL_IMAGES_DIRECTORY = "C:\\Temp\\output\\"

  Original images save location.
- const bool dto::Configuration::SHOW_FG_IMAGES = true

  Configure if foreground images should be displayed.
- const bool dto::Configuration::SAVE_FG_IMAGES = true

  Configure if foreground images should be saved.
- const std::string dto::Configuration::FG_IMAGES_DIRECTORY = "C:\\Temp\\output\\"

  Original images save location.
- const bool dto::Configuration::SHOW_FG_MASKS = false

  Configure if foreground mask should be displayed.
- const bool dto::Configuration::SAVE_FG_MASK = false

  Configure if foreground mask should be saved.
- const std::string dto::Configuration::FG_MASKS_DIRECTORY = "C:\\Temp\\output\\"

  Foreground mask save location.
- const bool dto::Configuration::SHOW_ALL_CONTOURS = false

  Configure if all countours should be displayed.
- const bool dto::Configuration::SAVE_ALL_CONTOURS = true

  Configure if all contours should be saved.
- const std::string dto::Configuration::ALL_CONTOURS_DIRECTORY = "C:\\Temp\\output\\"

  All contours save directory.
- const bool dto::Configuration::SHOW_CONTOUR_IMAGES = false

  Configure if contour images should be displayed.
- const bool dto::Configuration::SAVE_CONTOUR_IMAGES = true

  Configure if contour images should be saved.
- const std::string dto::Configuration::CONTOUR_IMAGES_DIRECTORY = "C:\\Temp\\output\\"

  Contour images save directory.
- const bool dto::Configuration::SHOW_TRACK_IMAGES = false

Configure if track images should be displayed.

- const bool dto::Configuration::SAVE_TRACK_IMAGES = true

  Configure if track images should be saved.
- const std::string dto::Configuration::TRACK_IMAGES_DIRECTORY = "C:\\Temp\\output\\"

  Track images save directory.
- const int dto::Configuration::MAX_NUMBER_OF_MISSING_FRAMES_IN_TRACK = 8

  Configure the maximum number of missing frames in a track.
- const int dto::Configuration::MIN_NUMBERS_OF_FRAMES_IN_TRACK = 8

  Configure the minimum number of frames in a track.
- const std::string dto::Configuration::yoloConfig = "yolo.cfg.txt"

  Name of the yolo config file.
- const std::string dto::Configuration::yoloWeights = "yolo.weights"

  Name of the yolo weights file.
- const int dto::Configuration::yoloPersonObjectId = 0

  Yolo person object id.
- const bool dto::Configuration::PRINT_YOLO_PERSONS = false

  Configure if Yolo BBs should be printed on console.
- const bool dto::Configuration::SHOW_YOLO_PERSONS_IMAGES = true

  Configure if Yolo BBs should be displayed.
- const bool dto::Configuration::SAVE_YOLO_PERSONS_IMAGES = true

  Configure if Yolo BBs should be saved.
- const bool dto::Configuration::USE_HIGH_CONTRAST_IMAGE_FOR_YOLO = false

  Configure if local contrast correction should be applied on input image.
- const std::string dto::Configuration::YOLO_PERSONS_IMAGES_DIRECTORY = "C:\\Temp\\output\\"

  Yolo BB Images save directory.
- const int dto::Configuration::YOLO_GPU_ID = 0

  GPU ID for Yolo inferencing.
- const bool dto::Configuration::USE_YOLO_FOR_TRACKING = true

  Configure if Yolo images should be used for tracking.
- const bool dto::Configuration::USE_YOLO_FOR_FRAMESELECTION = true

  Configure if Yolo images should be used for frame selection.
- const bool dto::Configuration::USE_FG_IMAGE_FOR_YOLO = false

  Configure if FG images should be used as input for yolo.
- const std::string dto::Configuration::PERSON_COUNTER_0_PUBLISH_URL = "https://sc-analyzer-reporter.firebaseio.com/person_counter_0.json"

  Configure Publish URL for person counter.
- const float dto::Configuration::IMAGE_BORDER_PERCENTATGE = 0.03f

  Configure percentage of border. This is used for entry side detection.
- const bool dto::Configuration::SAVE_TRACK_STATISTICS = true

  Configure if statistics file for a track should be saved.
- const std::string dto::Configuration::STATISTICS_DIRECTORY = "C:\\Temp\\output\\"

  Statistics save directory.
- const double dto::Configuration::MAX_TRACKING_DISTANCE = 50

  Configure maximum number of pixels a bounding box can move between two frames.
- const bool dto::Configuration::USE_POINT_FOR_OPTIMAL_TRACK = true

  Configure if optimal track position should use a point.
- const long dto::Configuration::MAX_OPTIMAL_DISTANCE = 50

  Configure max distance from optimal point to bounding box.
- const bool dto::Configuration::PRINT_FRAME_SELECTION_STEPS = false

  Configure if frame selection steps should be printed to console.
- const bool dto::Configuration::SAVE_OPTIMAL_TRACK_IMAGE = true

Configure if optimal track image should be saved.

- const bool dto::Configuration::SAVE_OPTIMAL_TRACK_IMAGE_CUT = true

  Configure if cut optimal track image should be saved.
- const std::string dto::Configuration::OPTIMAL_TRACK_DIRECTORY = "C:\\Temp\\output\\"

  Optimal track image save directory.
- const bool dto::Configuration::SAVE_BODY_PARTS_IMAGES = true

  Configure if body part images should be saved.
- const bool dto::Configuration::SAVE_HUE_IMAGE = true

  Configure if generated image with hue only should be saved.
- const bool dto::Configuration::PRINT_HSV_VALUES = false

  Configure if HSV values should be printed on console.
- const bool dto::Configuration::SAVE_FEATURE_POINT_IMAGES = true

  Configure if feature point images should be saved.
- const bool dto::Configuration::USE_FG_IMAGE_FOR_FEATURE_POINTS = true

  Configure if foreground image should be used for feature point extraction.
- const bool dto::Configuration::STORE_TRACK_RESULTS_IN_DB = false

  Configure if tracks should be persisted in the MSSQL database.
- const nanodbc::string dto::Configuration::DATABASE_ODBC_NAME = "sc_analyzer"

  ODBC connection name. Must be configured on the system.
- const nanodbc::string dto::Configuration::DATABASE_USER = "db_user"

  ODBC user name to connect to the database.
- const nanodbc::string dto::Configuration::DATABASE_PASSWORD = "Rtchir3ORJe2"

  ODBC user password to connect to the database.
- const float dto::Configuration::surf_keypoint_suggestion_weight = 0.4f

  Weight of SURF keypoints.
- const float dto::Configuration::sift_keypoint_suggestion_weight = 0.0f

  Weigth of sift keypoints.
- const float dto::Configuration::size_width_suggestion_weight = 0.15f

  Weight of the estimtated width.
- const float dto::Configuration::size_height_suggestion_weight = 0.15f

  Weight of the estimated height.
- const float dto::Configuration::upper_body_color_suggestion_weight = 0.15f

  Weight of the upper body color.
- const float dto::Configuration::lower_body_color_suggestion_weight = 0.15f

  Weight of the lower body color.
- const bool dto::Configuration::SAVE_PERSON_IDENTIFICATION = true

  Configure if person identification results should be saved.
- const float dto::Configuration::ALWAYS_MATCH_LIKELIHOOD = 0.5f

  Configure minimum likelihood between two persons.
- const bool dto::Configuration::PRINT_PERSON_SELECTOR_STEPS = false

  Configure if person selection steps should be printed on console.

6.2.1 Detailed Description

Contains compile time program options.

This file contains all global configurable options. These options are only availabe at compile time.

6.3 src/dto/Image.h File Reference

Contains DTO for the Image.

```
#include <string>
#include <opencv2/opencv.hpp>
#include <opencv2/core/mat.hpp>
#include <arrayfire.h>
#include "Region.h"
#include <yolo/yolo_v2_class.hpp>
```

Classes

- struct dto::Image

  The image dto struct cotnains all image properties.

Namespaces

- dto

  The dto namespace contains all DTOs.

6.3.1 Detailed Description

Contains DTO for the Image.

This file contains the image dto with all required properties.

6.4 src/dto/Person.h File Reference

Contains DTO for the Person.

```
#include "Track.h"
```

Classes

- struct dto::Person

  The person dto struct cotnains all person properties.

Namespaces

- dto

  The dto namespace contains all DTOs.

6.4.1 Detailed Description

Contains DTO for the Person.

This file contains the person dto with all required properties.

6.5 src/dto/Region.h File Reference

Contains DTO for the Region.

```
#include <opencv2/core.hpp>
#include <vector>
```

Classes

- struct dto::Region

  The region dto struct cotnains all region properties.

Namespaces

- dto

  The dto namespace contains all DTOs.

6.5.1 Detailed Description

Contains DTO for the Region.

This file contains the region dto with all required properties.

6.6 src/dto/SQLHelper.h File Reference

Contains a helper class for SQL opoerations.

```
#include "../../external/nanodbc/nanodbc.h"
#include "Camera.h"
#include "Track.h"
#include "Person.h"
```

Classes

- class dto::SQLHelper

  The SQLHelper class contains all helper functions and manages the connection state.

Namespaces

- dto

  The dto namespace contains all DTOs.

6.6.1 Detailed Description

Contains a helper class for SQL opoerations.

This file contains helper functions to persist and read objects from the central database. It uses ODBC connection to access the database.

6.7 src/dto/Track.h File Reference

Contains the track dto..

```
#include <vector>
#include "Region.h"
#include "Image.h"
#include "Camera.h"
```

Classes

- struct dto::Track

  The track dto struct cotnains all track properties.
- struct dto::Track::Cv_optimalPersonBodyParts

  This struct contains OpenCV representation of the body parts.
- struct dto::Track::primaryColorIds

  This struct contains the ids of the extracted colors.
- struct dto::Track::personSize

  This struct contains the recognized person sizes.
- struct dto::Track::suggestion

  This struct contains a suggestion to assign this track to another one.

Namespaces

- dto

  The dto namespace contains all DTOs.

6.7.1 Detailed Description

Contains the track dto..

This file contains the tarc dto with all required properties.

6.8 src/feature_extraction/BodyPartExtractor.h File Reference

Contains BodyPartExtractor class.

```
#include "../dto/Camera.h"
#include "../dto/Track.h"
```

Classes

- class feature_extraction::BodyPartExtractor

  This class is used to extract body parts from an input image.

Namespaces

- feature_extraction

  The feature_extraction namespace contains all classes used for feature extraction.

6.8.1 Detailed Description

Contains BodyPartExtractor class.

This file contains the BodyPartExtractor class. It is used to split the body parts of an input image in different parts.

6.9 src/feature_extraction/ColorExtractor.h File Reference

Contains ColorExtractor class.

```
#include "../dto/Camera.h"
#include "../dto/Track.h"
```

Classes

- class feature_extraction::ColorExtractor

  This class is used to extract the primary body colors.

Namespaces

- feature_extraction

  The feature_extraction namespace contains all classes used for feature extraction.

6.9.1 Detailed Description

Contains ColorExtractor class.

This file contains the ColorExtractor class. It is used to analyze the primary color of the different body parts.

6.10 src/feature_extraction/DirectionExtractor.h File Reference

Contains DirectionExtractor class.

```
#include "../dto/Track.h"
#include "../dto/Camera.h"
```

Classes

- class feature_extraction::DirectionExtractor

  This class is used to estimate the walking direction of a person.

Namespaces

- feature_extraction

  The feature_extraction namespace contains all classes used for feature extraction.

6.10.1 Detailed Description

Contains DirectionExtractor class.

This file contains the DirectionExtractor class. It is used to analyze if a person is entering or leaving the room.

6.11 src/feature_extraction/FeaturePointExtractor.h File Reference

Contains FeaturePointExtractor class.

```
#include "../dto/Camera.h"
#include "../dto/Track.h"
#include "opencv2/features2d.hpp"
#include "opencv2/xfeatures2d/nonfree.hpp"
```

Classes

- class feature_extraction::FeaturePointExtractor

  This class is used to generate sift and surf feature point descriptors.

Namespaces

- feature_extraction

  The feature_extraction namespace contains all classes used for feature extraction.

6.11.1 Detailed Description

Contains FeaturePointExtractor class.

This file contains the FeaturePointExtractor class. It is used to detect sift and surf feature points.

6.12 src/feature_extraction/FrameSelector.h File Reference

Contains FrameSelector class.

```
#include "../dto/Camera.h"
#include "../dto/Track.h"
```

Classes

- class feature_extraction::FrameSelector

  This class is used to select an optimal frame in a track.

Namespaces

- feature_extraction

  The feature_extraction namespace contains all classes used for feature extraction.

6.12.1 Detailed Description

Contains FrameSelector class.

This file contains the FrameSelector class. It is used to choose an optimal frame for further processing.

6.13 src/feature_extraction/SizeExtractor.h File Reference

Contains SizeExtractor class.

```
#include "../dto/Camera.h"
#include "../dto/Track.h"
```

Classes

- class feature_extraction::SizeExtractor

  This class is used to estimate body height and width.

Namespaces

- feature_extraction

  The feature_extraction namespace contains all classes used for feature extraction.

6.13.1 Detailed Description

Contains SizeExtractor class.

This file contains the SizeExtractor class. It is used to estimate body height and width based on an input image.

6.14 src/feature_extraction/TrackPersistor.h File Reference

Contains TrackPersistor class.

```
#include "../dto/Camera.h"
#include "../dto/Track.h"
#include "../dto/SQLHelper.h"
```

Classes

- class feature_extraction::TrackPersistor

  This class is used to persist tracks on the sql database.

Namespaces

- feature_extraction

  The feature_extraction namespace contains all classes used for feature extraction.

6.14.1 Detailed Description

Contains TrackPersistor class.

This file contains the TrackPersistor class. It is used to persist all relevant properties of a track in the central database.

6.15 src/identification/ColorMatcher.h File Reference

Contains ColorMatcher class.

```
#include <vector>
#include "dto/Track.h"
```

Classes

- class identification::ColorMatcher

  This class is used to compare the color of two tracks.

Namespaces

- identification

  The identification namespace contains all classes used for identification.

6.15.1 Detailed Description

Contains ColorMatcher class.

This file contains the ColorMatcher class. It is used to compare the color of two persons and assign a likelihood that they contain the same person.

6.16 src/identification/FeaturePointMatcher.h File Reference

Contains FeaturePointMatcher class.

```
#include <opencv2/features2d.hpp>
#include "dto/Track.h"
#include <opencv2/xfeatures2d/nonfree.hpp>
```

Classes

- class identification::FeaturePointMatcher

  This class is used to compare feature point descriptors.

Namespaces

- identification

  The identification namespace contains all classes used for identification.

6.16.1 Detailed Description

Contains FeaturePointMatcher class.

This file contains the FeaturePointMatcher class. It is used to compare feature points (sift and surf) and assign them a likelihood that they contain the same person.

6.17 src/identification/LikelihoodCalculator.h File Reference

Contains LikelihoodCalculator class.

```
#include "dto/Track.h"
```

Classes

- class identification::LikelihoodCalculator

  This class is used to calculate the overall likelihood two tracks match.

Namespaces

- identification

  The identification namespace contains all classes used for identification.

6.17.1 Detailed Description

Contains LikelihoodCalculator class.

This file contains the LikelihoodCalculator class. It is used to calculate the overall likelihood that two tracks contain the same person.

6.18 src/identification/PersonAssigner.h File Reference

Contains PersonAssigner class.

```
#include <vector>
#include "../dto/Track.h"
#include "../dto/Person.h"
```

Classes

- class identification::PersonAssigner

  This class is used to assign a person to each track.

Namespaces

- identification

  The identification namespace contains all classes used for identification.

6.18.1 Detailed Description

Contains PersonAssigner class.

This file contains the PersonAssigner class. It is used to assign each track to a unique person.

6.19 src/identification/SizeMatcher.h File Reference

Contains SizeMatcher class.

```
#include <vector>
#include "dto/Track.h"
```

Classes

- class identification::SizeMatcher

  This class is used to compare sizes.

Namespaces

- identification

  The identification namespace contains all classes used for identification.

6.19.1 Detailed Description

Contains SizeMatcher class.

This file contains the SizeMatcher class. It is used to compare the person sizes of two tracks and assign them a likelihood that they contain the same person.

6.20 src/image_acquisition/JPGFileLoader.h File Reference

Contains JPGFileLoader class.

```
#include <string>
#include <image_segmentation/Controller.h>
#include "dto/Camera.h"
```

Classes

- class image_acquisition::FileLoader

  This class is used to stream files in a local folder.

Namespaces

- image_acquisition

  The image_acquisition namespace contains all classes used for image acquisition.

6.20.1 Detailed Description

Contains JPGFileLoader class.

This file contains the JPGFileLoader class. It is used to load all JPG files in a local folder based on their prefix.

6.21 src/image_acquisition/MKVFileLoader.h File Reference

Contains MKVFileLoader class.

```
#include <string>
#include "../image_segmentation/Controller.h"
#include "../dto/Camera.h"
```

Classes

- class image_acquisition::MKVFileLoader

  This class is used to streams frames from a local MKV file.

Namespaces

- image_acquisition

  The image_acquisition namespace contains all classes used for image acquisition.

6.21.1 Detailed Description

Contains MKVFileLoader class.

This file contains the MKVFileLoader class. It is used to load one local MKV file and stream each frame in the pipeline.

6.22 src/image_acquisition/RTSPImageCapture.h File Reference

Contains RTSPImageCapture class.

```
#include <image_segmentation/Controller.h>
#include "dto/Camera.h"
```

Classes

- class image_acquisition::RTSPImageCapture

  This class is used to stream frames from a RTSP source.

Namespaces

- image_acquisition

  The image_acquisition namespace contains all classes used for image acquisition.

6.22.1 Detailed Description

Contains RTSPImageCapture class.

This file contains the RTSPImageCapture class. It is used to connect to an rtsp source and stream each frame in the pipeline.

6.23 src/image_acquisition/URLImageLoader.h File Reference

Contains URLImageLoader class.

```
#include <image_segmentation/Controller.h>
#include "dto/Camera.h"
```

Classes

- class image_acquisition::URLImageLoader

  This class is used to stream frames from an WebURL.

Namespaces

- image_acquisition

  The image_acquisition namespace contains all classes used for image acquisition.

6.23.1 Detailed Description

Contains URLImageLoader class.

This file contains the URLImageLoader class. It is used to load images based on a URL. Each time a frame was processed, a new image is requested at the configured URL.

6.24 src/image_segmentation/BackgroundRemover.h File Reference

Contains BackgroundRemover class.

```
#include <opencv2/core.hpp>
#include <opencv2/video/background_segm.hpp>
#include <opencv2/cudabgsegm.hpp>
#include "../dto/Image.h"
#include "../dto/Camera.h"
```

Classes

- class image_segmentation::BackgroundRemover
  This class is used to remove image background.

Namespaces

- image_segmentation
  The image_segmentation namespace contains all classes used for image segmentation.

6.24.1 Detailed Description

Contains BackgroundRemover class.

This file contains the BackgroundRemover class. It is used to remove the image background.

6.25 src/image_segmentation/PersonDetector.h File Reference

Contains PersonDetector class.

```
#include "../dto/Image.h"
#include <yolo/yolo_v2_class.hpp>
#include "../dto/Camera.h"
```

Classes

- class image_segmentation::PersonDetector
  This class is used to detect persons on an image.

Namespaces

- image_segmentation
  The image_segmentation namespace contains all classes used for image segmentation.

6.25.1 Detailed Description

Contains PersonDetector class.

This file contains the PersonDetector class. It is used to detect persons based on yolo.

6.26 src/image_segmentation/PersonDetectorHog.h File Reference

Contains PersonDetectorHog class.

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

Classes

- class image_segmentation::PersonDetectorHog

  This class is used to detect persons on an image.

Namespaces

- image_segmentation

  The image_segmentation namespace contains all classes used for image segmentation.

6.26.1 Detailed Description

Contains PersonDetectorHog class.

This file contains the PersonDetectorHog class. It is used to detect persons based on the Hog algorithm.

6.27 src/image_tracking/ObjectTracker.h File Reference

Contains ObjectTracker class.

```
#include <vector>
#include "../dto/Image.h"
#include "../dto/Track.h"
#include <random>
#include "../feature_extraction/Controller.h"
```

Classes

- class image_tracking::ObjectTracker

  This class is used to track objects.

Namespaces

- image_tracking
  The image_tracking namespace contains all classes used for image tracking.

6.27.1 Detailed Description

Contains ObjectTracker class.

This file contains the ObjectTracker class. It is used to track objects based on the extracted regions.

6.28 src/image_tracking/ObjectTrackerYolo.h File Reference

Contains ObjectTrackerYolo class.

```
#include "../dto/Track.h"
#include "../dto/Image.h"
#include "../feature_extraction/Controller.h"
#include <random>
```

Classes

- class image_tracking::ObjectTrackerYolo
  This class is used to track objects.

Namespaces

- image_tracking
  The image_tracking namespace contains all classes used for image tracking.

6.28.1 Detailed Description

Contains ObjectTrackerYolo class.

This file contains the ObjectTrackerYolo class. It is used to track objects based on bounding boxes.