

Tracker

Generated by Doxygen 1.8.11

Contents

1	Tracker	1
2	Namespace Index	1
2.1	Namespace List	1
3	Hierarchical Index	2
3.1	Class Hierarchy	2
4	Class Index	2
4.1	Class List	2
5	File Index	2
5.1	File List	2
6	Namespace Documentation	3
6.1	tracker Namespace Reference	3
6.1.1	Typedef Documentation	3
7	Class Documentation	3
7.1	tracker::LAP_JVSpase< FloatT > Class Template Reference	3
7.1.1	Detailed Description	4
7.1.2	Member Function Documentation	4
7.2	tracker::LAPTrack Class Reference	7
7.2.1	Detailed Description	10
7.2.2	Member Typedef Documentation	10
7.2.3	Member Enumeration Documentation	12
7.2.4	Constructor & Destructor Documentation	12
7.2.5	Member Function Documentation	12
7.2.6	Member Data Documentation	17
7.3	tracker::LogicalError Struct Reference	22
7.3.1	Detailed Description	23
7.3.2	Constructor & Destructor Documentation	23
7.4	tracker::ParameterValueError Struct Reference	24
7.4.1	Detailed Description	24
7.4.2	Constructor & Destructor Documentation	25
7.5	tracker::Tracker Class Reference	25
7.5.1	Detailed Description	27
7.5.2	Member Typedef Documentation	27
7.5.3	Constructor & Destructor Documentation	29
7.5.4	Member Function Documentation	29
7.5.5	Member Data Documentation	30

8 File Documentation	33
8.1 LAP_JVSparse.cpp File Reference	33
8.1.1 Detailed Description	34
8.2 LAP_JVSparse.h File Reference	34
8.2.1 Detailed Description	35
8.3 LAPTrack.cpp File Reference	35
8.3.1 Detailed Description	36
8.4 LAPTrack.h File Reference	36
8.4.1 Detailed Description	37
8.5 README.md File Reference	37
8.6 Tracker.cpp File Reference	37
8.6.1 Detailed Description	38
8.7 Tracker.h File Reference	38
8.7.1 Detailed Description	39
Index	41

1 Tracker

Particle tracking trajectory connector using a sparse LAP solver.

Documentation

The Tracker Doxygen documentation can be build with the `OPT_DOC` CMake option and is also available on online:

- [Tracker HTML Manual](#)
- [Tracker PDF Manual](#)
- [Tracker github repository](#)

2 Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

tracker	3
-------------------------	-------------------

3 Hierarchical Index

3.1 Class Hierarchy

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

tracker::LAP_JVSpase< FloatT >	3
tracker::Tracker	25
tracker::LAPTrack	7
TrackerError	
tracker::LogicalError	22
tracker::ParameterValueError	24

4 Class Index

4.1 Class List

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

tracker::LAP_JVSpase< FloatT >	3
tracker::LAPTrack	7
tracker::LogicalError	
Parameter value is not valid	22
tracker::ParameterValueError	
Parameter value is not valid	24
tracker::Tracker	25

5 File Index

5.1 File List

Here is a list of all files with brief descriptions:

LAP_JVSpase.cpp	
The member definitions for the LAP Jonker Volgenant algorithm	33

LAP_JVSpase.h	The class declaration for the LAP Jonker Volgenant algorithm	34
LAPTrack.cpp	The member definitions for LAPTrack	35
LAPTrack.h	The class declaration and inline and templated functions for LAPTrack	36
Tracker.cpp	The member definitions for Tracker	37
Tracker.h	The class declaration and inline and templated functions for Tracker	38

6 Namespace Documentation

6.1 tracker Namespace Reference

Classes

- class [LAP_JVSpase](#)
- class [LAPTrack](#)
- struct [LogicalError](#)
Parameter value is not valid.
- struct [ParameterValueError](#)
Parameter value is not valid.
- class [Tracker](#)

Typedefs

- using [TrackerError](#) = `backtrace_exception::BacktraceException`

6.1.1 Typedef Documentation

6.1.1.1 using `tracker::TrackerError` = `typedef backtrace_exception::BacktraceException`

Definition at line 28 of file Tracker.h.

7 Class Documentation

7.1 `tracker::LAP_JVSpase<FloatT>` Class Template Reference

```
#include </home/travis/build/markjolah/Tracker/include/Tracker/LAP_JVSpase.h>
```

Static Public Member Functions

- static [IVecT solve](#) (const SpMatT &C)
- static void [solveLAP_orig](#) (const SpMatT &C, IVecT &x, IVecT &y, VecT &u, VecT &v)
- static VecT [computeCost](#) (const SpMatT &C, const IVecT &row_sol)
- static bool [checkCosts](#) (const SpMatT &C)
- static bool [checkSolution](#) (const SpMatT &C, const IVecT &x, const IVecT &y, const VecT &u, const VecT &v)

7.1.1 Detailed Description

```
template<class FloatT>
class tracker::LAP_JVSpase< FloatT >
```

Definition at line 21 of file LAP_JVSpase.h.

7.1.2 Member Function Documentation

7.1.2.1 `template<class FloatT > bool tracker::LAP_JVSpase< FloatT >::checkCosts (const SpMatT & C) [static]`

Definition at line 95 of file LAP_JVSpase.cpp.

7.1.2.2 `template<class FloatT > bool tracker::LAP_JVSpase< FloatT >::checkSolution (const SpMatT & C, const IVecT & x, const IVecT & y, const VecT & u, const VecT & v) [static]`

Definition at line 118 of file LAP_JVSpase.cpp.

7.1.2.3 `template<class FloatT > LAP_JVSpase< FloatT >::VecT tracker::LAP_JVSpase< FloatT >::computeCost (const SpMatT & C, const IVecT & row_sol) [static]`

Compute the total cost of a solution

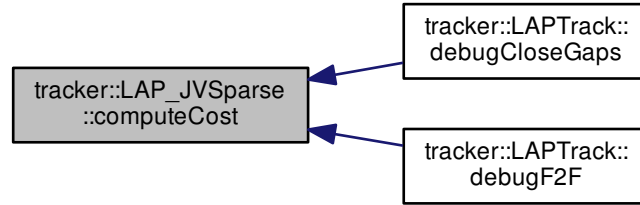
Parameters

<code>in</code>	<code>row_sol</code>	This is the 'x' output from the solver giving the col assignment for each row in order
-----------------	----------------------	--

Definition at line 85 of file LAP_JVSpase.cpp.

Referenced by `tracker::LAPTrack::debugCloseGaps()`, and `tracker::LAPTrack::debugF2F()`.

Here is the caller graph for this function:

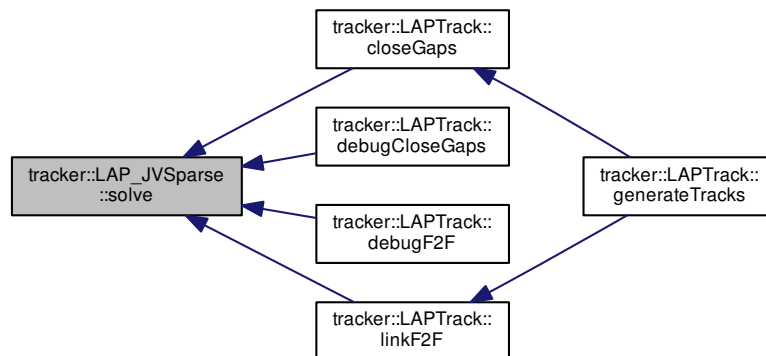


7.1.2.4 `template<class FloatT > LAP_JVSpase< FloatT >::IVecT tracker::LAP_JVSpase< FloatT >::solve (const SpMatT & C) [static]`

Definition at line 21 of file LAP_JVSpase.cpp.

Referenced by `tracker::LAPTrack::closeGaps()`, `tracker::LAPTrack::debugCloseGaps()`, `tracker::LAPTrack::debugF2F()`, and `tracker::LAPTrack::linkF2F()`.

Here is the caller graph for this function:



7.1.2.5 `template<class FloatT > void tracker::LAP_JVSpase< FloatT >::solveLAP_orig (const SpMatT & C, IVecT & x, IVecT & y, VecT & u, VecT & v) [static]`

This wraps the original sparse lap implementation that for some reason uses 1-based indexing, which we correct with some pointer arithmetic and adjusting of appropriate indicies in the sparse matrix implementation.

Furthermore because the `lap_orig` code assumes a compressed-row format, but we pass it the internal datastore of a compressed-col format sparse matrix. We invert x/y and u/v on the call to `lap_orig` to effectively let the transformation work easily with the legacy code.

This means x is the row sol and y is the col sol, as it normally would be.

Parameters

in	C	costs sparse matrix
out	x	- row assignments
out	y	- col assignments
out	u	- reduced row costs
out	v	- reduced column costs

Definition at line 50 of file LAP_JVSpase.cpp.

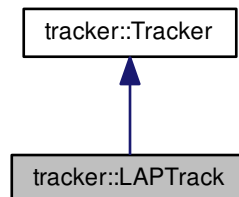
The documentation for this class was generated from the following files:

- [LAP_JVSpase.h](#)
- [LAP_JVSpase.cpp](#)

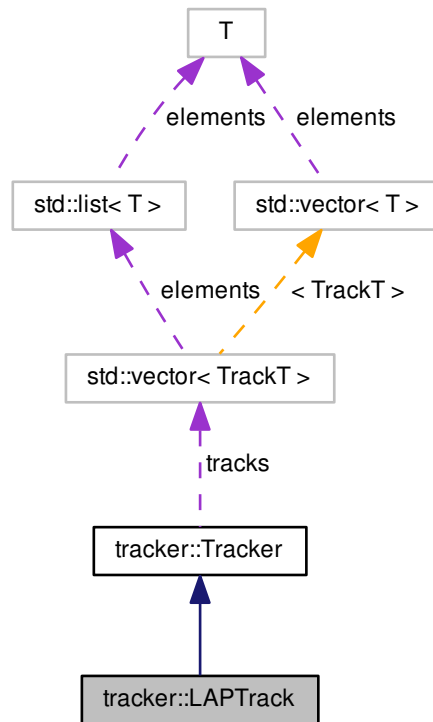
7.2 tracker::LAPTrack Class Reference

```
#include </home/travis/build/markjolah/Tracker/include/Tracker/LAPTrack.h>
```

Inheritance diagram for tracker::LAPTrack:



Collaboration diagram for `tracker::LAPTrack`:



Public Types

- using `SpMatT` = `arma::SpMat< FloatT >`
- using `UVecT` = `arma::Col< arma::uword >`
- using `UMatT` = `arma::umat`
- using `FloatT` = `double`
- using `IdxT` = `int32_t`
- using `VecT` = `arma::Col< FloatT >`
- using `MatT` = `arma::Mat< FloatT >`
- using `IVecT` = `arma::Col< IdxT >`
- using `IMatT` = `arma::Mat< IdxT >`
- using `IVecFieldT` = `arma::field< IVecT >`
- using `IndexVectorT` = `std::vector< IdxT >`
- using `TrackT` = `std::list< IdxT >`
- using `TrackVecT` = `std::vector< TrackT >`
- using `ParamT` = `std::map< std::string, FloatT >`
- using `VecParamT` = `std::map< std::string, VecT >`

Public Member Functions

- [LAPTrack](#) (const [VecParamT](#) ¶m)
- [VecParamT](#) [getStats](#) () const
- void [initializeTracks](#) (const [IVecT](#) &frameIdx_, const [MatT](#) &position_, const [MatT](#) &SE_position_)
- void [initializeTracks](#) (const [IVecT](#) &frameIdx_, const [MatT](#) &position_, const [MatT](#) &SE_position_, const [MatT](#) &feature_, const [MatT](#) &SE_feature_)
- void [linkF2F](#) ()
- void [closeGaps](#) ()
- [SpMatT](#) [computeF2FCostMat](#) (int curFrame, int nextFrame) const
- void [debugF2F](#) (int [frameIdx](#), [IVecT](#) &cur_locs, [IVecT](#) &next_locs, [SpMatT](#) &cost, [IMatT](#) &connections, [VecT](#) &conn_costs) const
- void [debugCloseGaps](#) ([SpMatT](#) &cost, [IMatT](#) &connections, [VecT](#) &conn_costs) const
- [SpMatT](#) [computeGapCloseMatrix](#) () const
- void [generateTracks](#) ()
- void [checkFrameIdxs](#) ()
- void [printTracks](#) () const

Public Attributes

- [FloatT](#) [D](#)
- [FloatT](#) [kon](#)
- [FloatT](#) [koff](#)
- [FloatT](#) [rho](#)
- [VecT](#) [featureVar](#)
- [FloatT](#) [maxSpeed](#) = 0
- [FloatT](#) [maxPositionDisplacementSigma](#) = 5.0
- [VecT](#) [maxFeatureDisplacementSigma](#)
- [IdxT](#) [maxGapCloseFrames](#) = 20
- [IdxT](#) [minGapCloseTrackLength](#) = 1
- [IdxT](#) [minFinalTrackLength](#) = 1
- const [FloatT](#) [cost_epsilon](#) = std::numeric_limits<[FloatT](#)>::epsilon()
- [IdxT](#) [N](#) = 0
- [IdxT](#) [nDims](#) = 0
- [IdxT](#) [nFeatures](#) = 0
- [IVecT](#) [frameIdx](#)
- [MatT](#) [position](#)
- [MatT](#) [SE_position](#)
- [MatT](#) [feature](#)
- [MatT](#) [SE_feature](#)
- [IdxT](#) [firstFrame](#) = 0
- [IdxT](#) [lastFrame](#) = 0
- [IdxT](#) [nFrames](#) = 0
- [IVecT](#) [nFrameLocs](#)
- [IVecFieldT](#) [frameLocIdx](#)
- [TrackVecT](#) [tracks](#)

Protected Types

- enum [StateT](#) { [UNTRACKED](#), [F2F_LINKED](#), [GAPS_CLOSED](#) }

Protected Attributes

- [FloatT minCost](#) = 1e-6
- [FloatT log1mkoff](#)
- [FloatT log1mkon](#)
- [FloatT logrho](#)
- [FloatT logkon](#)
- [FloatT logkoff](#)
- [StateT state](#)
- [IndexVectorT birthFrameIdx](#)
- [IVecT frameBirthStartIdx](#)
- [IVecT trackAssignment](#)

Static Protected Attributes

- static const [FloatT log2pi](#) = log(2*arma::Datum<[Tracker::FloatT](#)>::pi)

7.2.1 Detailed Description

Definition at line 16 of file LAPTrack.h.

7.2.2 Member Typedef Documentation

7.2.2.1 using [tracker::Tracker::FloatT](#) = double [inherited]

Definition at line 47 of file Tracker.h.

7.2.2.2 using [tracker::Tracker::IdxT](#) = int32_t [inherited]

Definition at line 48 of file Tracker.h.

7.2.2.3 using [tracker::Tracker::IMatT](#) = arma::Mat<[IdxT](#)> [inherited]

Definition at line 52 of file Tracker.h.

7.2.2.4 using [tracker::Tracker::IndexVectorT](#) = std::vector<[IdxT](#)> [inherited]

Definition at line 54 of file Tracker.h.

7.2.2.5 using [tracker::Tracker::IVecFieldT](#) = arma::field<[IVecT](#)> [inherited]

Definition at line 53 of file Tracker.h.

7.2.2.6 `using tracker::Tracker::IVecT = arma::Col<IdxT> [inherited]`

Definition at line 51 of file Tracker.h.

7.2.2.7 `using tracker::Tracker::MatT = arma::Mat<FloatT> [inherited]`

Definition at line 50 of file Tracker.h.

7.2.2.8 `using tracker::Tracker::ParamT = std::map<std::string,FloatT> [inherited]`

A convenient form for reporting dictionaries of named FP data to matlab

Definition at line 57 of file Tracker.h.

7.2.2.9 `using tracker::LAPTrack::SpMatT = arma::SpMat<FloatT>`

Definition at line 18 of file LAPTrack.h.

7.2.2.10 `using tracker::Tracker::TrackT = std::list<IdxT> [inherited]`

A type for an individual track

Definition at line 55 of file Tracker.h.

7.2.2.11 `using tracker::Tracker::TrackVecT = std::vector<TrackT> [inherited]`

A type for a vector of tracks

Definition at line 56 of file Tracker.h.

7.2.2.12 `using tracker::LAPTrack::UMatT = arma::umat`

Definition at line 20 of file LAPTrack.h.

7.2.2.13 `using tracker::LAPTrack::UVecT = arma::Col<arma::uword>`

Definition at line 19 of file LAPTrack.h.

7.2.2.14 `using tracker::Tracker::VecParamT = std::map<std::string,VecT> [inherited]`

A convenient form for reporting dictionaries of named FP data to matlab

Definition at line 58 of file Tracker.h.

7.2.2.15 `using tracker::Tracker::VecT = arma::Col<FloatT> [inherited]`

Definition at line 49 of file Tracker.h.

7.2.3 Member Enumeration Documentation

7.2.3.1 enum tracker::LAPTrack::StateT [protected]

Enumerator

UNTRACKED

F2F_LINKED

GAPS_CLOSED

Definition at line 58 of file LAPTrack.h.

7.2.4 Constructor & Destructor Documentation

7.2.4.1 tracker::LAPTrack::LAPTrack (const VecParamT & param)

Definition at line 11 of file LAPTrack.cpp.

References D, featureVar, koff, kon, log1mkoff, log1mkon, logkoff, logkon, logrho, maxFeatureDisplacementSigma, maxGapCloseFrames, maxPositionDisplacementSigma, maxSpeed, minFinalTrackLength, minGapCloseTrackLength, and rho.

7.2.5 Member Function Documentation

7.2.5.1 void tracker::LAPTrack::checkFrameIdxs ()

Definition at line 335 of file LAPTrack.cpp.

References F2F_LINKED, tracker::Tracker::firstFrame, frameBirthStartIdx, tracker::Tracker::frameIdx, tracker::Tracker::lastFrame, state, and tracker::Tracker::tracks.

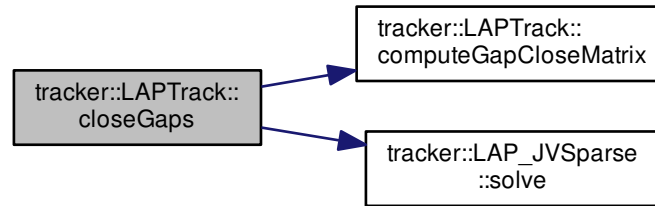
7.2.5.2 void tracker::LAPTrack::closeGaps ()

Definition at line 376 of file LAPTrack.cpp.

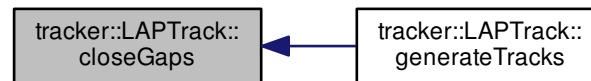
References birthFrameIdx, computeGapCloseMatrix(), F2F_LINKED, frameBirthStartIdx, GAPS_CLOSED, minFinalTrackLength, tracker::LAP_JVSParse< FloatT >::solve(), state, tracker::Tracker::trackAssignment, and tracker::Tracker::tracks.

Referenced by generateTracks().

Here is the call graph for this function:



Here is the caller graph for this function:



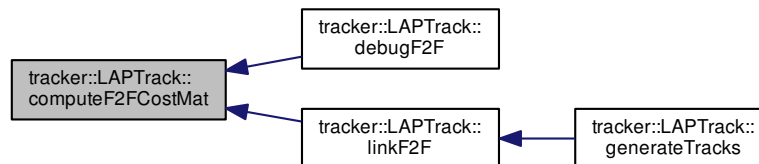
7.2.5.3 LAPTrack::SpMatT tracker::LAPTrack::computeF2FCostMat (int curFrame, int nextFrame) const

Definition at line 231 of file LAPTrack.cpp.

References `cost_epsilon`, `D`, `tracker::Tracker::feature`, `featureVar`, `tracker::Tracker::firstFrame`, `tracker::Tracker::frameLocIdx`, `log1mkoff`, `tracker::Tracker::log2pi`, `logkoff`, `logkon`, `logrho`, `maxFeatureDisplacementSigma`, `maxPositionDisplacementSigma`, `maxSpeed`, `tracker::Tracker::nDims`, `tracker::Tracker::nFeatures`, `tracker::Tracker::nFrameLocs`, `tracker::Tracker::position`, `tracker::Tracker::SE_feature`, and `tracker::Tracker::SE_position`.

Referenced by `debugF2F()`, and `linkF2F()`.

Here is the caller graph for this function:



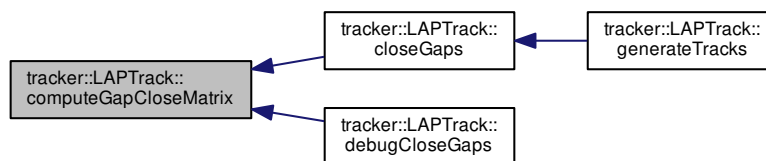
7.2.5.4 LAPTrack::SpMatT tracker::LAPTrack::computeGapCloseMatrix () const

Definition at line 415 of file LAPTrack.cpp.

References birthFrameIdx, cost_epsilon, D, tracker::Tracker::feature, featureVar, tracker::Tracker::firstFrame, frameBirthStartIdx, tracker::Tracker::frameIdx, tracker::Tracker::lastFrame, tracker::Tracker::log2pi, logkoff, logkon, logrho, maxFeatureDisplacementSigma, maxGapCloseFrames, maxPositionDisplacementSigma, maxSpeed, minGapCloseTrackLength, tracker::Tracker::nDims, tracker::Tracker::nFeatures, tracker::Tracker::position, tracker::Tracker::SE_feature, tracker::Tracker::SE_position, and tracker::Tracker::tracks.

Referenced by closeGaps(), and debugCloseGaps().

Here is the caller graph for this function:

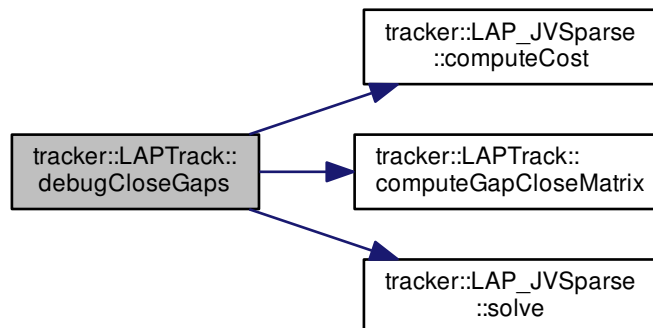


7.2.5.5 void tracker::LAPTrack::debugCloseGaps (SpMatT & cost, IMatT & connections, VecT & conn_costs) const

Definition at line 351 of file LAPTrack.cpp.

References tracker::LAP_JVSpase< FloatT >::computeCost(), computeGapCloseMatrix(), cost_epsilon, F2F_LINKED, tracker::LAP_JVSpase< FloatT >::solve(), state, and tracker::Tracker::tracks.

Here is the call graph for this function:

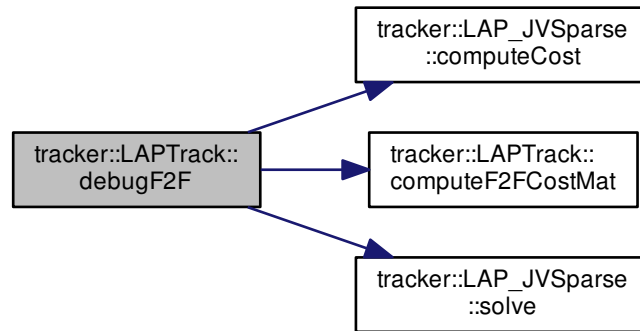


7.2.5.6 void tracker::LAPTrack::debugF2F (int *frameldx*, IVecT & *cur_locs*, IVecT & *next_locs*, SpMatT & *cost*, IMatT & *connections*, VecT & *conn_costs*) const

Definition at line 92 of file LAPTrack.cpp.

References `tracker::LAP_JV Sparse< FloatT >::computeCost()`, `computeF2FCostMat()`, `cost_epsilon`, `tracker::Tracker::firstFrame`, `tracker::Tracker::frameLocIdx`, `tracker::Tracker::lastFrame`, `tracker::Tracker::nFrameLocs`, and `tracker::LAP_JV Sparse< FloatT >::solve()`.

Here is the call graph for this function:



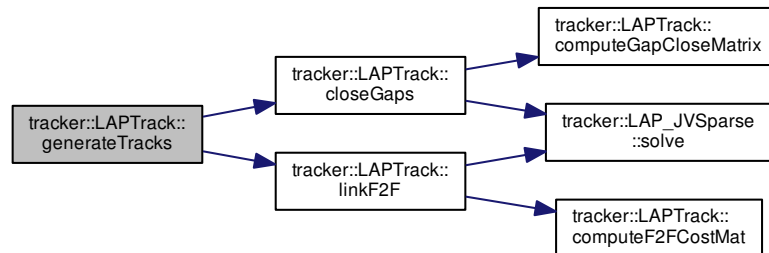
7.2.5.7 void tracker::LAPTrack::generateTracks () [virtual]

Implements [tracker::Tracker](#).

Definition at line 77 of file LAPTrack.cpp.

References `closeGaps()`, `F2F_LINKED`, `GAPS_CLOSED`, `linkF2F()`, `state`, and `UNTRACKED`.

Here is the call graph for this function:



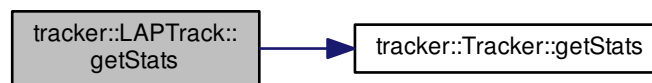
7.2.5.8 LAPTrack::VecParamT tracker::LAPTrack::getStats () const [virtual]

Reimplemented from [tracker::Tracker](#).

Definition at line 46 of file LAPTrack.cpp.

References `D`, `featureVar`, `tracker::Tracker::getStats()`, `koff`, `kon`, `maxFeatureDisplacementSigma`, `maxGapCloseFrames`, `maxPositionDisplacementSigma`, `maxSpeed`, `minFinalTrackLength`, `minGapCloseTrackLength`, and `rho`.

Here is the call graph for this function:



7.2.5.9 void tracker::LAPTrack::initializeTracks (const IVecT & *frameldx_*, const MatT & *position_*, const MatT & *SE_position_*) [virtual]

Reimplemented from [tracker::Tracker](#).

Definition at line 63 of file LAPTrack.cpp.

7.2.5.10 void tracker::LAPTrack::initializeTracks (const IVecT & *frameldx_*, const MatT & *position_*, const MatT & *SE_position_*, const MatT & *feature_*, const MatT & *SE_feature_*) [virtual]

Reimplemented from [tracker::Tracker](#).

Definition at line 69 of file LAPTrack.cpp.

References `birthFrameldx`, `frameBirthStartIdx`, `tracker::Tracker::initializeTracks()`, `state`, and `UNTRACKED`.

Here is the call graph for this function:



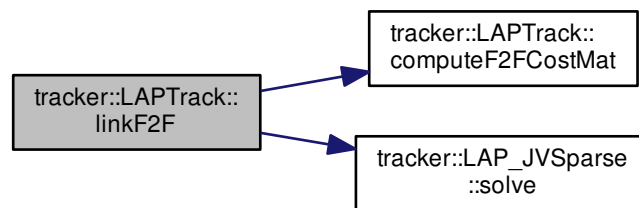
7.2.5.11 void tracker::LAPTrack::linkF2F ()

Definition at line 130 of file LAPTrack.cpp.

References birthFrameIdx, computeF2FCostMat(), F2F_LINKED, tracker::Tracker::firstFrame, frameBirthStartIdx, tracker::Tracker::frameLocIdx, tracker::Tracker::lastFrame, tracker::Tracker::nFrameLocs, tracker::Tracker::nFrames, tracker::LAP_JVSpase< FloatT >::solve(), state, tracker::Tracker::trackAssignment, tracker::Tracker::tracks, and UNTRACKED.

Referenced by generateTracks().

Here is the call graph for this function:



Here is the caller graph for this function:



7.2.5.12 void tracker::Tracker::printTracks () const [inherited]

Definition at line 126 of file Tracker.cpp.

References tracker::Tracker::frameIdx, and tracker::Tracker::tracks.

7.2.6 Member Data Documentation

7.2.6.1 IndexVectorT tracker::LAPTrack::birthFrameIdx [protected]

Definition at line 62 of file LAPTrack.h.

Referenced by closeGaps(), computeGapCloseMatrix(), initializeTracks(), and linkF2F().

7.2.6.2 `const FloatT tracker::LAPTrack::cost_epsilon = std::numeric_limits<FloatT>::epsilon()`

Definition at line 35 of file LAPTrack.h.

Referenced by `computeF2FCostMat()`, `computeGapCloseMatrix()`, `debugCloseGaps()`, and `debugF2F()`.

7.2.6.3 `FloatT tracker::LAPTrack::D`

Definition at line 22 of file LAPTrack.h.

Referenced by `computeF2FCostMat()`, `computeGapCloseMatrix()`, `getStats()`, and `LAPTrack()`.

7.2.6.4 `MatT tracker::Tracker::feature` `[inherited]`

Definition at line 66 of file Tracker.h.

Referenced by `computeF2FCostMat()`, `computeGapCloseMatrix()`, and `tracker::Tracker::initializeTracks()`.

7.2.6.5 `VecT tracker::LAPTrack::featureVar`

Definition at line 26 of file LAPTrack.h.

Referenced by `computeF2FCostMat()`, `computeGapCloseMatrix()`, `getStats()`, and `LAPTrack()`.

7.2.6.6 `IdxT tracker::Tracker::firstFrame = 0` `[inherited]`

Definition at line 68 of file Tracker.h.

Referenced by `checkFrameIdxs()`, `computeF2FCostMat()`, `computeGapCloseMatrix()`, `debugF2F()`, `tracker::Tracker::getStats()`, `tracker::Tracker::initializeTracks()`, and `linkF2F()`.

7.2.6.7 `IVecT tracker::LAPTrack::frameBirthStartIdx` `[protected]`

Definition at line 63 of file LAPTrack.h.

Referenced by `checkFrameIdxs()`, `closeGaps()`, `computeGapCloseMatrix()`, `initializeTracks()`, and `linkF2F()`.

7.2.6.8 `IVecT tracker::Tracker::frameldx` `[inherited]`

Definition at line 63 of file Tracker.h.

Referenced by `checkFrameIdxs()`, `computeGapCloseMatrix()`, `tracker::Tracker::initializeTracks()`, and `tracker::Tracker::printTracks()`.

7.2.6.9 `IVecFieldT tracker::Tracker::frameLocIdx` `[inherited]`

Definition at line 74 of file Tracker.h.

Referenced by `computeF2FCostMat()`, `debugF2F()`, `tracker::Tracker::initializeTracks()`, and `linkF2F()`.

7.2.6.10 FloatT tracker::LAPTrack::koff

Definition at line 24 of file LAPTrack.h.

Referenced by `getStats()`, and `LAPTrack()`.

7.2.6.11 FloatT tracker::LAPTrack::kon

Definition at line 23 of file LAPTrack.h.

Referenced by `getStats()`, and `LAPTrack()`.

7.2.6.12 IdxT tracker::Tracker::lastFrame = 0 `[inherited]`

Definition at line 69 of file Tracker.h.

Referenced by `checkFrameIdxs()`, `computeGapCloseMatrix()`, `debugF2F()`, `tracker::Tracker::getStats()`, `tracker::Tracker::initializeTracks()`, and `linkF2F()`.

7.2.6.13 FloatT tracker::LAPTrack::log1mkoff `[protected]`

Definition at line 52 of file LAPTrack.h.

Referenced by `computeF2FCostMat()`, and `LAPTrack()`.

7.2.6.14 FloatT tracker::LAPTrack::log1mkon `[protected]`

Definition at line 53 of file LAPTrack.h.

Referenced by `LAPTrack()`.

7.2.6.15 const Tracker::FloatT tracker::Tracker::log2pi = log(2*arma::Datum<Tracker::FloatT>::pi) `[static], [protected], [inherited]`

Definition at line 92 of file Tracker.h.

Referenced by `computeF2FCostMat()`, and `computeGapCloseMatrix()`.

7.2.6.16 FloatT tracker::LAPTrack::logkoff `[protected]`

Definition at line 56 of file LAPTrack.h.

Referenced by `computeF2FCostMat()`, `computeGapCloseMatrix()`, and `LAPTrack()`.

7.2.6.17 FloatT tracker::LAPTrack::logkon `[protected]`

Definition at line 55 of file LAPTrack.h.

Referenced by `computeF2FCostMat()`, `computeGapCloseMatrix()`, and `LAPTrack()`.

7.2.6.18 FloatT tracker::LAPTrack::logrho [protected]

Definition at line 54 of file LAPTrack.h.

Referenced by computeF2FCostMat(), computeGapCloseMatrix(), and LAPTrack().

7.2.6.19 VecT tracker::LAPTrack::maxFeatureDisplacementSigma

Definition at line 29 of file LAPTrack.h.

Referenced by computeF2FCostMat(), computeGapCloseMatrix(), getStats(), and LAPTrack().

7.2.6.20 IdxT tracker::LAPTrack::maxGapCloseFrames = 20

Definition at line 30 of file LAPTrack.h.

Referenced by computeGapCloseMatrix(), getStats(), and LAPTrack().

7.2.6.21 FloatT tracker::LAPTrack::maxPositionDisplacementSigma = 5.0

Definition at line 28 of file LAPTrack.h.

Referenced by computeF2FCostMat(), computeGapCloseMatrix(), getStats(), and LAPTrack().

7.2.6.22 FloatT tracker::LAPTrack::maxSpeed = 0

Definition at line 27 of file LAPTrack.h.

Referenced by computeF2FCostMat(), computeGapCloseMatrix(), getStats(), and LAPTrack().

7.2.6.23 FloatT tracker::LAPTrack::minCost = 1e-6 [protected]

Definition at line 51 of file LAPTrack.h.

7.2.6.24 IdxT tracker::LAPTrack::minFinalTrackLength = 1

Definition at line 32 of file LAPTrack.h.

Referenced by closeGaps(), getStats(), and LAPTrack().

7.2.6.25 IdxT tracker::LAPTrack::minGapCloseTrackLength = 1

Definition at line 31 of file LAPTrack.h.

Referenced by computeGapCloseMatrix(), getStats(), and LAPTrack().

7.2.6.26 `IdxT tracker::Tracker::N = 0` `[inherited]`

Definition at line 60 of file Tracker.h.

Referenced by `tracker::Tracker::getStats()`, and `tracker::Tracker::initializeTracks()`.

7.2.6.27 `IdxT tracker::Tracker::nDims = 0` `[inherited]`

Definition at line 61 of file Tracker.h.

Referenced by `computeF2FCostMat()`, `computeGapCloseMatrix()`, `tracker::Tracker::getStats()`, and `tracker::Tracker::initializeTracks()`.

7.2.6.28 `IdxT tracker::Tracker::nFeatures = 0` `[inherited]`

Definition at line 62 of file Tracker.h.

Referenced by `computeF2FCostMat()`, `computeGapCloseMatrix()`, `tracker::Tracker::getStats()`, and `tracker::Tracker::initializeTracks()`.

7.2.6.29 `IVecT tracker::Tracker::nFrameLocs` `[inherited]`

Definition at line 73 of file Tracker.h.

Referenced by `computeF2FCostMat()`, `debugF2F()`, `tracker::Tracker::initializeTracks()`, and `linkF2F()`.

7.2.6.30 `IdxT tracker::Tracker::nFrames = 0` `[inherited]`

Definition at line 70 of file Tracker.h.

Referenced by `tracker::Tracker::getStats()`, `tracker::Tracker::initializeTracks()`, and `linkF2F()`.

7.2.6.31 `MatT tracker::Tracker::position` `[inherited]`

Definition at line 64 of file Tracker.h.

Referenced by `computeF2FCostMat()`, `computeGapCloseMatrix()`, and `tracker::Tracker::initializeTracks()`.

7.2.6.32 `FloatT tracker::LAPTrack::rho`

Definition at line 25 of file LAPTrack.h.

Referenced by `getStats()`, and `LAPTrack()`.

7.2.6.33 `MatT tracker::Tracker::SE_feature` `[inherited]`

Definition at line 67 of file Tracker.h.

Referenced by `computeF2FCostMat()`, `computeGapCloseMatrix()`, and `tracker::Tracker::initializeTracks()`.

7.2.6.34 **MatT** `tracker::Tracker::SE_position` `[inherited]`

Definition at line 65 of file Tracker.h.

Referenced by `computeF2FCostMat()`, `computeGapCloseMatrix()`, and `tracker::Tracker::initializeTracks()`.

7.2.6.35 **StateT** `tracker::LAPTrack::state` `[protected]`

Definition at line 59 of file LAPTrack.h.

Referenced by `checkFrameIdxs()`, `closeGaps()`, `debugCloseGaps()`, `generateTracks()`, `initializeTracks()`, and `linkF2F()`.

7.2.6.36 **IVecT** `tracker::Tracker::trackAssignment` `[protected]`, `[inherited]`

Definition at line 93 of file Tracker.h.

Referenced by `closeGaps()`, `tracker::Tracker::getStats()`, `tracker::Tracker::initializeTracks()`, and `linkF2F()`.

7.2.6.37 **TrackVecT** `tracker::Tracker::tracks` `[inherited]`

Definition at line 77 of file Tracker.h.

Referenced by `checkFrameIdxs()`, `closeGaps()`, `computeGapCloseMatrix()`, `debugCloseGaps()`, `tracker::Tracker::getStats()`, `tracker::Tracker::initializeTracks()`, `linkF2F()`, and `tracker::Tracker::printTracks()`.

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

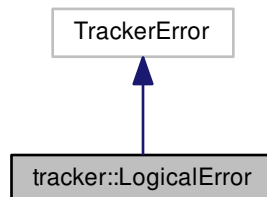
- [LAPTrack.h](#)
- [LAPTrack.cpp](#)

7.3 `tracker::LogicalError` Struct Reference

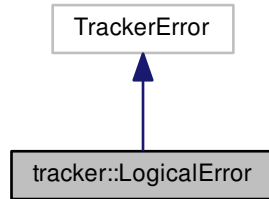
Parameter value is not valid.

```
#include </home/travis/build/markjolah/Tracker/include/Tracker/Tracker.h>
```

Inheritance diagram for `tracker::LogicalError`:



Collaboration diagram for `tracker::LogicalError`:



Public Member Functions

- [LogicalError](#) (`std::string message`)

7.3.1 Detailed Description

Parameter value is not valid.

Definition at line 40 of file `Tracker.h`.

7.3.2 Constructor & Destructor Documentation

7.3.2.1 `tracker::LogicalError::LogicalError (std::string message) [inline]`

Definition at line 42 of file `Tracker.h`.

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

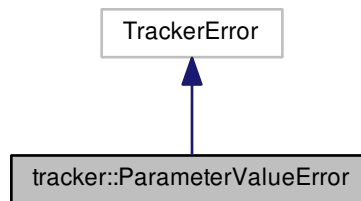
- [Tracker.h](#)

7.4 tracker::ParameterValueError Struct Reference

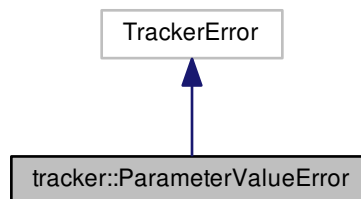
Parameter value is not valid.

```
#include </home/travis/build/markjolah/Tracker/include/Tracker/Tracker.h>
```

Inheritance diagram for tracker::ParameterValueError:



Collaboration diagram for tracker::ParameterValueError:



Public Member Functions

- [ParameterValueError](#) (std::string message)

7.4.1 Detailed Description

Parameter value is not valid.

Definition at line 33 of file Tracker.h.

7.4.2 Constructor & Destructor Documentation

7.4.2.1 tracker::ParameterValueError::ParameterValueError (std::string *message*) [inline]

Definition at line 35 of file Tracker.h.

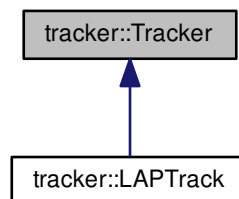
The documentation for this struct was generated from the following file:

- [Tracker.h](#)

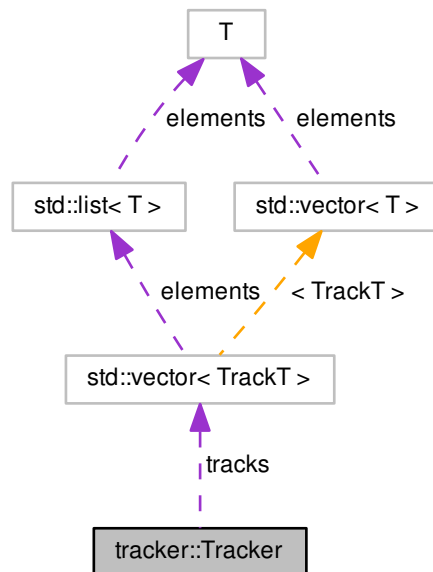
7.5 tracker::Tracker Class Reference

```
#include </home/travis/build/markjolah/Tracker/include/Tracker/Tracker.h>
```

Inheritance diagram for tracker::Tracker:



Collaboration diagram for `tracker::Tracker`:



Public Types

- using `FloatT` = `double`
- using `IdxT` = `int32_t`
- using `VecT` = `arma::Col< FloatT >`
- using `MatT` = `arma::Mat< FloatT >`
- using `IVecT` = `arma::Col< IdxT >`
- using `IMatT` = `arma::Mat< IdxT >`
- using `IVecFieldT` = `arma::field< IVecT >`
- using `IndexVectorT` = `std::vector< IdxT >`
- using `TrackT` = `std::list< IdxT >`
- using `TrackVecT` = `std::vector< TrackT >`
- using `ParamT` = `std::map< std::string, FloatT >`
- using `VecParamT` = `std::map< std::string, VecT >`

Public Member Functions

- `Tracker` (`const VecParamT ¶m`)
- virtual `~Tracker` ()
- virtual `VecParamT getStats` () const
- virtual void `initializeTracks` (`const IVecT &frameldx_`, `const MatT &position_`, `const MatT &SE_position_`)
- virtual void `initializeTracks` (`const IVecT &frameldx_`, `const MatT &position_`, `const MatT &SE_position_`, `const MatT &feature_`, `const MatT &SE_feature_`)
- virtual void `generateTracks` ()=0
- void `printTracks` () const

Public Attributes

- [IdxT N](#) = 0
- [IdxT nDims](#) = 0
- [IdxT nFeatures](#) = 0
- [IVecT frameIdx](#)
- [MatT position](#)
- [MatT SE_position](#)
- [MatT feature](#)
- [MatT SE_feature](#)
- [IdxT firstFrame](#) = 0
- [IdxT lastFrame](#) = 0
- [IdxT nFrames](#) = 0
- [IVecT nFrameLocs](#)
- [IVecFieldT frameLocIdx](#)
- [TrackVecT tracks](#)

Protected Attributes

- [IVecT trackAssignment](#)

Static Protected Attributes

- static const [FloatT log2pi](#) = log(2*arma::Datum<[Tracker::FloatT](#)>::pi)

7.5.1 Detailed Description

Definition at line 45 of file Tracker.h.

7.5.2 Member Typedef Documentation

7.5.2.1 using tracker::Tracker::FloatT = double

Definition at line 47 of file Tracker.h.

7.5.2.2 using tracker::Tracker::IdxT = int32_t

Definition at line 48 of file Tracker.h.

7.5.2.3 using tracker::Tracker::IMatT = arma::Mat<IdxT>

Definition at line 52 of file Tracker.h.

7.5.2.4 using `tracker::Tracker::IndexVectorT = std::vector<IdxT>`

Definition at line 54 of file Tracker.h.

7.5.2.5 using `tracker::Tracker::IVecFieldT = arma::field<IVecT>`

Definition at line 53 of file Tracker.h.

7.5.2.6 using `tracker::Tracker::IVecT = arma::Col<IdxT>`

Definition at line 51 of file Tracker.h.

7.5.2.7 using `tracker::Tracker::MatT = arma::Mat<FloatT>`

Definition at line 50 of file Tracker.h.

7.5.2.8 using `tracker::Tracker::ParamT = std::map<std::string,FloatT>`

A convenient form for reporting dictionaries of named FP data to matlab

Definition at line 57 of file Tracker.h.

7.5.2.9 using `tracker::Tracker::TrackT = std::list<IdxT>`

A type for an individual track

Definition at line 55 of file Tracker.h.

7.5.2.10 using `tracker::Tracker::TrackVecT = std::vector<TrackT>`

A type for a vector of tracks

Definition at line 56 of file Tracker.h.

7.5.2.11 using `tracker::Tracker::VecParamT = std::map<std::string,VecT>`

A convenient form for reporting dictionaries of named FP data to matlab

Definition at line 58 of file Tracker.h.

7.5.2.12 using `tracker::Tracker::VecT = arma::Col<FloatT>`

Definition at line 49 of file Tracker.h.

7.5.3 Constructor & Destructor Documentation

7.5.3.1 tracker::Tracker::Tracker (const VecParamT & param)

param - A dictionary of floating point values to pass in. This is a flexible interface to the higher-level matlab code allowing each subclass to take in arbitrary floating point arguments.

Definition at line 15 of file Tracker.cpp.

7.5.3.2 virtual tracker::Tracker::~Tracker () [inline],[virtual]

Definition at line 84 of file Tracker.h.

7.5.4 Member Function Documentation

7.5.4.1 virtual void tracker::Tracker::generateTracks () [pure virtual]

Implemented in [tracker::LAPTrack](#).

7.5.4.2 Tracker::VecParamT tracker::Tracker::getStats () const [virtual]

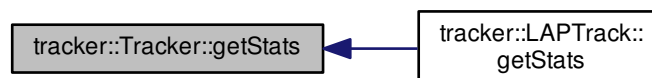
Reimplemented in [tracker::LAPTrack](#).

Definition at line 19 of file Tracker.cpp.

References `firstFrame`, `lastFrame`, `N`, `nDims`, `nFeatures`, `nFrames`, `trackAssignment`, and `tracks`.

Referenced by `tracker::LAPTrack::getStats()`.

Here is the caller graph for this function:



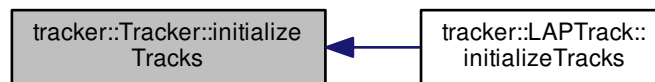
7.5.4.3 `void tracker::Tracker::initializeTracks (const IVecT & frameldx_, const MatT & position_, const MatT & SE_position_)`
`[virtual]`

Reimplemented in [tracker::LAPTrack](#).

Definition at line 33 of file Tracker.cpp.

Referenced by `tracker::LAPTrack::initializeTracks()`.

Here is the caller graph for this function:



7.5.4.4 `void tracker::Tracker::initializeTracks (const IVecT & frameldx_, const MatT & position_, const MatT & SE_position_, const MatT & feature_, const MatT & SE_feature_)`
`[virtual]`

Reimplemented in [tracker::LAPTrack](#).

Definition at line 39 of file Tracker.cpp.

References `feature`, `firstFrame`, `frameldx`, `frameLocIdx`, `lastFrame`, `N`, `nDims`, `nFeatures`, `nFrameLocs`, `nFrames`, `position`, `SE_feature`, `SE_position`, `trackAssignment`, and `tracks`.

7.5.4.5 `void tracker::Tracker::printTracks () const`

Definition at line 126 of file Tracker.cpp.

References `frameldx`, and `tracks`.

7.5.5 Member Data Documentation

7.5.5.1 `MatT` `tracker::Tracker::feature`

Definition at line 66 of file Tracker.h.

Referenced by `tracker::LAPTrack::computeF2FCostMat()`, `tracker::LAPTrack::computeGapCloseMatrix()`, and `initializeTracks()`.

7.5.5.2 IdxT tracker::Tracker::firstFrame = 0

Definition at line 68 of file Tracker.h.

Referenced by tracker::LAPTrack::checkFrameIdxs(), tracker::LAPTrack::computeF2FCostMat(), tracker::LAPTrack::computeGapCloseMatrix(), tracker::LAPTrack::debugF2F(), getStats(), initializeTracks(), and tracker::LAPTrack::linkF2F().

7.5.5.3 IVecT tracker::Tracker::frameIdx

Definition at line 63 of file Tracker.h.

Referenced by tracker::LAPTrack::checkFrameIdxs(), tracker::LAPTrack::computeGapCloseMatrix(), initializeTracks(), and printTracks().

7.5.5.4 IVecFieldT tracker::Tracker::frameLocIdx

Definition at line 74 of file Tracker.h.

Referenced by tracker::LAPTrack::computeF2FCostMat(), tracker::LAPTrack::debugF2F(), initializeTracks(), and tracker::LAPTrack::linkF2F().

7.5.5.5 IdxT tracker::Tracker::lastFrame = 0

Definition at line 69 of file Tracker.h.

Referenced by tracker::LAPTrack::checkFrameIdxs(), tracker::LAPTrack::computeGapCloseMatrix(), tracker::LAPTrack::debugF2F(), getStats(), initializeTracks(), and tracker::LAPTrack::linkF2F().

7.5.5.6 const Tracker::FloatT tracker::Tracker::log2pi = log(2*arma::Datum<Tracker::FloatT>::pi) [static], [protected]

Definition at line 92 of file Tracker.h.

Referenced by tracker::LAPTrack::computeF2FCostMat(), and tracker::LAPTrack::computeGapCloseMatrix().

7.5.5.7 IdxT tracker::Tracker::N = 0

Definition at line 60 of file Tracker.h.

Referenced by getStats(), and initializeTracks().

7.5.5.8 IdxT tracker::Tracker::nDims = 0

Definition at line 61 of file Tracker.h.

Referenced by tracker::LAPTrack::computeF2FCostMat(), tracker::LAPTrack::computeGapCloseMatrix(), getStats(), and initializeTracks().

7.5.5.9 **IdxT** `tracker::Tracker::nFeatures = 0`

Definition at line 62 of file Tracker.h.

Referenced by `tracker::LAPTrack::computeF2FCostMat()`, `tracker::LAPTrack::computeGapCloseMatrix()`, `getStats()`, and `initializeTracks()`.

7.5.5.10 **IVecT** `tracker::Tracker::nFrameLocs`

Definition at line 73 of file Tracker.h.

Referenced by `tracker::LAPTrack::computeF2FCostMat()`, `tracker::LAPTrack::debugF2F()`, `initializeTracks()`, and `tracker::LAPTrack::linkF2F()`.

7.5.5.11 **IdxT** `tracker::Tracker::nFrames = 0`

Definition at line 70 of file Tracker.h.

Referenced by `getStats()`, `initializeTracks()`, and `tracker::LAPTrack::linkF2F()`.

7.5.5.12 **MatT** `tracker::Tracker::position`

Definition at line 64 of file Tracker.h.

Referenced by `tracker::LAPTrack::computeF2FCostMat()`, `tracker::LAPTrack::computeGapCloseMatrix()`, and `initializeTracks()`.

7.5.5.13 **MatT** `tracker::Tracker::SE_feature`

Definition at line 67 of file Tracker.h.

Referenced by `tracker::LAPTrack::computeF2FCostMat()`, `tracker::LAPTrack::computeGapCloseMatrix()`, and `initializeTracks()`.

7.5.5.14 **MatT** `tracker::Tracker::SE_position`

Definition at line 65 of file Tracker.h.

Referenced by `tracker::LAPTrack::computeF2FCostMat()`, `tracker::LAPTrack::computeGapCloseMatrix()`, and `initializeTracks()`.

7.5.5.15 **IVecT** `tracker::Tracker::trackAssignment` `[protected]`

Definition at line 93 of file Tracker.h.

Referenced by `tracker::LAPTrack::closeGaps()`, `getStats()`, `initializeTracks()`, and `tracker::LAPTrack::linkF2F()`.

7.5.5.16 TrackVecT tracker::Tracker::tracks

Definition at line 77 of file Tracker.h.

Referenced by tracker::LAPTrack::checkFrameIdxs(), tracker::LAPTrack::closeGaps(), tracker::LAPTrack::computeGapCloseMatrix(), tracker::LAPTrack::debugCloseGaps(), getStats(), initializeTracks(), tracker::LAPTrack::linkF2F(), and printTracks().

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

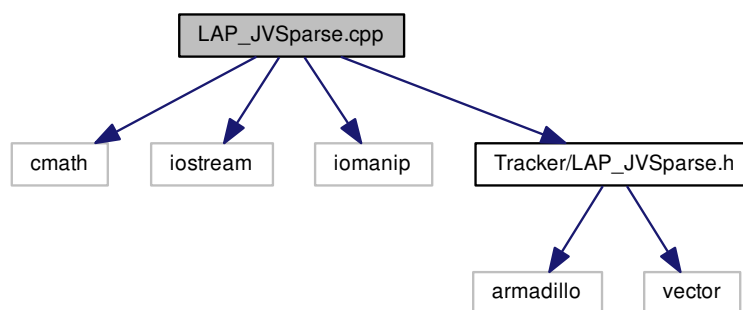
- [Tracker.h](#)
- [Tracker.cpp](#)

8 File Documentation

8.1 LAP_JVSpase.cpp File Reference

The member definitions for the LAP Jonker Volgenant algorithm.

```
#include <cmath>
#include <iostream>
#include <iomanip>
#include "Tracker/LAP_JVSpase.h"
Include dependency graph for LAP_JVSpase.cpp:
```



Namespaces

- [tracker](#)

8.1.1 Detailed Description

The member definitions for the LAP Jonker Volgenant algorithm.

Author

Mark J. Olah (mjo at cs.unm.edu)

Date

05-2015 This is a modern dense/sparse C++ implementation of Jonker Volgenant algoirthm using armadillo and presenting C++ and Matlab interface.

Adapted from text of Jonker and Volgenant. Computing 38, 324-340 (1986)

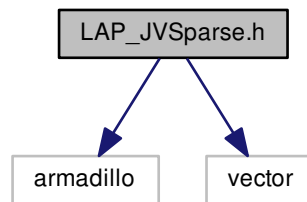
8.2 LAP_JVSpase.h File Reference

The class declaration for the LAP Jonker Volgenant algorithm.

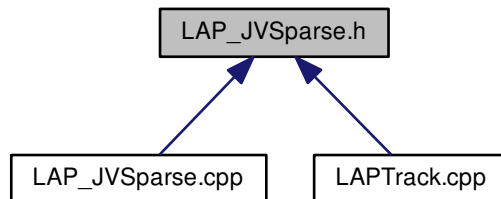
```
#include <armadillo>
```

```
#include <vector>
```

Include dependency graph for LAP_JVSpase.h:



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



Classes

- class [tracker::LAP_JV Sparse< FloatT >](#)

Namespaces

- [tracker](#)

8.2.1 Detailed Description

The class declaration for the LAP Jonker Volgenant algorithm.

Author

Mark J. Olah (mjo@cs.unm.edu)

Date

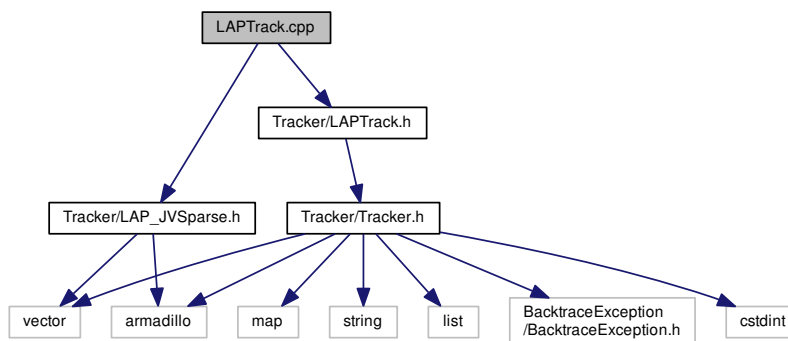
05-2015 This is a modern dense/sparse C++ implementation of Jonker Volgenant algoirthm using armadillo and presenting C++ and Matlab interface.

Adapted from text of Jonker and Volgenant. Computing 38, 324-340 (1986)

8.3 LAPTrack.cpp File Reference

The member definitions for LAPTrack.

```
#include "Tracker/LAPTrack.h"
#include "Tracker/LAP_JV Sparse.h"
Include dependency graph for LAPTrack.cpp:
```



Namespaces

- [tracker](#)

8.3.1 Detailed Description

The member definitions for LAPTrack.

Author

Mark J. Olah (mjo at cs.unm.edu)

Date

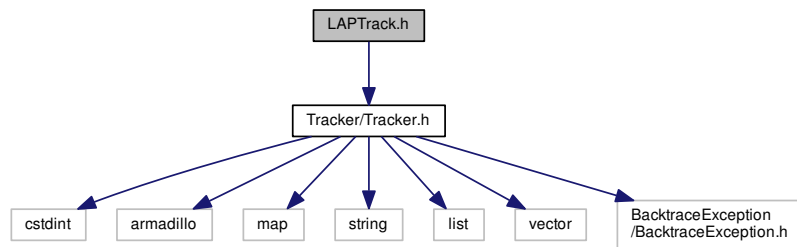
04-2015

8.4 LAPTrack.h File Reference

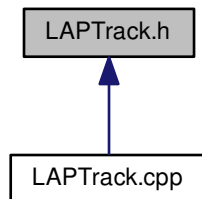
The class declaration and inline and templated functions for LAPTrack.

```
#include "Tracker/Tracker.h"
```

Include dependency graph for LAPTrack.h:



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



Classes

- class [tracker::LAPTrack](#)

Namespaces

- [tracker](#)

8.4.1 Detailed Description

The class declaration and inline and templated functions for LAPTrack.

Author

Mark J. Olah (mjo@cs.unm.edu)

Date

02-2015 A simple LAP/Jaquuman based tracker

8.5 README.md File Reference

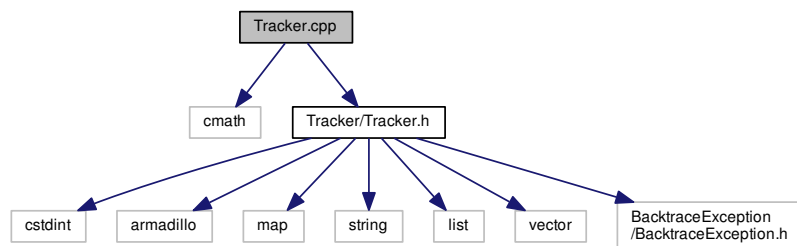
8.6 Tracker.cpp File Reference

The member definitions for Tracker.

```
#include <cmath>
```

```
#include "Tracker/Tracker.h"
```

Include dependency graph for Tracker.cpp:



Namespaces

- [tracker](#)

8.6.1 Detailed Description

The member definitions for Tracker.

Author

Mark J. Olah (mjo at cs.unm.edu)

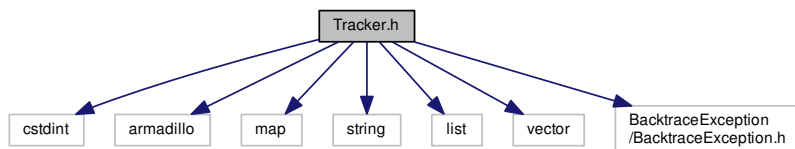
Date

04-2015

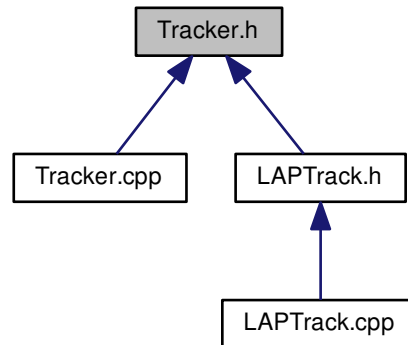
8.7 Tracker.h File Reference

The class declaration and inline and templated functions for Tracker.

```
#include <cstdint>
#include <armadillo>
#include <map>
#include <string>
#include <list>
#include <vector>
#include "BacktraceException/BacktraceException.h"
Include dependency graph for Tracker.h:
```



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



Classes

- struct [tracker::ParameterValueError](#)
Parameter value is not valid.
- struct [tracker::LogicalError](#)
Parameter value is not valid.
- class [tracker::Tracker](#)

Namespaces

- [tracker](#)

Typedefs

- using [tracker::TrackerError](#) = [backtrace_exception::BacktraceException](#)

8.7.1 Detailed Description

The class declaration and inline and templated functions for Tracker.

Author

Mark J. Olah (mjo@cs.unm.edu)

Date

02-2015 The base class for all Tracking models

Insted of templating on the FloatT type, which is problematic for inheritance hierarchies of templated base classes. Instead wuse a typedef to allow configuration of use with either float/double. Default is double.

Index

~Tracker
 [tracker::Tracker, 29](#)

birthFrameIdx
 [tracker::LAPTrack, 17](#)

checkCosts
 [tracker::LAP_JVSpase, 4](#)

checkFrameIdxs
 [tracker::LAPTrack, 12](#)

checkSolution
 [tracker::LAP_JVSpase, 4](#)

closeGaps
 [tracker::LAPTrack, 12](#)

computeCost
 [tracker::LAP_JVSpase, 4](#)

computeF2FCostMat
 [tracker::LAPTrack, 13](#)

computeGapCloseMatrix
 [tracker::LAPTrack, 13](#)

cost_epsilon
 [tracker::LAPTrack, 17](#)

D
 [tracker::LAPTrack, 18](#)

debugCloseGaps
 [tracker::LAPTrack, 14](#)

debugF2F
 [tracker::LAPTrack, 14](#)

F2F_LINKED
 [tracker::LAPTrack, 12](#)

feature
 [tracker::LAPTrack, 18](#)
 [tracker::Tracker, 30](#)

featureVar
 [tracker::LAPTrack, 18](#)

firstFrame
 [tracker::LAPTrack, 18](#)
 [tracker::Tracker, 30](#)

FloatT
 [tracker::LAPTrack, 10](#)
 [tracker::Tracker, 27](#)

frameBirthStartIdx
 [tracker::LAPTrack, 18](#)

frameIdx
 [tracker::LAPTrack, 18](#)
 [tracker::Tracker, 31](#)

frameLocIdx
 [tracker::LAPTrack, 18](#)
 [tracker::Tracker, 31](#)

GAPS_CLOSED
 [tracker::LAPTrack, 12](#)

generateTracks
 [tracker::LAPTrack, 15](#)
 [tracker::Tracker, 29](#)

getStats
 [tracker::LAPTrack, 15](#)
 [tracker::Tracker, 29](#)

IMatT
 [tracker::LAPTrack, 10](#)
 [tracker::Tracker, 27](#)

IVecFieldT
 [tracker::LAPTrack, 10](#)
 [tracker::Tracker, 28](#)

IVecT
 [tracker::LAPTrack, 10](#)
 [tracker::Tracker, 28](#)

IdxT
 [tracker::LAPTrack, 10](#)
 [tracker::Tracker, 27](#)

IndexVectorT
 [tracker::LAPTrack, 10](#)
 [tracker::Tracker, 27](#)

initializeTracks
 [tracker::LAPTrack, 16](#)
 [tracker::Tracker, 29, 30](#)

koff
 [tracker::LAPTrack, 18](#)

kon
 [tracker::LAPTrack, 19](#)

LAP_JVSpase.cpp, [33](#)
LAP_JVSpase.h, [34](#)
LAPTrack
 [tracker::LAPTrack, 12](#)
LAPTrack.cpp, [35](#)
LAPTrack.h, [36](#)

lastFrame
 [tracker::LAPTrack, 19](#)
 [tracker::Tracker, 31](#)

linkF2F
 [tracker::LAPTrack, 16](#)

log1mkoff
 [tracker::LAPTrack, 19](#)

log1mkon
 [tracker::LAPTrack, 19](#)

log2pi
 [tracker::LAPTrack, 19](#)
 [tracker::Tracker, 31](#)

LogicalError

- tracker::LogicalError, [23](#)
- logkoff
 - tracker::LAPTrack, [19](#)
- logkon
 - tracker::LAPTrack, [19](#)
- logrho
 - tracker::LAPTrack, [19](#)
- MatT
 - tracker::LAPTrack, [11](#)
 - tracker::Tracker, [28](#)
- maxFeatureDisplacementSigma
 - tracker::LAPTrack, [20](#)
- maxGapCloseFrames
 - tracker::LAPTrack, [20](#)
- maxPositionDisplacementSigma
 - tracker::LAPTrack, [20](#)
- maxSpeed
 - tracker::LAPTrack, [20](#)
- minCost
 - tracker::LAPTrack, [20](#)
- minFinalTrackLength
 - tracker::LAPTrack, [20](#)
- minGapCloseTrackLength
 - tracker::LAPTrack, [20](#)
- N
 - tracker::LAPTrack, [20](#)
 - tracker::Tracker, [31](#)
- nDims
 - tracker::LAPTrack, [21](#)
 - tracker::Tracker, [31](#)
- nFeatures
 - tracker::LAPTrack, [21](#)
 - tracker::Tracker, [31](#)
- nFrameLocs
 - tracker::LAPTrack, [21](#)
 - tracker::Tracker, [32](#)
- nFrames
 - tracker::LAPTrack, [21](#)
 - tracker::Tracker, [32](#)
- ParameterValueError
 - tracker::ParameterValueError, [25](#)
- ParamT
 - tracker::LAPTrack, [11](#)
 - tracker::Tracker, [28](#)
- position
 - tracker::LAPTrack, [21](#)
 - tracker::Tracker, [32](#)
- printTracks
 - tracker::LAPTrack, [17](#)
 - tracker::Tracker, [30](#)
- README.md, [37](#)
- rho
 - tracker::LAPTrack, [21](#)
- SE_feature
 - tracker::LAPTrack, [21](#)
 - tracker::Tracker, [32](#)
- SE_position
 - tracker::LAPTrack, [21](#)
 - tracker::Tracker, [32](#)
- solve
 - tracker::LAP_JVSpase, [5](#)
- solveLAP_orig
 - tracker::LAP_JVSpase, [5](#)
- SpMatT
 - tracker::LAPTrack, [11](#)
- state
 - tracker::LAPTrack, [22](#)
- StateT
 - tracker::LAPTrack, [12](#)
- trackAssignment
 - tracker::LAPTrack, [22](#)
 - tracker::Tracker, [32](#)
- TrackVecT
 - tracker::LAPTrack, [11](#)
 - tracker::Tracker, [28](#)
- Tracker
 - tracker::Tracker, [29](#)
- tracker, [3](#)
 - TrackerError, [3](#)
- Tracker.cpp, [37](#)
- Tracker.h, [38](#)
- tracker::LAP_JVSpase
 - checkCosts, [4](#)
 - checkSolution, [4](#)
 - computeCost, [4](#)
 - solve, [5](#)
 - solveLAP_orig, [5](#)
- tracker::LAP_JVSpase< FloatT >, [3](#)
- tracker::LAPTrack, [7](#)
 - birthFrameIdx, [17](#)
 - checkFrameIdxs, [12](#)
 - closeGaps, [12](#)
 - computeF2FCostMat, [13](#)
 - computeGapCloseMatrix, [13](#)
 - cost_epsilon, [17](#)
 - D, [18](#)
 - debugCloseGaps, [14](#)
 - debugF2F, [14](#)
 - F2F_LINKED, [12](#)
 - feature, [18](#)
 - featureVar, [18](#)
 - firstFrame, [18](#)
 - FloatT, [10](#)
 - frameBirthStartIdx, [18](#)

- frameIdx, 18
- frameLocIdx, 18
- GAPS_CLOSED, 12
- generateTracks, 15
- getStats, 15
- IMatT, 10
- IVecFieldT, 10
- IVecT, 10
- IdxT, 10
- IndexVectorT, 10
- initializeTracks, 16
- koff, 18
- kon, 19
- LAPTrack, 12
- lastFrame, 19
- linkF2F, 16
- log1mkoff, 19
- log1mkon, 19
- log2pi, 19
- logkoff, 19
- logkon, 19
- logrho, 19
- MatT, 11
- maxFeatureDisplacementSigma, 20
- maxGapCloseFrames, 20
- maxPositionDisplacementSigma, 20
- maxSpeed, 20
- minCost, 20
- minFinalTrackLength, 20
- minGapCloseTrackLength, 20
- N, 20
- nDims, 21
- nFeatures, 21
- nFrameLocs, 21
- nFrames, 21
- ParamT, 11
- position, 21
- printTracks, 17
- rho, 21
- SE_feature, 21
- SE_position, 21
- SpMatT, 11
- state, 22
- StateT, 12
- trackAssignment, 22
- TrackVecT, 11
- tracks, 22
- TrackT, 11
- UMatT, 11
- UNTRACKED, 12
- UVecT, 11
- VecParamT, 11
- VecT, 11
- tracker::LogicalError, 22
- LogicalError, 23
- tracker::ParameterValueError, 24
- ParameterValueError, 25
- tracker::Tracker, 25
- ~Tracker, 29
- feature, 30
- firstFrame, 30
- FloatT, 27
- frameIdx, 31
- frameLocIdx, 31
- generateTracks, 29
- getStats, 29
- IMatT, 27
- IVecFieldT, 28
- IVecT, 28
- IdxT, 27
- IndexVectorT, 27
- initializeTracks, 29, 30
- lastFrame, 31
- log2pi, 31
- MatT, 28
- N, 31
- nDims, 31
- nFeatures, 31
- nFrameLocs, 32
- nFrames, 32
- ParamT, 28
- position, 32
- printTracks, 30
- SE_feature, 32
- SE_position, 32
- trackAssignment, 32
- TrackVecT, 28
- Tracker, 29
- tracks, 32
- TrackT, 28
- VecParamT, 28
- VecT, 28
- TrackerError
- tracker, 3
- tracks
- tracker::LAPTrack, 22
- tracker::Tracker, 32
- TrackT
- tracker::LAPTrack, 11
- tracker::Tracker, 28
- UMatT
- tracker::LAPTrack, 11
- UNTRACKED
- tracker::LAPTrack, 12
- UVecT
- tracker::LAPTrack, 11
- VecParamT

tracker::LAPTrack, [11](#)

tracker::Tracker, [28](#)

VecT

tracker::LAPTrack, [11](#)

tracker::Tracker, [28](#)