Motion RF

Generated by Doxygen 1.8.1

Wed Jun 5 2013 13:57:27

Contents

1 Class Index **1**

1.1 Class List . . . 1

2 Class Documentation **3**

2.1 FeaturesMotion Class Reference . . . 3

2.1.1 Detailed Description . . . 3

2.1.2 Member Function Documentation . . . 4

2.1.2.1 flip . . . 4

2.1.2.2 push_backHisto . . . 4

2.1.2.3 push_backImages . . . 4

2.1.2.4 push_backVelX . . . 4

2.1.2.5 push_backVelY . . . 4

2.1.2.6 velX . . . 4

2.1.2.7 velX . . . 4

2.2 MotionLeafNode Struct Reference . . . 4

2.2.1 Constructor & Destructor Documentation . . . 5

2.2.1.1 MotionLeafNode . . . 5

2.2.2 Member Function Documentation . . . 5

2.2.2.1 appearanceProb . . . 5

2.2.2.2 motionProb . . . 5

2.2.2.3 readLeaf . . . 5

2.2.2.4 readLeafBin . . . 5

2.2.2.5 showLeafBin . . . 5

2.2.2.6 showLeafTxt . . . 5

2.3 MotionPatch< T, F > Class Template Reference . . . 5

2.3.1 Member Enumeration Documentation . . . 7

2.3.1.1 HistoType . . . 7

2.3.2 Member Function Documentation . . . 7

2.3.2.1 computeHistograms . . . 7

2.3.2.2 computeHistograms . . . 7

2.3.2.3 extractFeatures . . . 7

2.3.2.4 extractMotionAbsolute 7
2.3.2.5 extractMotionRelative 7
2.3.2.6 extractPatches 7
2.3.2.7 extractPatchesOF 7
2.3.2.8 findmatches 8
2.3.2.9 findThreshold 8
2.3.2.10 findThresholdAngle 8
2.3.2.11 getAngleHisto 8
2.3.2.12 getMagniHisto 8
2.3.2.13 getRansacAffineTransform 8
2.3.2.14 imagePairs 8
2.3.2.15 justFlow 8
2.3.2.16 loadPatches 8
2.3.2.17 motionW 8
2.3.2.18 pickRandomNames 9
2.3.2.19 reset 9
2.3.2.20 savePatches 9
2.3.2.21 showOF 9
2.3.2.22 warpOpenCV 9
2.3.2.23 warpSecond2First 9
2.4 MotionPatchFeature< F > Class Template Reference 9
2.4.1 Detailed Description 10
2.4.2 Constructor & Destructor Documentation 10
2.4.2.1 MotionPatchFeature 10
2.4.3 Member Function Documentation 10
2.4.3.1 histo 10
2.4.3.2 histo 10
2.4.3.3 histoCenter 10
2.4.3.4 image 10
2.4.3.5 motion 10
2.4.3.6 motion 11
2.4.3.7 motionW 11
2.4.3.8 motionW 11
2.5 MotionPuzzle< P > Class Template Reference 11
2.5.1 Member Function Documentation 12
2.5.1.1 approxKernel 12
2.5.1.2 checkConvergence 12
2.5.1.3 histInfo 12
2.5.1.4 initialPick 12
2.5.1.5 mostVotedFlow 12

2.5.1.6 perTreePredictions 12

2.5.1.7 pickApproxKernel 12

2.5.1.8 pickMean 12

2.5.1.9 proposePrediction 12

2.5.1.10 proposePredictionMax 13

2.5.1.11 proposePredictionOverlap 13

2.5.1.12 proposePredictionSum 13

2.5.1.13 selectPatches 13

2.5.1.14 showSamples 13

2.5.1.15 solve 13

2.5.1.16 threshold 13

2.6 MotionPuzzlePatch Struct Reference 13

2.6.1 Detailed Description 14

2.6.2 Constructor & Destructor Documentation 14

2.6.2.1 MotionPuzzlePatch 14

2.6.3 Member Function Documentation 14

2.6.3.1 appearanceProb 14

2.6.3.2 appearanceProb 14

2.6.3.3 pos2pt 14

2.7 MotionRF< L, M, T, F, U > Class Template Reference 15

2.7.1 Member Function Documentation 15

2.7.1.1 sigmaratio 15

2.7.1.2 trainForestTree 15

2.8 MotionRFdetector< L, M, T, F, U > Class Template Reference 15

2.8.1 Member Function Documentation 16

2.8.1.1 detectColor 16

2.8.1.2 detectPyramid 16

2.8.1.3 motionW 16

2.8.1.4 motionW 16

2.9 MotionTree< M, T, F, U > Class Template Reference 16

2.9.1 Constructor & Destructor Documentation 18

2.9.1.1 MotionTree 18

2.9.2 Member Function Documentation 18

2.9.2.1 dotProd 18

2.9.2.2 getFreqAngle 18

2.9.2.3 getFreqMagni 19

2.9.2.4 grow 19

2.9.2.5 growTree 19

2.9.2.6 initDataSizes 19

2.9.2.7 leafApprox 19

2.9.2.8 leafMean 19
2.9.2.9 makeLeaf 19
2.9.2.10 measureSet 19
2.9.2.11 mpick 20
2.9.2.12 mpick 20
2.9.2.13 optimizeTest 20
2.9.2.14 patchAppearanceSim 20
2.9.2.15 patchApprox 20
2.9.2.16 patchDist2Mean 20
2.9.2.17 performSplit 20
2.9.2.18 readNode 20
2.9.2.19 readNodeBin 21
2.9.2.20 readTree 21
2.9.2.21 readTreeBin 21
2.9.2.22 saveTree 21
2.9.2.23 saveTreeBin 21
2.9.2.24 saveTreeTxt 21
2.9.2.25 setFreq 21
2.9.2.26 showPickedSplit 21
2.9.2.27 showSamples 21
2.9.2.28 showSamples 21
2.9.2.29 splitApproxKernel 22
2.9.2.30 splitDistance2mean 22
2.9.2.31 stopCosSimilarity 22
2.9.2.32 stopEuclDist 22
2.10 RunMotionRF< L, M, T, F, U > Class Template Reference 22
2.10.1 Member Enumeration Documentation 23
2.10.1.1 MODE 23
2.10.2 Member Function Documentation 23
2.10.2.1 batchExtractOF 23
2.10.2.2 batchTest 24
2.10.2.3 batchTrain 24
2.10.2.4 extract 24
2.10.2.5 extractOF 24
2.10.2.6 generateExtractCommands 24
2.10.2.7 generateTestCommands 24
2.10.2.8 jobrunnerExtractOF 24
2.10.2.9 jobrunnerTest 24
2.10.2.10 jobrunnerTrain 24
2.10.2.11 motionWidth 25

2.10.2.12 motionWidth 25
2.10.2.13 run 25
2.10.2.14 runExtract 25
2.10.2.15 runExtract1 25
2.10.2.16 runExtractOF 25
2.10.2.17 runTest 25
2.10.2.18 runTest1 25
2.10.2.19 runTrain 25
2.10.2.20 runTrain1 26
2.10.2.21 test 26

Chapter 1

Class Index

1.1 Class List

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

FeaturesMotion 3
MotionLeafNode 4
MotionPatch< T, F > 5
MotionPatchFeature< F > 9
MotionPuzzle< P > 11
MotionPuzzlePatch 13
MotionRF< L, M, T, F, U > 15
MotionRFdetector< L, M, T, F, U > 15
MotionTree< M, T, F, U > 16
RunMotionRF< L, M, T, F, U > 22

Chapter 2

Class Documentation

2.1 FeaturesMotion Class Reference

```
#include <MotionPatch.h>
```

Public Member Functions

- **FeaturesMotion** (const std::vector< cv::Mat > &velX, const std::vector< cv::Mat > &velY, const std::vector< cv::Mat > &labelImg, const std::vector< std::vector< IplImage * > > &vImg, const std::vector< cv::Mat > &images, const std::vector< std::vector< cv::Mat > > &histo)
- FeaturesMotion * flip () const
- void push_backVelX (const cv::Mat &velX)
- void push_backVelY (const cv::Mat &velY)
- void push_backImages (const cv::Mat &images)
- void push_backHisto (const std::vector< cv::Mat > &hist)
- virtual unsigned **size** () const
- std::vector< cv::Mat > velX () const
- cv::Mat **velX** (unsigned pos) const
- std::vector< cv::Mat > **velY** () const
- cv::Mat **velY** (unsigned pos) const
- std::vector< cv::Mat > **images** () const
- std::vector< std::vector < cv::Mat > > **histo** () const
- std::vector< cv::Mat > **histo** (unsigned pos) const
- cv::Mat **images** (unsigned pos) const
- std::vector< float > **histinfo** () const
- void velX (const std::vector< cv::Mat > &velX)
- void **velX** (unsigned pos, const cv::Mat velX)
- void **velY** (const std::vector< cv::Mat > &velY)
- void **velY** (unsigned pos, const cv::Mat velY)
- void **images** (const std::vector< cv::Mat > &images)
- void **images** (unsigned pos, const cv::Mat images)
- void **histinfo** (std::vector< float > &histinfo)
- void **histo** (const std::vector< std::vector< cv::Mat > > &histo)

2.1.1 Detailed Description

To keep inside all the vectors we need.

2.1.2 Member Function Documentation

2.1.2.1 FeaturesMotion∗ FeaturesMotion::flip ( ) const [inline]

Flips the motion patch.

2.1.2.2 void FeaturesMotion::push_backHisto (const std::vector< cv::Mat > & *hist*) [inline]

Adds a matrix to the histogram of matrices for angle.

2.1.2.3 void FeaturesMotion::push_backImages (const cv::Mat & *images*) [inline]

Adds a matrix to the vector of images.

2.1.2.4 void FeaturesMotion::push_backVelX (const cv::Mat & *velX*) [inline]

Adds a matrix to the vector of velocity matrices on X.

2.1.2.5 void FeaturesMotion::push_backVelY (const cv::Mat & *velY*) [inline]

Adds a matrix to the vector of velocity matrices on Y.

2.1.2.6 std::vector<cv::Mat> FeaturesMotion::velX ( ) const [inline]

Getter for class members.

2.1.2.7 void FeaturesMotion::velX (const std::vector< cv::Mat > & *velX*) [inline]

Setter for class members.

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

- MotionPatch.h

2.2 MotionLeafNode Struct Reference

Public Member Functions

- **MotionLeafNode** (const char ∗path2models, long unsigned leafid, unsigned treeid, bool binary)
- virtual void readLeaf (const char ∗path2models, long unsigned leafid, unsigned treeid)
- void readLeafBin (const char ∗path2models, long unsigned leafid, unsigned treeid)
- virtual void showLeafTxt (const char ∗path2models, long unsigned leafid, unsigned treeid)
- virtual void showLeafBin (const char ∗path2models, long unsigned leafid, unsigned treeid)
- float motionProb () const
- float **appearanceProb** () const
- cv::Mat ∗ **vMotion** () const
- cv::Mat ∗ **vAppearance** () const
- float **vMotion** (cv::Point &pt) const
- std::vector< cv::Mat > **vHistos** () const
- void appearanceProb (float appearanceProb)
- void **motionProb** (float motionProb)

- void **vMotion** (const cv::Mat ∗vMotion)
- void **vAppearance** (const cv::Mat ∗vAppearance)
- void **vHistos** (const std::vector< cv::Mat > &vHistos)
- MotionLeafNode (MotionLeafNode const &rhs)

2.2.1 Constructor & Destructor Documentation

2.2.1.1 MotionLeafNode::MotionLeafNode (**MotionLeafNode** const & *rhs*) `[inline]`

Copy constructors for trees (to put them in the forest). The assignment operator is done.

2.2.2 Member Function Documentation

2.2.2.1 void MotionLeafNode::appearanceProb (float *appearanceProb*) `[inline]`

Setters for the class members.

2.2.2.2 float MotionLeafNode::motionProb ( ) const `[inline]`

Getters for the class members.

2.2.2.3 virtual void MotionLeafNode::readLeaf (const char ∗ *path2models,* long unsigned *leafid,* unsigned *treeid*) `[inline]`, `[virtual]`

Reads the leaf from a regular file.

2.2.2.4 void MotionLeafNode::readLeafBin (const char ∗ *path2models,* long unsigned *leafid,* unsigned *treeid*) `[inline]`

Reads the leaf from a binary file.

2.2.2.5 virtual void MotionLeafNode::showLeafBin (const char ∗ *path2models,* long unsigned *leafid,* unsigned *treeid*) `[inline]`, `[virtual]`

Writes the leaf info into an opened binary file.

2.2.2.6 virtual void MotionLeafNode::showLeafTxt (const char ∗ *path2models,* long unsigned *leafid,* unsigned *treeid*) `[inline]`, `[virtual]`

Writes the leaf info into an opened file.

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

- MotionTree.h

2.3 MotionPatch< T, F > Class Template Reference

Public Types

- enum HistoType {
CENTER, **RANDOM**, **CENTER_RANDOM**, **MEAN_DIFF**,
APPROX_MAGNI_KERNEL, **APPROX_ANGLE_KERNEL** }
- enum **Algorithm** { **Farneback**, **LucasKanade**, **HornSchunck** }

Public Member Functions

- **MotionPatch** (CvRNG *pRNG, unsigned patchW, unsigned patchH, unsigned noCls, unsigned labW, unsigned labH, unsigned trainSize, unsigned noPatches, unsigned consideredCls, unsigned balance, unsigned step, unsigned motionW=0, unsigned motionH=0, bool warpping=false, bool ofThresh=true, unsigned histotype=4, float sigmaratio=0.5, unsigned bins=25)
- int warpOpenCV (cv::Mat &curr, cv::Mat &next)
- void pickRandomNames (const std::string &featpath, const std::vector< std::string > &folders, const std::string &ext, const std::string &imgpath, std::vector< unsigned > &shuffle)
- int extractMotionRelative (const std::vector< std::string > &tuples, Algorithm algo, std::string &featpath, bool save=false)
- void computeHistograms (unsigned pos)
- void extractMotionAbsolute (const std::vector< std::string > &tuples, Algorithm algo, std::string &featpath, bool save=false)
- void extractPatchesOF (const std::string &imgpath, std::string &featpath, const std::vector< std::string > &vFilenames, const std::string &ext, const MotionPatch< T, F >::Algorithm &algo, bool justimages=false)
- int warpSecond2First (cv::Mat &curr, cv::Mat &next)
- void findmatches (const cv::Mat &points1, const cv::Mat &points2, const cv::Mat &img1, const cv::Mat &img2, double minDist, double maxDist, cv::Mat &outpoints1, cv::Mat &outpoints2)
- cv::Mat getRansacAffineTransform (const cv::Mat &points1, const cv::Mat &points2, const cv::Mat &img1, const cv::Mat &img2, double limit, int &isgood)
- virtual void reset ()
- virtual void savePatches (const std::string &path2feat, unsigned pos)
- virtual void loadPatches (const std::string &path2feat, bool showWhere=false)
- virtual void extractFeatures (IplImage *img, const std::string &path2feat, bool showWhere=false)
- virtual void extractPatches (const std::string &imgpath, const std::string &labpath, std::string &featpath, const std::vector< std::string > &vFilenames, const std::map< cv::Vec3b, unsigned, vec3bCompare > &classinfo, const std::string &labTerm, const std::string &ext, bool justimages=false)
- void getAngleHisto ()
- void getMagniHisto ()
- void computeHistograms ()
- unsigned motionW () const
- unsigned **motionH** () const
- unsigned **maximsize** () const
- std::vector< std::vector < std::string > > **imagePairs** () const
- bool **relativeOF** () const
- bool **display** () const
- bool **storefeat** () const
- unsigned **bins** () const
- void imagePairs (const std::vector< std::vector< std::string > > &imagePairs)
- void **motionW** (unsigned motionW)
- void **motionH** (unsigned motionH)
- void **relativeOF** (bool relativeOF)
- void **display** (bool display)
- void **maximsize** (unsigned maximsize)
- void **storefeat** (bool storefeat)
- void **bins** (unsigned bins)

Static Public Member Functions

- static cv::Mat findThresholdAngle (const cv::Mat &valuesX, const cv::Mat &valuesY, float &minTr, float &maxTr)
- static void findThreshold (const cv::Mat &values, float &minTr, float &maxTr)
- static cv::Mat showOF (const cv::Mat &velX, const cv::Mat &velY, const cv::Mat &image, unsigned step=1, bool display=false, const std::string &winname="Of", const cv::Rect &roi=cv::Rect(0, 0, 0, 0))
- static cv::Mat justFlow (const cv::Mat &current, cv::Mat &next, Algorithm algo, std::string &featpath, unsigned motionSz, bool save, const std::string &imName)

2.3.1 Member Enumeration Documentation

2.3.1.1 template<class T, class F> enum MotionPatch::HistoType

On what do we evaluate the entropy for precomputhing histos.

2.3.2 Member Function Documentation

2.3.2.1 template<class T, class F> void MotionPatch< T, F >::computeHistograms (unsigned *pos*)

Compute histograms of angle magnitude.

2.3.2.2 template<class T , class F > void MotionPatch< T, F >::computeHistograms ( )

Compute histograms of angle magnitude.

2.3.2.3 template<class T , class F > void MotionPatch< T, F >::extractFeatures (IplImage ∗ *img,* const std::string & *path2feat,* bool *showWhere* = `false`) `[virtual]`

Computes features if not there for loading.

2.3.2.4 template<class T , class F > void MotionPatch< T, F >::extractMotionAbsolute (const std::vector< std::string > & *tuple,* Algorithm *algo,* std::string & *featpath,* bool *save* = `false`)

Load or extract the optical flow vector from two consecutive images.

2.3.2.5 template<class T , class F > int MotionPatch< T, F >::extractMotionRelative (const std::vector< std::string > & *tuple,* Algorithm *algo,* std::string & *featpath,* bool *save* = `false`)

Load or extract the optical flow vector from two pairs of consecutive images, and then take the difference of their OFs.

2.3.2.6 template<class T , class F > void MotionPatch< T, F >::extractPatches (const std::string & *imgpath,* const std::string & *labpath,* std::string & *featpath,* const std::vector< std::string > & *vFilenames,* const std::map< cv::Vec3b, unsigned, vec3bCompare > & *classinfo,* const std::string & *labTerm,* const std::string & *ext,* bool *justimages* = `false`) `[virtual]`

Extracts the feature patches but also the label patches. imgpath – path to the images labpath – path to labels ofpath – path to optical flow featpath – path to features vFilenames – vector of image names classinfo – mapping from pixel color to label ID labH – label patch height labW – label patch width

2.3.2.7 template<class T , class F > void MotionPatch< T, F >::extractPatchesOF (const std::string & *imgpath,* std::string & *featpath,* const std::vector< std::string > & *vFilenames,* const std::string & *ext,* const MotionPatch< T, F >::Algorithm & *algo,* bool *justimages* = `false`)

Extracts the feature patches but also the label patches.

Extracts the feature patches but also the label patches. imgpath – path to the images labpath – path to labels ofpath – path to optical flow featpath – path to features vFilenames – vector of image names classinfo – mapping from pixel color to label ID labH – label patch height labW – label patch width

2.3.2.8 template<class T , class F > void MotionPatch< T, F >::findmatches (const cv::Mat & *points1*, const cv::Mat & *points2*, const cv::Mat & *img1*, const cv::Mat & *img2*, double *minDist*, double *maxDist*, cv::Mat & *outpoints1*, cv::Mat & *outpoints2*)

Finds matches between a set of points Finds matches between a set of points

Finds matches between a set of points

2.3.2.9 template<class T , class F > void MotionPatch< T, F >::findThreshold (const cv::Mat & *values*, float & *minTr*, float & *maxTr*) [static]

Find threshold by cutting the histogram at 0.90.

2.3.2.10 template<class T , class F > cv::Mat MotionPatch< T, F >::findThresholdAngle (const cv::Mat & *valuesX*, const cv::Mat & *valuesY*, float & *minTr*, float & *maxTr*) [static]

Find threshold by cutting the histogram at 0.90.

2.3.2.11 template<class T , class F > void MotionPatch< T, F >::getAngleHisto ( )

Get the histogram of angles.

2.3.2.12 template<class T , class F > void MotionPatch< T, F >::getMagniHisto ( )

Get the histogram of magnitudes.

2.3.2.13 template<class T , class F > cv::Mat MotionPatch< T, F >::getRansacAffineTransform (const cv::Mat & *points1*, const cv::Mat & *points2*, const cv::Mat & *img1*, const cv::Mat & *img2*, double *limit*, int & *isgood*)

My own little sweet RANSAC.

2.3.2.14 template<class T, class F> void MotionPatch< T, F >::imagePairs (const std::vector< std::vector< std::string > > & *imagePairs*) [inline]

Setter for the class members.

2.3.2.15 template<class T , class F > cv::Mat MotionPatch< T, F >::justFlow (const cv::Mat & *current*, cv::Mat & *next*, Algorithm *algo*, std::string & *featpath*, unsigned *motionSz*, bool *save*, const std::string & *imName*) [static]

Just extracts OF for a pair of images using a given algorithm (hardcoded parameters).

2.3.2.16 template<class T , class F > void MotionPatch< T, F >::loadPatches (const std::string & *path2feat*, bool *showWhere* = false) [virtual]

Loads the labels and the image features — 1 file per image.

2.3.2.17 template<class T, class F> unsigned MotionPatch< T, F >::motionW ( ) const [inline]

Getter for the class members.

2.3.2.18 template<class T , class F > void MotionPatch< T, F >::pickRandomNames (const std::string & *featpath,* const std::vector< std::string > & *folders,* const std::string & *ext,* const std::string & *imgpath,* std::vector< unsigned > & *shuffle*)

Randomly picks a subset of the images names to be used for training – pairs of 2 images for OF.

2.3.2.19 template<class T , class F > void MotionPatch< T, F >::reset ( ) `[virtual]`

Resets the class members to add new patches.

2.3.2.20 template<class T , class F > void MotionPatch< T, F >::savePatches (const std::string & *path2feat,* unsigned *pos*) `[virtual]`

Saves the labels and the image features — for each image make one file.

2.3.2.21 template<class T , class F > cv::Mat MotionPatch< T, F >::showOF (const cv::Mat & *velX,* const cv::Mat & *velY,* const cv::Mat & *image,* unsigned *step* = `1`, bool *display* = `false`, const std::string & *winname* = `"Of"`, const cv::Rect & *roi* = `cv::Rect(0,0,0,0)`) `[static]`

Showing OF vectors (for check only).

2.3.2.22 template<class T , class F > int MotionPatch< T, F >::warpOpenCV (cv::Mat & *curr,* cv::Mat & *next*)

Finds interest points and warps the second image to the first image.

2.3.2.23 template<class T , class F > int MotionPatch< T, F >::warpSecond2First (cv::Mat & *curr,* cv::Mat & *next*)

Finds interest points and warps the second image to the first image.

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

- MotionPatch.h
- MotionPatch.cpp

2.4 MotionPatchFeature< F > Class Template Reference

```
#include <MotionPatch.h>
```

Public Types

- enum **HistoType** {
  CENTER, **RANDOM**, **CENTER_RANDOM**, **MEAN_DIFF**,
  APPROX_MAGNI_KERNEL, **APPROX_ANGLE_KERNEL** }

Public Member Functions

- **MotionPatchFeature** (unsigned featW, unsigned featH, unsigned labW, unsigned labH, unsigned imIndex, const cv::Point &point, unsigned motionW=0, unsigned motionH=0)
- cv::Mat ∗ motion (const F ∗feature) const
- void motion (const F ∗feature, cv::Mat ∗motionX, cv::Mat ∗motionY) const
- cv::Mat ∗ image (const F ∗feature) const

- std::vector< float > histoCenter (const F *feature) const
- std::vector< float > histo (const F *feature, const cv::Point pt) const
- std::vector< cv::Mat > histo (const F *feature) const
- unsigned motionW () const
- unsigned **motionH** () const
- void motionW (unsigned motionW)
- void **motionH** (unsigned motionH)
- MotionPatchFeature (MotionPatchFeature< F > const &rhs)
- MotionPatchFeature & **operator=** (MotionPatchFeature< F > const &rhs)

2.4.1 Detailed Description

template<class F>class MotionPatchFeature< F >

Patches are always relative to corner: top-left.

2.4.2 Constructor & Destructor Documentation

2.4.2.1 template<class F> MotionPatchFeature< F >::MotionPatchFeature (MotionPatchFeature< F > const & *rhs*) [inline]

Copy constructors the default ones are not good with IplImages

2.4.3 Member Function Documentation

2.4.3.1 template<class F > std::vector< float > MotionPatchFeature< F >::histo (const F * *feature,* const cv::Point *pt*) const [inline]

Gets the histogram at a random point in the patch.

2.4.3.2 template<class F > std::vector< cv::Mat > MotionPatchFeature< F >::histo (const F * *feature*) const [inline]

Gets the patch histogram at current point.

Gets the whole histogram patch.

2.4.3.3 template<class F > std::vector< float > MotionPatchFeature< F >::histoCenter (const F * *feature*) const [inline]

Gets the histogram at the current point position.

2.4.3.4 template<class F > cv::Mat * MotionPatchFeature< F >::image (const F * *feature*) const [inline]

Gets the image patch around the current pixel as a matrix.

2.4.3.5 template<class F > cv::Mat * MotionPatchFeature< F >::motion (const F * *feature*) const [inline]

Gets the motion patch around the current pixel as a vector.

2.4.3.6 template<class F > void MotionPatchFeature< F >::motion (const F ∗ *feature,* cv::Mat ∗ *motionX,* cv::Mat ∗ *motionY*) const [inline]

Gets the motion patches on x and y around the current pixel as a vector.

2.4.3.7 template<class F> unsigned MotionPatchFeature< F >::motionW ( ) const [inline]

Getters for the class members

2.4.3.8 template<class F> void MotionPatchFeature< F >::motionW (unsigned *motionW*) [inline]

Setters for the class members

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

- MotionPatch.h

2.5 MotionPuzzle< P > Class Template Reference

Static Public Member Functions

- static void threshold (cv::Mat &splitX, cv::Mat &splitY)
- static void solve (const std::vector< std::vector< P > > &patches, const cv::Size &featsize, unsigned motionW, unsigned motionH, cv::Mat &motionX, cv::Mat &motionY, cv::Mat &appear, MotionTreeClass::ENTROPY entropy, Puzzle< PuzzlePatch >::METHOD method, unsigned predCols, unsigned step, const std::string &imname, bool neighborhood=false, unsigned maxIter=75, bool display=false)
- static std::vector< P > initialPick (const std::vector< std::vector< P > > &candidates, unsigned motionW, unsigned motionH, MotionTreeClass::ENTROPY entropy, unsigned predCols, bool neighborhood, bool display=false)
- static void showSamples (const P &leaf, unsigned sampleW, unsigned sampleH)
- static cv::Mat proposePrediction (const std::vector< P > &candidates, const cv::Size &featsize, unsigned motionW, unsigned motionH, MotionTreeClass::ENTROPY entropy, cv::Mat &appearance)
- static cv::Mat proposePredictionSum (const std::vector< P > &candidates, const cv::Size &featsize, unsigned motionW, unsigned motionH, cv::Mat &appearance)
- static cv::Mat proposePredictionMax (const std::vector< P > &candidates, const cv::Size &featsize, unsigned motionW, unsigned motionH, cv::Mat &appearance)
- static cv::Mat proposePredictionOverlap (const std::vector< P > &candidates, const cv::Size &featsize, unsigned predCols, unsigned motionW, unsigned motionH, unsigned step, MotionTreeClass::ENTROPY entropy, cv::Mat &appearance)
- static std::vector< P > selectPatches (const cv::Mat &motion, const std::vector< std::vector< P > > &candidates, unsigned motionW, unsigned motionH)
- static bool checkConvergence (const cv::Mat &motion, const cv::Mat &prevMotion)
- static std::vector< float > histInfo (const std::vector< std::vector< P > > &candidates, unsigned motionW, unsigned motionH, MotionTreeClass::ENTROPY entropy)
- static std::vector< P > pickMean (const std::vector< std::vector< P > > &candidates, unsigned motionW, unsigned motionH, bool display)
- static std::vector< P > pickApproxKernel (const std::vector< std::vector< P > > &candidates, unsigned motionW, unsigned motionH, MotionTreeClass::ENTROPY entropy, unsigned predCols, bool neighborhood, bool display)
- static std::vector< float > approxKernel (const std::vector< P > &candidates, unsigned motionW, unsigned motionH, unsigned &bestId)
- static void perTreePredictions (const std::vector< std::vector< P > > &candidates, unsigned motionW, unsigned motionH, const cv::Size &featsize, unsigned predCols, unsigned step, MotionTreeClass::ENTROPY entropy, const std::string &imname, bool display)
- static std::vector< float > mostVotedFlow (const std::vector< std::vector< float > > &histo, unsigned &bestId)

2.5.1 Member Function Documentation

2.5.1.1 template<class P > std::vector< float > MotionPuzzle< P >::approxKernel (const std::vector< P > & *candidates,* unsigned *motionW,* unsigned *motionH,* unsigned & *bestId*) [static]

Given a vector of candidate patches compute their probabilities.

2.5.1.2 template<class P > bool MotionPuzzle< P >::checkConvergence (const cv::Mat & *motion,* const cv::Mat & *prevMotion*) [static]

Checks to see how much the prediction has changed between iterations.

2.5.1.3 template<class P > static std::vector<float> MotionPuzzle< P >::histInfo (const std::vector< std::vector< P > > & *candidates,* unsigned *motionW,* unsigned *motionH,* MotionTreeClass::ENTROPY *entropy*) [static]

Just get the the binning information.

2.5.1.4 template<class P > std::vector< P > MotionPuzzle< P >::initialPick (const std::vector< std::vector< P > > & *candidates,* unsigned *motionW,* unsigned *motionH,* MotionTreeClass::ENTROPY *entropy,* unsigned *predCols,* bool *neighborhood,* bool *display* = false) [static]

Generate a set of initial candidates (picks the most likely patches among the trees).

2.5.1.5 template<class P > std::vector< float > MotionPuzzle< P >::mostVotedFlow (const std::vector< std::vector< float > > & *histo,* unsigned & *bestId*) [static]

Find the most voted flow.

2.5.1.6 template<class P > void MotionPuzzle< P >::perTreePredictions (const std::vector< std::vector< P > > & *candidates,* unsigned *motionW,* unsigned *motionH,* const cv::Size & *featsize,* unsigned *predCols,* unsigned *step,* MotionTreeClass::ENTROPY *entropy,* const std::string & *imname,* bool *display*) [static]

Show per tree predictions and appearance.

2.5.1.7 template<class P > std::vector< P > MotionPuzzle< P >::pickApproxKernel (const std::vector< std::vector< P > > & *candidates,* unsigned *motionW,* unsigned *motionH,* MotionTreeClass::ENTROPY *entropy,* unsigned *predCols,* bool *neighborhood,* bool *display*) [static]

Picks the best patch per position among trees based on probability approximated using kernel density estimation.

2.5.1.8 template<class P > std::vector< P > MotionPuzzle< P >::pickMean (const std::vector< std::vector< P > > & *candidates,* unsigned *motionW,* unsigned *motionH,* bool *display*) [static]

Generates the best patch per position among trees as the mean of all patches.

2.5.1.9 template<class P > cv::Mat MotionPuzzle< P >::proposePrediction (const std::vector< P > & *candidates,* const cv::Size & *featsize,* unsigned *motionW,* unsigned *motionH,* MotionTreeClass::ENTROPY *entropy,* cv::Mat & *appearance*) [static]

Propose the final prediction over overlapping neighborhoods.

2.5.1.10 template< class P > cv::Mat **MotionPuzzle**< P >::proposePredictionMax (const std::vector< P > & *candidates,* const cv::Size & *featsize,* unsigned *motionW,* unsigned *motionH,* cv::Mat & *appearance*) `[static]`

Returns proposals appearance with the maximum.

2.5.1.11 template< class P > cv::Mat **MotionPuzzle**< P >::proposePredictionOverlap (const std::vector< P > & *candidates,* const cv::Size & *featsize,* unsigned *predCols,* unsigned *motionW,* unsigned *motionH,* unsigned *step,* MotionTreeClass::ENTROPY *entropy,* cv::Mat & *appearance*) `[static]`

Propose the final prediction over overlapping neighborhoods.

2.5.1.12 template< class P > cv::Mat **MotionPuzzle**< P >::proposePredictionSum (const std::vector< P > & *candidates,* const cv::Size & *featsize,* unsigned *motionW,* unsigned *motionH,* cv::Mat & *appearance*) `[static]`

Returns proposals for motions on X and Y.

2.5.1.13 template< class P > std::vector< P > **MotionPuzzle**< P >::selectPatches (const cv::Mat & *motion,* const std::vector< std::vector< P > > & *candidates,* unsigned *motionW,* unsigned *motionH*) `[static]`

Selects the patches that agree the most with the previous prediction.

2.5.1.14 template< class P > void **MotionPuzzle**< P >::showSamples (const P & *leaf,* unsigned *sampleW,* unsigned *sampleH*) `[static]`

Displays the set of predicted leaves.

2.5.1.15 template< class P > void **MotionPuzzle**< P >::solve (const std::vector< std::vector< P > > & *patches,* const cv::Size & *featsize,* unsigned *motionW,* unsigned *motionH,* cv::Mat & *motionX,* cv::Mat & *motionY,* cv::Mat & *appear,* MotionTreeClass::ENTROPY *entropy,* Puzzle< PuzzlePatch >::METHOD *method,* unsigned *predCols,* unsigned *step,* const std::string & *imname,* bool *neighborhood =* `false`*,* unsigned *maxIter =* `75`*,* bool *display =* `false`) `[static]`

Solves choosing the final motion prediction problem:

- choosing among the trees
- choosing in the neighborhood.

2.5.1.16 template< class P > void **MotionPuzzle**< P >::threshold (cv::Mat & *splitX,* cv::Mat & *splitY*) `[static]`

Threshold magnitudes.

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

- MotionPuzzle.h
- MotionPuzzle.cpp

2.6 MotionPuzzlePatch Struct Reference

```
#include <MotionPuzzle.h>
```

Public Member Functions

- **MotionPuzzlePatch** (const cv::Point &center, const std::vector< unsigned > &piece, const cv::Mat &motion, const cv::Mat &appearance, const std::vector< cv::Mat > &histo, float logProb, float motionProb, float appProb)
- virtual cv::Point pos2pt (unsigned pos, unsigned motionW, unsigned motionH) const
- float appearanceProb () const
- float **motionProb** () const
- cv::Mat **motion** () const
- cv::Mat **appearance** () const
- std::vector< cv::Mat > **histo** () const
- std::vector< float > **histo** (const cv::Point &pt) const
- void appearanceProb (float appearanceProb)
- void **motionProb** (float motionProb)
- void **motion** (const cv::Mat motion)
- void **appearance** (const cv::Mat appearance)
- void **histo** (const std::vector< cv::Mat > &histo)
- float **motionAgreement** (const cv::Mat &motion) const
- MotionPuzzlePatch (MotionPuzzlePatch const &rhs)
- MotionPuzzlePatch & **operator=** (MotionPuzzlePatch const &rhs)

2.6.1 Detailed Description

For storing the possible label-ings.

2.6.2 Constructor & Destructor Documentation

2.6.2.1 MotionPuzzlePatch::MotionPuzzlePatch (MotionPuzzlePatch const & *rhs*) [inline]

Copy constructors the default ones are not good with IplImages

2.6.3 Member Function Documentation

2.6.3.1 float MotionPuzzlePatch::appearanceProb ( ) const [inline]

Getters for the class members.

2.6.3.2 void MotionPuzzlePatch::appearanceProb (float *appearanceProb*) [inline]

Setters for the class members.

2.6.3.3 virtual cv::Point MotionPuzzlePatch::pos2pt (unsigned *pos,* unsigned *motionW,* unsigned *motionH*) const [inline], [virtual]

Finds the corresponding position in the original (small) image.

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

- MotionPuzzle.h

2.7 MotionRF< L, M, T, F, U > Class Template Reference

Public Member Functions

- **MotionRF** (int trees=0)
- float sigmaratio () const
- virtual void trainForestTree (unsigned min_s, unsigned max_d, CvRNG ∗pRNG, const M &TrData, unsigned samples, unsigned treeId, const char ∗path2models, const std::string &runName, typename StructureTree< M, T, F, U >::ENTROPY entropy, unsigned consideredCls, bool binary, float sigmaratio, bool leafavg, bool parentfreq, bool leafparentfreq, const std::string &runname, float entropysigma)

2.7.1 Member Function Documentation

2.7.1.1 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > float MotionRF< L, M, T, F, U >::sigmaratio ( ) const

Returns the value of sigma from one of the trees.

2.7.1.2 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > void MotionRF< L, M, T, F, U >::trainForestTree (unsigned *min_s,* unsigned *max_d,* CvRNG ∗ *pRNG,* const M & *TrData,* unsigned *samples,* unsigned *treeId,* const char ∗ *path2models,* const std::string & *runName,* typename StructureTree< M, T, F, U >::ENTROPY *entropy,* unsigned *consideredCls,* bool *binary,* float *sigmaratio,* bool *leafavg,* bool *parentfreq,* bool *leafparentfreq,* const std::string & *runname,* float *entropysigma*) [virtual]

Trains a specified tree in the forest on the given patches.

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

- MotionRF.h
- MotionRF.cpp

2.8 MotionRFdetector< L, M, T, F, U > Class Template Reference

Public Member Functions

- **MotionRFdetector** (StructureRF< L, M, T, F, U > ∗pRF, int w, int h, unsigned cls, unsigned labW, unsigned labH, unsigned motionW, unsigned motionH, typename Puzzle< PuzzlePatch >::METHOD method, unsigned step, typename MotionTree< M, T, F, U >::ENTROPY entropy)
- virtual void detectColor (const F ∗features, cv::Mat &motionDetect, cv::Mat &arrowsDetect, cv::Mat &appearDetect, const std::vector< const T ∗ > &patches, const cv::Size &imsize, const std::string &imname, const std::string &path2model, bool display=false)
- virtual void detectPyramid (const std::string &imname, const std::string &path2img, const std::string &path2feat, const std::string &ext, const std::vector< float > &pyramid, std::vector< cv::Mat > &vMotionDetect, std::vector< cv::Mat > &vArrowsDetect, std::vector< cv::Mat > &vAppearDetect, const std::string &path2model)
- unsigned motionW () const
- unsigned **motionH** () const
- MotionTree< M, T, F, U >::ENTROPY **entropy** () const
- void motionW (unsigned motionW)
- void **motionH** (unsigned motionH)
- void **entropy** (typename MotionTree< M, T, F, U >::ENTROPY entropy)

2.8.1 Member Function Documentation

2.8.1.1 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > void MotionRFdetector< L, M, T, F, U >::detectColor (const F ∗ *features,* cv::Mat & *motionDetect,* cv::Mat & *arrowsDetect,* cv::Mat & *appearDetect,* const std::vector< const T ∗ > & *patches,* const cv::Size & *imsize,* const std::string & *imname,* const std::string & *path2model,* bool *display =* `false`) `[virtual]`

Gets an input image and returns a detection image (pixel labels by RF regression). Given a set of predicted leafs for current pixel, get the final label: Simple: [1] Just get the most voted pixel label per position. Puzzle: [2] Optimized the patch selection label **[kontschider]**}.

2.8.1.2 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > void MotionRFdetector< L, M, T, F, U >::detectPyramid (const std::string & *imname,* const std::string & *path2img,* const std::string & *path2feat,* const std::string & *ext,* const std::vector< float > & *pyramid,* std::vector< cv::Mat > & *vMotionDetect,* std::vector< cv::Mat > & *vArrowsDetect,* std::vector< cv::Mat > & *vAppearDetect,* const std::string & *path2model*) `[virtual]`

Scales the image at a number of sizes and it labels each scale [?].

2.8.1.3 template<template< class M, class T, class F, class U > class L, class M, class T, class F, class U> unsigned MotionRFdetector< L, M, T, F, U >::motionW ( ) const `[inline]`

Getters for the class members.

2.8.1.4 template<template< class M, class T, class F, class U > class L, class M, class T, class F, class U> void MotionRFdetector< L, M, T, F, U >::motionW (unsigned *motionW*) `[inline]`

Setters for the class members.

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

- MotionRFdetector.h
- MotionRFdetector.cpp

2.9 MotionTree< M, T, F, U > Class Template Reference

Public Types

- typedef std::vector
  < std::vector< const T ∗ >
  >::const_iterator **vectConstIterT**
- typedef std::vector< const T ∗ >
  ::const_iterator **constIterT**
- typedef std::vector
  < std::vector< const T ∗ >
  >::iterator **vectIterT**
- typedef std::vector< const T ∗ >
  ::iterator **IterT**

Public Member Functions

- **MotionTree** (const char ∗filename, unsigned treeid, bool binary)

- **MotionTree** (unsigned minS, unsigned maxD, CvRNG ∗pRNG, unsigned labSz, unsigned patchW, unsigned patchH, unsigned patchCh, unsigned treeId, const char ∗path2models, const std::string &runName, typename StructureTree< M, T, F, U >::ENTROPY entropy, unsigned consideredCls, bool binary, float sigmaratio=0.2, bool leafavg=false, bool parentfreq=true, bool leafParentFreq=true, const std::string &runname="", float entropythresh=1e-3)
- void readTreeBin ()
- void readTree ()
- float stopCosSimilarity (const std::vector< std::vector< const T ∗ > > &SetA, const F ∗features)
- float stopEuclDist (const std::vector< std::vector< const T ∗ > > &trainSet, const F ∗features)
- float splitDistance2mean (const std::vector< std::vector< const T ∗ > > &SetA, const F ∗features, float &sizeA)
- float splitApproxKernel (const std::vector< std::vector< const T ∗ > > &SetA, const F ∗features, float &sizeA, std::vector< float > &prevfreq, std::vector< float > &bininfo)
- float showPickedSplit (const std::vector< std::vector< const T ∗ > > &SetA, const std::vector< std::vector< const T ∗ > > &SetB, const F ∗features, long double nodeid)
- void leafMean (const F ∗features, const std::vector< std::vector< const T ∗ > > &trainSet, int first, unsigned totsize, float &bestAppProb, float &bestMotionProb, cv::Mat ∗bestMotion, const std::vector< float > &prevfreq, const std::vector< float > &bininfo)
- void leafApprox (const F ∗features, const std::vector< std::vector< const T ∗ > > &trainSet, int first, unsigned totsize, float &bestAppProb, float &bestMotionProb, cv::Mat ∗bestApp, cv::Mat ∗bestMotion, std::vector< cv::Mat > &bestHisto, std::vector< float > &prevfreq, std::vector< float > &bininfo)
- std::vector< std::vector< float > > patchAppearanceSim (const std::vector< std::vector< const T ∗ > > &trainSet, const F ∗features, unsigned totPatches)
- std::vector< std::vector< float > > patchDist2Mean (const std::vector< std::vector< const T ∗ > > &trainSet, const F ∗features, unsigned totsize, const std::vector< float > &prevfreq, const std::vector< float > &bininfo)
- std::vector< std::vector< float > > patchApprox (const F ∗features, const std::vector< std::vector< const T ∗ > > &trainSet, unsigned totPatches, std::vector< float > &prevfreq, std::vector< float > &bininfo)
- std::vector< float > setFreq (const F ∗features, const std::vector< std::vector< const T ∗ > > &allTrainSet, std::vector< float > &bininfo)
- virtual void readNodeBin (node< U > ∗parent, std::ifstream &in, typename Tree< U >::SIDE side)
- virtual void readNode (node< U > ∗parent, std::ifstream &in, typename Tree< U >::SIDE side)
- virtual bool saveTree ()
- virtual bool saveTreeBin ()
- virtual bool saveTreeTxt ()
- virtual void initDataSizes (const M &trData)
- virtual void growTree (const M &trData, int samples)
- virtual void grow (const std::vector< std::vector< const T ∗ > > &trainSet, const F ∗features, long double &nodeid, unsigned int depth, int samples, node< U > ∗parent, typename Tree< U >::SIDE side, std::vector< float > &prevfreq, std::vector< float > &prevprevfreq, std::vector< float > &bininfo, bool showSplits=false)
- virtual bool optimizeTest (std::vector< std::vector< const T ∗ > > &SetA, std::vector< std::vector< const T ∗ > > &SetB, const std::vector< std::vector< const T ∗ > > &TrainSet, const F ∗features, long double ∗test, unsigned int iter, unsigned pick, std::vector< float > &freqA, std::vector< float > &bininfoA, std::vector< float > &freqB, std::vector< float > &bininfoB, float &best, float &motionA, float &motionB)
- float performSplit (std::vector< std::vector< const T ∗ > > &tmpA, std::vector< std::vector< const T ∗ > > &tmpB, const std::vector< std::vector< const T ∗ > > &TrainSet, const F ∗features, const std::vector< std::vector< IntIndex > > &valSet, unsigned pick, int threshold, unsigned &sizeA, unsigned &sizeB, std::vector< float > &freqA, std::vector< float > &freqB, std::vector< float > &bininfoA, std::vector< float > &bininfoB, float &motionA, float &motionB)
- virtual float measureSet (const std::vector< std::vector< const T ∗ > > &SetA, const std::vector< std::vector< const T ∗ > > &SetB, const F ∗features, unsigned pick, std::vector< float > &freqA, std::vector< float > &freqB, std::vector< float > &bininfoA, std::vector< float > &bininfoB, float &motionA, float &motionB)
- virtual void makeLeaf (const F ∗features, const std::vector< std::vector< const T ∗ > > &trainSet, long double nodeid, int depth, node< U > ∗parent, typename Tree< U >::SIDE side, unsigned nopatches, std::vector< float > &prevfreq, std::vector< float > &bininfo, float entropy, bool showLeaves=false)
- virtual void showSamples (const std::vector< std::vector< const T ∗ > > &trainSet, const F ∗features, unsigned nodeid, float entropy=0, const cv::Mat ∗bestMotion=NULL, bool justdisplay=false)

- cv::Point mpick () const
- unsigned **motionW** () const
- unsigned **motionH** () const
- float **sigmaratio** () const
- bool **parentFreq** () const
- bool **leafParentFreq** () const
- bool **leafavg** () const
- std::string **runname** () const
- float **entropythresh** () const
- clock_t **clockbegin** () const
- void mpick (const cv::Point &mpick)
- void **motionW** (unsigned motionW)
- void **motionH** (unsigned motionH)
- void **sigmaratio** (float sigmaratio)
- void **parentFreq** (bool parentFreq)
- void **leafParentFreq** (bool leafParentFreq)
- void **leafavg** (bool leafavg)
- void **runname** (const std::string &runname)
- void **entropythresh** (float entropythresh)
- void **clockbegin** (const clock_t &clockbegin)
- MotionTree (MotionTree const &rhs)
- MotionTree & **operator=** (MotionTree const &rhs)

Static Public Member Functions

- static float getFreqMagni (float valX, float valY, const std::vector< float > &bininfo, const std::vector< float > &freq)
- static float getFreqAngle (float angle, const std::vector< float > &bininfo, const std::vector< float > &freq)
- static void showSamples (const std::vector< const U ∗ > &leaves, unsigned sampleW, unsigned sampleH, const cv::Point &point)
- static float dotProd (const std::vector< float > &asmpl, const std::vector< float > &dimprobs)

2.9.1 Constructor & Destructor Documentation

2.9.1.1 template<class M, class T, class F, class U> MotionTree< M, T, F, U >::MotionTree (MotionTree< M, T, F, U > const & *rhs*) [inline]

Copy constructors for trees (to put them in the forest).

2.9.2 Member Function Documentation

2.9.2.1 template<class M , class T , class F , class U > float MotionTree< M, T, F, U >::dotProd (const std::vector< float > & *asmpl,* const std::vector< float > & *dimprobs*) [static]

Just dot product between vectors.

2.9.2.2 template<class M , class T , class F , class U > float MotionTree< M, T, F, U >::getFreqAngle (float *angle,* const std::vector< float > & *bininfo,* const std::vector< float > & *freq*) [static]

Given and input sample, find its corresponding inverse frequency.

2.9.2.3 template<class M , class T , class F , class U > float MotionTree< M, T, F, U >::getFreqMagni (float *valX,* float *valY,* const std::vector< float > & *bininfo,* const std::vector< float > & *freq*) [static]

Given and input sample, find its corresponding inverse frequency.

2.9.2.4 template<class M , class T , class F , class U > void MotionTree< M, T, F, U >::grow (const std::vector< std::vector< const T ∗ > > & *trainSet,* const F ∗ *features,* long double & *nodeid,* unsigned int *depth,* int *samples,* node< U > ∗ *parent,* typename Tree< U >::SIDE *side,* std::vector< float > & *prevfreq,* std::vector< float > & *prevprevfreq,* std::vector< float > & *bininfo,* bool *showSplits* = false) [virtual]

Creates the actual tree from the samples.

2.9.2.5 template<class M , class T , class F , class U > void MotionTree< M, T, F, U >::growTree (const M & *trData,* int *samples*) [virtual]

Implementing the <<growTee>> with multiple labels.

2.9.2.6 template<class M , class T , class F , class U > void MotionTree< M, T, F, U >::initDataSizes (const M & *trData*) [virtual]

Initializes the size of the labels, number of channels, etc.

2.9.2.7 template<class M , class T , class F , class U > void MotionTree< M, T, F, U >::leafApprox (const F ∗ *features,* const std::vector< std::vector< const T ∗ > > & *trainSet,* int *first,* unsigned *totsize,* float & *bestAppProb,* float & *bestMotionProb,* cv::Mat ∗ *bestApp,* cv::Mat ∗ *bestMotion,* std::vector< cv::Mat > & *bestHisto,* std::vector< float > & *prevfreq,* std::vector< float > & *bininfo*)

Keeps the most likely patch in the leaf given the approximation of kernel density estimation for the patch probability.

2.9.2.8 template<class M , class T , class F , class U > void MotionTree< M, T, F, U >::leafMean (const F ∗ *features,* const std::vector< std::vector< const T ∗ > > & *trainSet,* int *first,* unsigned *totsize,* float & *bestAppProb,* float & *bestMotionProb,* cv::Mat ∗ *bestMotion,* const std::vector< float > & *prevfreq,* const std::vector< float > & *bininfo*)

Take the mean of all patches arriving to the leaf.

2.9.2.9 template<class M , class T , class F , class U > void MotionTree< M, T, F, U >::makeLeaf (const F ∗ *features,* const std::vector< std::vector< const T ∗ > > & *trainSet,* long double *nodeid,* int *depth,* node< U > ∗ *parent,* typename Tree< U >::SIDE *side,* unsigned *nopatches,* std::vector< float > & *prevfreq,* std::vector< float > & *bininfo,* float *entropy,* bool *showLeaves* = false) [virtual]

Create leaf node from all patches.

2.9.2.10 template<class M , class T , class F , class U > float MotionTree< M, T, F, U >::measureSet (const std::vector< std::vector< const T ∗ > > & *SetA,* const std::vector< std::vector< const T ∗ > > & *SetB,* const F ∗ *features,* unsigned *pick,* std::vector< float > & *parentfreqA,* std::vector< float > & *parentfreqB,* std::vector< float > & *bininfoA,* std::vector< float > & *bininfoB,* float & *motionA,* float & *motionB*) [virtual]

Overloading the function to carry around the labels matrices.

2.9.2.11 template<class M, class T, class F, class U> cv::Point MotionTree< M, T, F, U >::mpick ( ) const [inline]

Getters for the class members.

2.9.2.12 template<class M, class T, class F, class U> void MotionTree< M, T, F, U >::mpick (const cv::Point & *mpick*) [inline]

Setters for the class members.

2.9.2.13 template<class M , class T , class F , class U > bool MotionTree< M, T, F, U >::optimizeTest (std::vector< std::vector< const T ∗ > > & *SetA*, std::vector< std::vector< const T ∗ > > & *SetB*, const std::vector< std::vector< const T ∗ > > & *TrainSet*, const F ∗ *features*, long double ∗ *test*, unsigned int *iter*, unsigned *pick*, std::vector< float > & *freqA*, std::vector< float > & *bininfoA*, std::vector< float > & *freqB*, std::vector< float > & *bininfoB*, float & *best*, float & *entropyA*, float & *entropyB*) [virtual]

Optimizes tests and thresholds. [1] Generate a 5 random values (for x1 y1 x2 y2 channel) in the <<test>> vector. [2] Evaluates the thresholds and finds the minimum and maximum index value [?]. [3] Iteratively generate random thresholds to split the index values [4] Split the data according to each threshold. [5] Find the best threshold and store it on the 6th position in <<test>>

2.9.2.14 template<class M , class T , class F , class U > std::vector< std::vector< float > > MotionTree< M, T, F, U >::patchAppearanceSim (const std::vector< std::vector< const T ∗ > > & *trainSet*, const F ∗ *features*, unsigned *totPatches*)

Gets the appearance probabilities in the leaf based on similarity.

2.9.2.15 template<class M , class T , class F , class U > std::vector< std::vector< float > > MotionTree< M, T, F, U >::patchApprox (const F ∗ *features*, const std::vector< std::vector< const T ∗ > > & *trainSet*, unsigned *totPatches*, std::vector< float > & *prevfreq*, std::vector< float > & *bininfo*)

For each patch finds it probability as 1/#bins sum_bins k(sample-bin).

2.9.2.16 template<class M , class T , class F , class U > std::vector< std::vector< float > > MotionTree< M, T, F, U >::patchDist2Mean (const std::vector< std::vector< const T ∗ > > & *trainSet*, const F ∗ *features*, unsigned *totsize*, const std::vector< float > & *prevfreq*, const std::vector< float > & *bininfo*)

Gets the closest patch to the mean-motion in the leaf (euclidian distance).

2.9.2.17 template<class M , class T , class F , class U > float MotionTree< M, T, F, U >::performSplit (std::vector< std::vector< const T ∗ > > & *tmpA*, std::vector< std::vector< const T ∗ > > & *tmpB*, const std::vector< std::vector< const T ∗ > > & *TrainSet*, const F ∗ *features*, const std::vector< std::vector< IntIndex > > & *valSet*, unsigned *pick*, int *threshold*, unsigned & *sizeA*, unsigned & *sizeB*, std::vector< float > & *parentfreqA*, std::vector< float > & *parentfreqB*, std::vector< float > & *bininfoA*, std::vector< float > & *bininfoB*, float & *entropyA*, float & *entropyB*)

Just splits the data into subsets and makes sure the subsets are not empty

2.9.2.18 template<class M , class T , class F , class U > void MotionTree< M, T, F, U >::readNode (node< U > ∗ *parent*, std::ifstream & *in*, typename Tree< U >::SIDE *side*) [virtual]

Recursively read tree from file.

2.9.2.19 template<class M , class T , class F , class U > void MotionTree< M, T, F, U >::readNodeBin (node< U > ∗ *parent,* std::ifstream & *in,* typename Tree< U >::SIDE *side*) [virtual]

Recursively read tree from binary file.

2.9.2.20 template<class M , class T , class F , class U > void MotionTree< M, T, F, U >::readTree ( )

Reads the tree from a regular text file.

2.9.2.21 template<class M , class T , class F , class U > void MotionTree< M, T, F, U >::readTreeBin ( )

Reads the tree from a binary file.

2.9.2.22 template<class M , class T , class F , class U > bool MotionTree< M, T, F, U >::saveTree ( ) [virtual]

Writes the current tree into a given binary file.

Writes the current tree into a given file.

2.9.2.23 template<class M , class T , class F , class U > bool MotionTree< M, T, F, U >::saveTreeBin ( ) [virtual]

Writes the current tree into a given binary file.

2.9.2.24 template<class M , class T , class F , class U > bool MotionTree< M, T, F, U >::saveTreeTxt ( ) [virtual]

Writes the current tree into a given file.

2.9.2.25 template<class M , class T , class F , class U > std::vector< float > MotionTree< M, T, F, U >::setFreq (const F ∗ *features,* const std::vector< std::vector< const T ∗ > > & *allTrainSet,* std::vector< float > & *bininfo*)

Get "class inverse frequencies" — inverse priors for reweighting.

2.9.2.26 template<class M , class T , class F , class U > float MotionTree< M, T, F, U >::showPickedSplit (const std::vector< std::vector< const T ∗ > > & *SetA,* const std::vector< std::vector< const T ∗ > > & *SetB,* const F ∗ *features,* long double *nodeid*)

show the mean to the samples for the picked best test.

2.9.2.27 template<class M , class T , class F , class U > void MotionTree< M, T, F, U >::showSamples (const std::vector< const U ∗ > & *leaves,* unsigned *sampleW,* unsigned *sampleH,* const cv::Point & *point*) [static]

Displays the set of predicted leaves.

2.9.2.28 template<class M , class T , class F , class U > void MotionTree< M, T, F, U >::showSamples (const std::vector< std::vector< const T ∗ > > & *trainSet,* const F ∗ *features,* unsigned *nodeid,* float *entropy* = 0, const cv::Mat ∗ *bestMotion* = NULL, bool *justdisplay* = false) [virtual]

Displays the samples among which we need to choose to make a leaf

2.9.2.29 template$<$class M , class T , class F , class U $>$ float **MotionTree**$<$ M, T, F, U $>$::splitApproxKernel (const std::vector$<$ std::vector$<$ const T $*$ $>$ $>$ & *SetA,* const F $*$ *features,* float & *sizeA,* std::vector$<$ float $>$ & *prevfreq,* std::vector$<$ float $>$ & *bininfo*)

Approximating continuous entropy with sum over sample probability, in turn approximated the density kernel estimation with pixel-wise kernels.

2.9.2.30 template$<$class M , class T , class F , class U $>$ float **MotionTree**$<$ M, T, F, U $>$::splitDistance2mean (const std::vector$<$ std::vector$<$ const T $*$ $>$ $>$ & *SetA,* const F $*$ *features,* float & *sizeA*)

In split: sum-squared-distance to the mean of the samples at the picked position.

Sum-Squared-Distance to the mean of the samples at the picked position.

2.9.2.31 template$<$class M , class T , class F , class U $>$ float **MotionTree**$<$ M, T, F, U $>$::stopCosSimilarity (const std::vector$<$ std::vector$<$ const T $*$ $>$ $>$ & *trainSet,* const F $*$ *features*)

Decides for a split based on the cosine similarity.

Check if all patches have converged to a single pattern by looking as MSE.

2.9.2.32 template$<$class M , class T , class F , class U $>$ float **MotionTree**$<$ M, T, F, U $>$::stopEuclDist (const std::vector$<$ std::vector$<$ const T $*$ $>$ $>$ & *trainSet,* const F $*$ *features*)

Check if all patches have converged to a single pattern by looking as MSE.

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

- MotionTree.h
- MotionTree.cpp

2.10 RunMotionRF$<$ L, M, T, F, U $>$ Class Template Reference

Public Types

- enum MODE {
  TRAIN_RF, **TEST_RF**, **TRAIN_TEST_RF**, **EXTRACT**,
  EXTRACT_OF, **TRAIN1**, **TEST1**, **EXTRACT_OF1** }

Public Member Functions

- **RunMotionRF** (const char $*$config)
- virtual void run (RunMotionRF::MODE mode, const std::vector$<$ const char $*$ $>$ &argv=std::vector$<$ const char $*$ $>$())
- virtual void runTrain ()
- virtual void runTest ()
- virtual void runExtract ()
- virtual void runExtractOF ()
- virtual void test (MotionRFdetector$<$ L, M, T, F, U $>$ &crDetect, std::string &testpath, std::string &resultpath, std::string &featpath)
- virtual void batchTest ()
- virtual void jobrunnerTest ()
- virtual void runTest1 (const std::vector$<$ const char $*$ $>$ &argv)

- void generateTestCommands (std::string &path2ims, std::string &path2results, const std::string &cwd, std::string &featpath)
- virtual void batchTrain ()
- virtual void jobrunnerTrain ()
- virtual void runTrain1 (const std::vector< const char * > &argv)
- virtual void batchExtractOF (const typename M::Algorithm &algo=M::Farneback)
- virtual void jobrunnerExtractOF (const typename M::Algorithm &algo=M::Farneback)
- virtual void generateExtractCommands (std::string &path2ims, const std::string &cwd, std::string &featpath)
- virtual void runExtract1 (const std::vector< const char * > &argv, const typename M::Algorithm &algo=M::Farneback)
- void extractOF (std::string &path2img, std::string &path2feat, const typename M::Algorithm &algo=M::Farneback)
- void extract (std::string &path2img, std::string &path2feat)
- unsigned motionWidth () const
- unsigned **motionHeight** () const
- float **sigmaratio** () const
- bool **warpping** () const
- bool **ofThresh** () const
- bool **leafavg** () const
- bool **parentfreq** () const
- bool **leafparentfreq** () const
- float **entropythresh** () const
- std::string **configfile** () const
- unsigned **serverport** () const
- bool **dryrun** () const
- unsigned **bins** () const
- void motionWidth (unsigned motionWidth)
- void **motionHeight** (unsigned motionHeight)
- void **sigmaratio** (float sigmaratio)
- void **warpping** (bool warpping)
- void **ofThresh** (bool ofThresh)
- void **leafavg** (bool leafavg)
- void **parentfreq** (bool parentfreq)
- void **leafparentfreq** (bool leafparentfreq)
- void **entropythresh** (float entropythresh)
- void **configfile** (std::string configfile)
- void **serverport** (unsigned serverport)
- void **dryrun** (bool dryrun)
- void **bins** (unsigned bins)

2.10.1 Member Enumeration Documentation

2.10.1.1 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > enum RunMotionRF::MODE

Modes of running the RF code

2.10.2 Member Function Documentation

2.10.2.1 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > void RunMotionRF< L, M, T, F, U >::batchExtractOF (const typename M::Algorithm & *algo* = `M::Farneback`) `[virtual]`

Extracts feature/label patches from all the images in a batch mode.

2.10.2.2 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > void RunMotionRF< L, M, T, F, U >::batchTest ( ) [virtual]

Predicts on a set of test images in batch mode.

2.10.2.3 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > void RunMotionRF< L, M, T, F, U >::batchTrain ( ) [virtual]

Trains the complete RF on the data set in a batch mode + threading.

2.10.2.4 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > void RunMotionRF< L, M, T, F, U >::extract (std::string & *path2img,* std::string & *path2feat*)

Recursively looks into the directories until it find the images it needs to extract features from.

2.10.2.5 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > void RunMotionRF< L, M, T, F, U >::extractOF (std::string & *path2img,* std::string & *path2feat,* const typename M::Algorithm & *algo* = M::Farneback)

Recursively looks into the directories until it find the images it needs to extract OF features from.

2.10.2.6 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > void RunMotionRF< L, M, T, F, U >::generateExtractCommands (std::string & *path2ims,* const std::string & *cwd,* std::string & *featpath*) [virtual]

Recursively reads the images from the test/train folder[s] and generates the commands to extract them with jobrunners.

2.10.2.7 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > void RunMotionRF< L, M, T, F, U >::generateTestCommands (std::string & *path2ims,* std::string & *path2results,* const std::string & *cwd,* std::string & *featpath*)

Recursively reads the images from the test folder[s].

2.10.2.8 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > void RunMotionRF< L, M, T, F, U >::jobrunnerExtractOF (const typename M::Algorithm & *algo* = M::Farneback) [virtual]

Generating commands to extract patches from every image separately.

2.10.2.9 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > void RunMotionRF< L, M, T, F, U >::jobrunnerTest ( ) [virtual]

Starts the jobrunner commands for testing 1 image at a time.

2.10.2.10 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > void RunMotionRF< L, M, T, F, U >::jobrunnerTrain ( ) [virtual]

Trains each tree separately with a jobrunner.

2.10.2.11 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > unsigned RunMotionRF< L, M, T, F, U >::motionWidth ( ) const [inline]

Getters for the class members.

2.10.2.12 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > void RunMotionRF< L, M, T, F, U >::motionWidth (unsigned *motionWidth*) [inline]

Setters for the class members.

2.10.2.13 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > void RunMotionRF< L, M, T, F, U >::run (RunMotionRF< L, M, T, F, U >::MODE *mode,* const std::vector< const char ∗ > & *argv* = std::vector<const char∗>()) [virtual]

Initialize and start training.

2.10.2.14 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > void RunMotionRF< L, M, T, F, U >::runExtract ( ) [virtual]

Extract the training/test features (what for?).

Extracts feature/label patches from all the images (WHAT IS THIS FOR?).

2.10.2.15 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > void RunMotionRF< L, M, T, F, U >::runExtract1 (const std::vector< const char ∗ > & *argv,* const typename M::Algorithm & *algo* = M::Farneback) [virtual]

Extract features from 1 image only with jobrunners.

2.10.2.16 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > void RunMotionRF< L, M, T, F, U >::runExtractOF ( ) [virtual]

Extracts only OF features.

Extracts feature/label patches from all the images.

2.10.2.17 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > void RunMotionRF< L, M, T, F, U >::runTest ( ) [virtual]

Initialize and start detector on test set.

Starts the prediction.

2.10.2.18 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > void RunMotionRF< L, M, T, F, U >::runTest1 (const std::vector< const char ∗ > & *argv*) [virtual]

Predicts on 1 image only with the jobrunners.

2.10.2.19 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > void RunMotionRF< L, M, T, F, U >::runTrain ( ) [virtual]

Initialize and start training.

2.10.2.20 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > void RunMotionRF< L, M, T, F, U >::runTrain1 (const std::vector< const char * > & *argv*) [virtual]

Starts the training for 1 tree only (with jobrunners).

Trains the one tree on the data set with the jobrunners.

2.10.2.21 template<template< class M, class T, class F, class U > class L, class M , class T , class F , class U > void RunMotionRF< L, M, T, F, U >::test (MotionRFdetector< L, M, T, F, U > & *crDetect,* std::string & *testpath,* std::string & *resultpath,* std::string & *featpath*) [virtual]

Performs the RF detection on test images.

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

- RunMotionRF.h
- RunMotionRF.cpp

Index

appearanceProb
    MotionLeafNode, 5
    MotionPuzzlePatch, 14
approxKernel
    MotionPuzzle, 12

batchExtractOF
    RunMotionRF, 23
batchTest
    RunMotionRF, 23
batchTrain
    RunMotionRF, 24

checkConvergence
    MotionPuzzle, 12
computeHistograms
    MotionPatch, 7

detectColor
    MotionRFdetector, 16
detectPyramid
    MotionRFdetector, 16
dotProd
    MotionTree, 18

extract
    RunMotionRF, 24
extractFeatures
    MotionPatch, 7
extractMotionAbsolute
    MotionPatch, 7
extractMotionRelative
    MotionPatch, 7
extractOF
    RunMotionRF, 24
extractPatches
    MotionPatch, 7
extractPatchesOF
    MotionPatch, 7

FeaturesMotion, 3
    flip, 4
    push_backHisto, 4
    push_backImages, 4
    push_backVelX, 4
    push_backVelY, 4
    velX, 4
findThreshold
    MotionPatch, 8
findThresholdAngle
    MotionPatch, 8
findmatches
    MotionPatch, 7
flip
    FeaturesMotion, 4

generateExtractCommands
    RunMotionRF, 24
generateTestCommands
    RunMotionRF, 24
getAngleHisto
    MotionPatch, 8
getFreqAngle
    MotionTree, 18
getFreqMagni
    MotionTree, 18
getMagniHisto
    MotionPatch, 8
getRansacAffineTransform
    MotionPatch, 8
grow
    MotionTree, 19
growTree
    MotionTree, 19

histInfo
    MotionPuzzle, 12
histo
    MotionPatchFeature, 10
histoCenter
    MotionPatchFeature, 10
HistoType
    MotionPatch, 7

image
    MotionPatchFeature, 10
imagePairs
    MotionPatch, 8
initDataSizes
    MotionTree, 19
initialPick
    MotionPuzzle, 12

jobrunnerExtractOF
    RunMotionRF, 24
jobrunnerTest
    RunMotionRF, 24
jobrunnerTrain
    RunMotionRF, 24
justFlow
    MotionPatch, 8

leafApprox
    MotionTree, 19
leafMean
    MotionTree, 19
loadPatches
    MotionPatch, 8

MODE
    RunMotionRF, 23
makeLeaf
    MotionTree, 19
measureSet
    MotionTree, 19
mostVotedFlow
    MotionPuzzle, 12
motion
    MotionPatchFeature, 10
MotionLeafNode, 4
    appearanceProb, 5
    MotionLeafNode, 5
    motionProb, 5
    MotionLeafNode, 5
    readLeaf, 5
    readLeafBin, 5
    showLeafBin, 5
    showLeafTxt, 5
MotionPatch
    computeHistograms, 7
    extractFeatures, 7
    extractMotionAbsolute, 7
    extractMotionRelative, 7
    extractPatches, 7
    extractPatchesOF, 7
    findThreshold, 8
    findThresholdAngle, 8
    findmatches, 7
    getAngleHisto, 8
    getMagniHisto, 8
    getRansacAffineTransform, 8
    HistoType, 7
    imagePairs, 8
    justFlow, 8
    loadPatches, 8
    motionW, 8
    pickRandomNames, 8
    reset, 9
    savePatches, 9
    showOF, 9
    warpOpenCV, 9
    warpSecond2First, 9
MotionPatch< T, F >, 5
MotionPatchFeature
    histo, 10
    histoCenter, 10
    image, 10
    motion, 10
    MotionPatchFeature, 10
    motionW, 11
    MotionPatchFeature, 10
MotionPatchFeature< F >, 9
motionProb
    MotionLeafNode, 5
MotionPuzzle
    approxKernel, 12
    checkConvergence, 12
    histInfo, 12
    initialPick, 12
    mostVotedFlow, 12
    perTreePredictions, 12
    pickApproxKernel, 12
    pickMean, 12
    proposePrediction, 12
    proposePredictionMax, 12
    proposePredictionOverlap, 13
    proposePredictionSum, 13
    selectPatches, 13
    showSamples, 13
    solve, 13
    threshold, 13
MotionPuzzle< P >, 11
MotionPuzzlePatch, 13
    appearanceProb, 14
    MotionPuzzlePatch, 14
    MotionPuzzlePatch, 14
    pos2pt, 14
MotionRF
    sigmaratio, 15
    trainForestTree, 15
MotionRF< L, M, T, F, U >, 15
MotionRFdetector
    detectColor, 16
    detectPyramid, 16
    motionW, 16
MotionRFdetector< L, M, T, F, U >, 15
MotionTree
    dotProd, 18
    getFreqAngle, 18
    getFreqMagni, 18
    grow, 19
    growTree, 19
    initDataSizes, 19
    leafApprox, 19
    leafMean, 19
    makeLeaf, 19
    measureSet, 19
    MotionTree, 18
    MotionTree, 18
    mpick, 19, 20
    optimizeTest, 20
    patchAppearanceSim, 20
    patchApprox, 20
    patchDist2Mean, 20
    performSplit, 20
    readNode, 20
    readNodeBin, 20
    readTree, 21
    readTreeBin, 21

saveTree, 21
saveTreeBin, 21
saveTreeTxt, 21
setFreq, 21
showPickedSplit, 21
showSamples, 21
splitApproxKernel, 21
splitDistance2mean, 22
stopCosSimilarity, 22
stopEuclDist, 22
MotionTree< M, T, F, U >, 16
motionW
MotionPatch, 8
MotionPatchFeature, 11
MotionRFdetector, 16
motionWidth
RunMotionRF, 24, 25
mpick
MotionTree, 19, 20

optimizeTest
MotionTree, 20

patchAppearanceSim
MotionTree, 20
patchApprox
MotionTree, 20
patchDist2Mean
MotionTree, 20
perTreePredictions
MotionPuzzle, 12
performSplit
MotionTree, 20
pickApproxKernel
MotionPuzzle, 12
pickMean
MotionPuzzle, 12
pickRandomNames
MotionPatch, 8
pos2pt
MotionPuzzlePatch, 14
proposePrediction
MotionPuzzle, 12
proposePredictionMax
MotionPuzzle, 12
proposePredictionOverlap
MotionPuzzle, 13
proposePredictionSum
MotionPuzzle, 13
push_backHisto
FeaturesMotion, 4
push_backImages
FeaturesMotion, 4
push_backVelX
FeaturesMotion, 4
push_backVelY
FeaturesMotion, 4

readLeaf
MotionLeafNode, 5
readLeafBin
MotionLeafNode, 5
readNode
MotionTree, 20
readNodeBin
MotionTree, 20
readTree
MotionTree, 21
readTreeBin
MotionTree, 21
reset
MotionPatch, 9
run
RunMotionRF, 25
runExtract
RunMotionRF, 25
runExtract1
RunMotionRF, 25
runExtractOF
RunMotionRF, 25
RunMotionRF
batchExtractOF, 23
batchTest, 23
batchTrain, 24
extract, 24
extractOF, 24
generateExtractCommands, 24
generateTestCommands, 24
jobrunnerExtractOF, 24
jobrunnerTest, 24
jobrunnerTrain, 24
MODE, 23
motionWidth, 24, 25
run, 25
runExtract, 25
runExtract1, 25
runExtractOF, 25
runTest, 25
runTest1, 25
runTrain, 25
runTrain1, 25
test, 26
RunMotionRF< L, M, T, F, U >, 22
runTest
RunMotionRF, 25
runTest1
RunMotionRF, 25
runTrain
RunMotionRF, 25
runTrain1
RunMotionRF, 25

savePatches
MotionPatch, 9
saveTree
MotionTree, 21
saveTreeBin
MotionTree, 21

saveTreeTxt
MotionTree, 21
selectPatches
MotionPuzzle, 13
setFreq
MotionTree, 21
showLeafBin
MotionLeafNode, 5
showLeafTxt
MotionLeafNode, 5
showOF
MotionPatch, 9
showPickedSplit
MotionTree, 21
showSamples
MotionPuzzle, 13
MotionTree, 21
sigmaratio
MotionRF, 15
solve
MotionPuzzle, 13
splitApproxKernel
MotionTree, 21
splitDistance2mean
MotionTree, 22
stopCosSimilarity
MotionTree, 22
stopEuclDist
MotionTree, 22

test
RunMotionRF, 26
threshold
MotionPuzzle, 13
trainForestTree
MotionRF, 15

velX
FeaturesMotion, 4

warpOpenCV
MotionPatch, 9
warpSecond2First
MotionPatch, 9