Automating Analysis of Spatial Grids Reference Manual

Generated by Doxygen 1.4.7

Fri Jan 27 09:58:25 2012

Contents

Chapter 1

Automating Analysis of Spatial Grids Namespace Index

1.1 Automating Analysis of Spatial Grids Package List

Here are the packages with brief descriptions (if available):

Chapter 2

Automating Analysis of Spatial Grids Hierarchical Index

2.1 Automating Analysis of Spatial Grids Class Hierarchy

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

Chapter 3

Automating Analysis of Spatial Grids Class Index

3.1 Automating Analysis of Spatial Grids Class List

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

Chapter 4

Automating Analysis of Spatial Grids File Index

4.1 Automating Analysis of Spatial Grids File List

Here is a list of all files with brief descriptions:

Chapter 5

Automating Analysis of Spatial Grids Namespace Documentation

5.1 Package edu

Packages

- package ou

5.2 Package edu.ou

Packages

- package asgbook

5.3 Package edu.ou.asgbook

Packages

- package core

  Classes that form the basis of spatial analysis techniques discussed in the book.

- package datamining

  Classes discussed in Chapter 8 (Data Mining).

- package dataset

  Classes that provide the ability to read the various datasets that are used to illustrate spatial analysis techniques.

- package distance

  Distance Transform methods (See Chapter 4).

- package filters

  Most of these filters are neighborhood/window based and are discussed in Chapter 5.

- package geocode

  Geocoding is discussed in Chapter 2.

- package gmm

  A parametric approximation to an image (See Chapter 3).

- package histogram

  Histograms and histogram-based techniques are discussed in Chapter 4.

- package imgstat

  Texture, vector quantization and other image statistics techniques discussed in Chapter 4.

- package io

  Helper classes to read ESRI grids and write out KML/PNG.

- package linearity

  Linearity verification and data transformation are discussed in Chapter 2.

- package motion

  Change and motion estimation techniques discussed in Chapter 7.

- package oban

Techniques to interpolate point observations to a spatial grid discussed in Chapter 3.

- package projections

  Map projection techniques discussed in Chapter 2.

- package rasterization

  Rasterization techniques discussed in Chapter 2.

- package rbf

  Parametric approximations discussed in Chapter 3.

- package segmentation

  Object identification techniques discussed in Chapter 6.

- package thinning

  Skeletonization techniques discussed in Chapter 5.

- package transforms

  Fourier Transforms are discussed in Chapter 5.

- package usage

  A classroom assignment; an example algorithm.

5.4 Package edu.ou.asgbook.core

Classes that form the basis of spatial analysis techniques discussed in the book.

Classes

- class LatLon

  A point on the earth's surface typically in WGS84.

- class LatLonGrid

  A geospatial grid of data in equilat equilon coordinates typically in WGS84 ellipsoid.

- class LevelSet

  A representation of a spatial grid as a set of levels.

- class Pair< X, Y >

  An utility class so that methods can return two objects.

- class Pixel

  A grid point in a spatial grid consists of a location and value.

- class PointObservations

  A set of observation points.

- class ScalarStatistic

  A utility class to compute mean, variance of a streaming set of inputs.

5.4.1 Detailed Description

Classes that form the basis of spatial analysis techniques discussed in the book.

These are discussed in Chapter 3.

5.5 Package edu.ou.asgbook.datamining

Classes discussed in Chapter 8 (Data Mining).

Classes

- class CityCategories

  Obtains city data for clustering.

- class CityGdiModels

  Applies different data mining models to each city.

- class FuzzyCandidateMarket

  Uses heuristic rules to choose the next market to enter.

- class FuzzyLogic

  A simple fuzzy logic engine.

- class GdiPattern

  The training pattern for each city.

- class PrimaryCities

  Identifies the primary cities in each country.

5.5.1 Detailed Description

Classes discussed in Chapter 8 (Data Mining).

5.6 Package edu.ou.asgbook.dataset

Classes that provide the ability to read the various datasets that are used to illustrate spatial analysis techniques.

Classes

- class CountryPolygons

  Reads country-by-country coordinates from a KML placemarks file.

- class DailyRainfall

  Reads the ASCII precipitation data available at http://madis-data.noaa.gov/public/hydrodumpguest.html.

- class GlobalPopulation

  Reads the ASCII population data available at http://sedac.ciesin.columbia.edu/gpw.

- class MadisTemperature

  Reads the ASCII temperature data available at http://madis-data.noaa.gov/public/sfcdumpguest.html.

- class NightimeLights

  Reads night-time lights data in ESRI grid format.

- class SeviriInfraredTemperature

  To read binary dump output from WDSS-II (http://www.wdssii.org/).

- class SurfaceAlbedo

  Reads lambert-conformal ascii grid.

- class WorldBankGDI

  Reads country-by-country Global development index from World Bank.

5.6.1 Detailed Description

Classes that provide the ability to read the various datasets that are used to illustrate spatial analysis techniques.

5.7 Package edu.ou.asgbook.distance

Distance Transform methods (See Chapter 4).

Classes

- interface EuclideanDT
- class EuclideanDTPropagation

  Implementation of Euclidean distance that updates the distance instead of computing it afresh each time.

- class EuclideanDTRecursivePropagation

  Note that this class is only for illustrative purposes.

- class EuclideanDTSaito

  The Saito technique of computing the distance transform by calculating in the two directions separately.

5.7.1 Detailed Description

Distance Transform methods (See Chapter 4).

5.8 Package edu.ou.asgbook.filters

Most of these filters are neighborhood/window based and are discussed in Chapter 5.

Classes

- class ConvolutionFilter

  Convolve an image by a window.

- class DilateErodeFilter

  Carries out paired dilation followed by erosion for filling in holes.

- class DilationFilter

  Expands entities by taking a local maximum.

- class ErodeDilateFilter

  Carries out paired erosion followed by dilation for denoising.

- class ErosionFilter

  Reduces the size of entities by taking a local mininum.

- class Inverter

  at every pixel, replaces its value (val) by (A - val)

- class LoGEdgeFilter

  Laplacian of a Gaussian edge filter.

- class MatchedFilter

  Convolve an image by a window that is akin to the features we want to extract.

- class MaxValueFilter

  Finds the highest value pixel in the image.

- class MedianFilter

  A smoothing operation that involves replacing a pixel by the local median.

- class MultiFilter

  Carries out multiple operations.

- class NHighest

  Finds the N highest valued-pixels in image.

- class NHighestLevelSetImpl

  Finds the N highest valued-pixels in image using a levelset implementation.

- class OrientedEllipseFilter

  A non-isotropic smoothing filter.

- class QuickSelect

  From Numerical Recipes, a fast way to find the kth smallest item in a list Useful to implement rank filters.

- class SaturateFilter

  Sets all values $<$ MIN to MIN and all values $>$ MAX to MAX.

- class SeparableConvolutionFilter

  An optimized convolution filter.

- class SimpleThresholder

  Replace pixel values with 1 or 0 depending on whether they are above or below a single threshold.

- class SobelEdgeFilter

  Find edges in a grid.

- interface SpatialFilter
- class SpeckleFilter

  Denoising filter that removes speckle.

5.8.1 Detailed Description

Most of these filters are neighborhood/window based and are discussed in Chapter 5.

5.9 Package edu.ou.asgbook.geocode

Geocoding is discussed in Chapter 2.

Classes

- class UsaZipcode

  Find the city for each zipcode in the USA.

5.9.1 Detailed Description

Geocoding is discussed in Chapter 2.

5.10 Package edu.ou.asgbook.gmm

A parametric approximation to an image (See Chapter 3).

Classes

- class GaussianComponent

  Component of a Gaussian Mixture Model.

- class GaussianMixtureModel

  A parametric approximation of a spatial grid as a sum of Gaussians.

5.10.1 Detailed Description

A parametric approximation to an image (See Chapter 3).

5.11 Package edu.ou.asgbook.histogram

Histograms and histogram-based techniques are discussed in Chapter 4.

Classes

- class CumulativeDistributionFunction

  Forms a CDF from a Histogram.

- class Entropy

  Compute entropy from a histogram.

- class Histogram

  A histogram is an empirical probability distribution.

- class HistogramBinSelection

  Tries out different values for the number of bins and replaces each pixel value by the center of its bin.

- class OtsuThresholdSelector

  Uses Otsu (1979) to select optimal threshold.

5.11.1 Detailed Description

Histograms and histogram-based techniques are discussed in Chapter 4.

5.12 Package edu.ou.asgbook.imgstat

Texture, vector quantization and other image statistics techniques discussed in Chapter 4.

Classes

- class GraylevelCooccurenceMatrix

  Computes texture properties from a GLCM.

- class LocalMeasures

  Statistics computed in the neighborhood of a pixel.

- class Quantizer

  Develops a quantization scheme using histogram equalization.

- class StructuralMeasures

  Statistics computed in the neighborhood of a pixel.

- class VectorQuantizer

  Develops a quantization scheme using vector quantization.

5.12.1 Detailed Description

Texture, vector quantization and other image statistics techniques discussed in Chapter 4.

5.13 Package edu.ou.asgbook.io

Helper classes to read ESRI grids and write out KML/PNG.

Classes

- class EsriGrid

  Read an ESRI grid.

- class KmlWriter

  Writes data out in KML form, for display in Google Earth or similar program.

- class OutputDirectory

  Change this to change the output directory that is used by all the main().

- class PngWriter

  Writes a spatial grid out as PNG file.

5.13.1 Detailed Description

Helper classes to read ESRI grids and write out KML/PNG.

5.14 Package edu.ou.asgbook.linearity

Linearity verification and data transformation are discussed in Chapter 2.

Classes

- class DataTransform

  Transform pixel values, usually to meet linearity requirements.

- class LinearityVerifier

  Given a 2D array of points, reports error measures of assuming linearity.

- class LinearScaling

  Scales pixel values as Ax.

- class LogScaling

  Transforms pixel values as log(x).

5.14.1 Detailed Description

Linearity verification and data transformation are discussed in Chapter 2.

5.15 Package edu.ou.asgbook.motion

Change and motion estimation techniques discussed in Chapter 7.

Classes

- class AlignAndDifference

  Aligns two grids and then computes their difference.

- class CrossCorrelation

  Estimates motion using cross-correlation.

- class Differencer

  Just computes a pixel-by-pixel difference.

- class EdgeBased

  Estimates motion based on the displacement of edges.

- class HornSchunk

  Horn-Schunk optical flow method of motion estimation.

- class HungarianAssigner

  Optimal assignment algorithm.

- class HybridTracker

  Estimates motion by finding cross-correlation of objects in one frame to the pixels in the previous frame.

- class KalmanFilter

  For the time smoothing of motion vectors.

- interface MotionEstimator
- class ObjectTracker

  Estimates motion based on assigning objects in one frame to objects in the previous frame.

- class PhaseCorrelation

  Estimate motion based on FFT.

- class PyramidalCrossCorrelation

  Cross-correlation at muliple resolutions.

5.15.1 Detailed Description

Change and motion estimation techniques discussed in Chapter 7.

5.16 Package edu.ou.asgbook.oban

Techniques to interpolate point observations to a spatial grid discussed in Chapter 3.

Classes

- class CressmanWeighting

  An interpolation method that uses 1/r^2.

- class GaussWeighting

  An interpolation method that uses exp(-1/r^2).

- class ObjectiveAnalysisUtils

  Utility functions for objective analysis.

- class WeightedAverage

  Interpolation methods for point observations.

- class WeightedAverageOptimized
- interface WeightFunction

  Used by WeightedAverage.

5.16.1 Detailed Description

Techniques to interpolate point observations to a spatial grid discussed in Chapter 3.

5.17 Package edu.ou.asgbook.projections

Map projection techniques discussed in Chapter 2.

Classes

- class Ellipsoid

  An ellipsoidal approximation to the earth.

- class LambertConformal2SP

  Lambert Conformation 2 Standard Parallels map projection.

- class Remapper

  Utilities to remap one map projection to another.

5.17.1 Detailed Description

Map projection techniques discussed in Chapter 2.

5.18 Package edu.ou.asgbook.rasterization

Rasterization techniques discussed in Chapter 2.

Classes

- class BoundingBox

  A rectangular bounding box of a polygon.

- class CatmullRom

  A Catmull-Rom spline, a local spline.

- class Line

  A line that connects two points on the earth's surface.

- class Polygon

  A polygon consisting of straight edges along the earth's surface.

5.18.1 Detailed Description

Rasterization techniques discussed in Chapter 2.

5.19 Package edu.ou.asgbook.rbf

Parametric approximations discussed in Chapter 3.

Classes

- class DataSimulator

  Simulates RBF data to be fit.

- class ProjectionPursuit

  Approximates a spatial grid by a RBF when nothing is known beyond the number of Gaussians desired.

- class RadialBasisFunction

  Finds best fit of a spatial grid to a sum of Gaussians when the centers and sigmas of the Gaussians are known.

5.19.1 Detailed Description

Parametric approximations discussed in Chapter 3.

5.20 Package edu.ou.asgbook.segmentation

Object identification techniques discussed in Chapter 6.

Classes

- class ContiguityEnhancedKMeansSegmenter

  Objects consist of pixels that are grown from initial centers using K-means.

- class EnhancedWatershedSegmenter

  Enhanced watershed segmentation following Lakshmanan, Hondl and Rabin.

- class HysteresisSegmenter

  Objects consist of pixels that are $>$ thresh2 but have at least one pixel $>$ thresh1.

- class LabelResult

  Result of segmentation.

- class MultiscaleKMeansSegmenter

  Quantizes image into K levels, then does multiscale segmentation Does not implement the pruning techniques discussed in the paper.

- class RegionGrowing

  Common object-identification utility.

- class RegionProperty

  Properties of a region such as geometric (centroid, area, etc) and physical (based on other grid values).

- interface Segmenter

  Object identification technique.

- class SnakeActiveContour

  Active contour method of identifying objects.

- class ThresholdSegmenter

  Simple object identification based on a single threshold.

- class WatershedSegmenter

  Watershed approach of object identification.

5.20.1 Detailed Description

Object identification techniques discussed in Chapter 6.

5.21 Package edu.ou.asgbook.thinning

Skeletonization techniques discussed in Chapter 5.

Classes

- class HilditchSkeletonization

  Hilditch method of skeletonizing a grid.

- class MedialAxisSkeletonization

  The MAT method of skeletonizing a grid.

5.21.1 Detailed Description

Skeletonization techniques discussed in Chapter 5.

5.22 Package edu.ou.asgbook.transforms

Fourier Transforms are discussed in Chapter 5.

Classes

- class AlignmentEstimator

  Estimate the degree of spatial displacement between two similar grids.
- class FFT

  FFT based on Sedgewick and Wayne.
- class FFT2D

  Two-dimensional FFT.
- class FFTBandpassFilter

  Removes noise (high frequencies) and the gross signal (low frequencies).
- class FFTConvolutionFilter

  An optimization for convolution using FFTs.
- class HoughTransform

  Finds lines in image.

5.22.1 Detailed Description

Fourier Transforms are discussed in Chapter 5.

The Hough Transform is discussed in Chapter 6.

5.23 Package edu.ou.asgbook.usage

A classroom assignment; an example algorithm.

Classes

- class Assignment4

  (1) Find optimal threshold on log(pop) Find distance of every grid point to a point < thresh Find optimal threshold of distance values Threshold image to keep only values < threshold

- class Sprawl

  Solution to a classroom assignment to identify regions of urban sprawl from the population density data.

5.23.1 Detailed Description

A classroom assignment; an example algorithm.

Chapter 6

Automating Analysis of Spatial Grids Class Documentation

6.1 edu.ou.asgbook.motion.AlignAndDifference Class Reference

Aligns two grids and then computes their difference.

Public Member Functions

- LatLonGrid compute (LatLonGrid data0, LatLonGrid data1, Pair< LatLonGrid, LatLonGrid > uv)
- LatLonGrid compute (LatLonGrid data0, LatLonGrid data1, Pair< LatLonGrid, LatLonGrid > uv, int MOT_SCALE)

6.1.1 Detailed Description

Aligns two grids and then computes their difference.

Author:

v.lakshmanan

6.1.2 Member Function Documentation

6.1.2.1 LatLonGrid edu.ou.asgbook.motion.AlignAndDifference.compute (LatLonGrid *data0*, LatLonGrid *data1*, Pair< LatLonGrid, LatLonGrid > *uv*, int *MOT_SCALE*)

```

        LatLonGrid result = LatLonGrid.copyOf(data1);

        final float mot_scale = MOT_SCALE; // integer division truncates
        for (int i=0; i < result.getNumLat(); ++i){
            for (int j=0; j < result.getNumLon(); ++j){
                // align by moving data0 to match up with data1
                // then compute difference
                int aligned0 = data0.getValue(i,j);
                // find motion at this point
                int motx = Math.round(uv.first.getValue(i,j) / mot_scale);
                int moty = Math.round(uv.second.getValue(i,j) / mot_scale);
                // grab pixel from old location
                int oldx = i - motx;
                int oldy = j - moty;
                if (data0.isValid(oldx, oldy)){
                    aligned0 = data0.getValue(oldx, oldy);
                }
                int diff = data1.getValue(i,j) - aligned0;
                result.setValue(i,j, diff);
            }
        }

        return result;
    }
```

6.1.2.2 LatLonGrid edu.ou.asgbook.motion.AlignAndDifference.compute (LatLonGrid *data0*, LatLonGrid *data1*, Pair< LatLonGrid, LatLonGrid > *uv*)

```

        return compute(data0, data1, uv, 1);
    }
```

6.2 edu.ou.asgbook.transforms.AlignmentEstimator Class Reference

Estimate the degree of spatial displacement between two similar grids.

Public Member Functions

- AlignmentEstimator (int maxu, int maxv, LatLonGrid a, LatLonGrid b)

  convolve a with b and find where the maximum correlation lies

Static Public Member Functions

- static Pixel computeCentroid (LatLonGrid a)
- static void main (String[ ] args) throws Exception

Package Attributes

- final int MAXU
- final int MAXV
- int motNS
- int motEW

6.2.1 Detailed Description

Estimate the degree of spatial displacement between two similar grids.

Author:

v.lakshmanan

6.2.2 Constructor & Destructor Documentation

6.2.2.1 edu.ou.asgbook.transforms.AlignmentEstimator.AlignmentEstimator (int *maxu*, int *maxv*, LatLonGrid *a*, LatLonGrid *b*)

convolve a with b and find where the maximum correlation lies

```
                                                                    {
        this.MAXU = maxu;
        this.MAXV = maxv;

        // a
        Complex[][] in1 = FFT2D.fft(FFT2D.zeropad(a));

        // zero-out an area of thickness MAXU/MAXV around the boundary to avoid boundary iss
        LatLonGrid centerb = LatLonGrid.copyOf(b);
        int minx = MAXU;
        int miny = MAXV;
        int maxx = centerb.getNumLat() - minx;
        int maxy = centerb.getNumLon() - miny;
        for (int i=0; i < b.getNumLat(); ++i){
            for (int j=0; j < b.getNumLon(); ++j){
                if (i < minx || j < miny || i > maxx || j > maxy){
                    centerb.setValue(i, j, 0);
                }
            }
        }
        Complex[][] in2 = FFT2D.fft(FFT2D.zeropad(centerb));

        // find phase shift at this point
        for (int i=0; i < in1.length; ++i) for (int j=0; j < in1[0].length; ++j){
            in1[i][j] = in1[i][j].multiply(in2[i][j].conjugate());
            in1[i][j] = in1[i][j].multiply( 1.0 / in1[i][j].norm() );
        }
        // take ifft
        Complex[][] result = FFT2D.ifft(in1);

        // find location at which the convolved result is maximum
        double bestValue = Integer.MIN_VALUE;
        int startx = 0; // result.length/2 - MAXU;
        int starty = 0; // result[0].length/2 - MAXV;
        int endx = result.length; // /2 + MAXU;
        int endy = result[0].length; // /2 + MAXV;
        for (int i=startx; i < endx; ++i) for (int j=starty; j < endy; ++j){
            if ( result[i][j].normsq() > bestValue ){
                bestValue = result[i][j].real;
                motNS = -i;
                motEW = -j;
            }
        }

        // we don't want a 345-degree phase shift; we want it to be 15-degrees
        if ( Math.abs(motNS) > result.length/2 ){
            if (motNS < 0) motNS += result.length;
            else motNS -= result.length;
        }
        if ( Math.abs(motEW) > result[0].length/2 ){
            if (motEW < 0) motEW += result[0].length;
            else motEW -= result[0].length;
        }
    }
```

6.2.3 Member Function Documentation

6.2.3.1 static Pixel edu.ou.asgbook.transforms.AlignmentEstimator.computeCentroid (LatLonGrid *a*) [static]

```
                                                       {
           double sumx = 0;
           double sumy = 0;
           double sumwt = 0;
           int N = 0;
           for (int i=0; i < a.getNumLat(); ++i) for (int j=0; j < a.getNumLon(); ++j){
               double wt = a.getValue(i,j);
               sumx += i * wt;
               sumy += j * wt;
               sumwt += wt;
               ++N;
           }
           return new Pixel((int)Math.round(sumx/sumwt), (int)Math.round(sumy/sumwt), (int)Math.round(sumw
       }
```

6.2.3.2 static void edu.ou.asgbook.transforms.AlignmentEstimator.main (String[ ] *args*) throws Exception [static]

```
                                                              {
           // because the alignment doesn't really check lat-lon extents,
           // cropping from offset corners will look like translation ...
           LatLonGrid conus = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA);
           LatLonGrid[] grids = new LatLonGrid[2];
            grids[0] = conus.crop(900, 2500, 256, 256);
            int motx = 5; int moty = 9;
            grids[1] = conus.crop(900-motx, 2500-moty, 256, 256);

            // do alg
            AlignmentEstimator alg = new AlignmentEstimator(30,30,grids[0], grids[1]);
            System.out.println("Motion N/S ="  + alg.motNS + " true N/S=" + motx);
            System.out.println("Motion E/W ="  + alg.motEW + " true E/W=" + moty);

            System.out.println("Centroid of first = " + computeCentroid(grids[0]));
            System.out.println("Centroid of second = " + computeCentroid(grids[1]));

            // based on edges alone
            SobelEdgeFilter edgeFilter = new SobelEdgeFilter();
            LatLonGrid edge1 = edgeFilter.edgeFilter(grids[0]);
            LatLonGrid edge2 = edgeFilter.edgeFilter(grids[1]);
            AlignmentEstimator alg2 = new AlignmentEstimator(30,30,edge1, edge2);
            System.out.println("Edge Motion N/S ="  + alg2.motNS );
            System.out.println("Edge Motion E/W ="  + alg2.motEW );
        }
```

6.2.4 Member Data Documentation

6.2.4.1 final int edu.ou.asgbook.transforms.AlignmentEstimator.MAXU

`[package]`

6.2.4.2 final int edu.ou.asgbook.transforms.AlignmentEstimator.MAXV

`[package]`

6.2.4.3 int edu.ou.asgbook.transforms.AlignmentEstimator.motEW

`[package]`

6.2.4.4 int edu.ou.asgbook.transforms.AlignmentEstimator.motNS

`[package]`

6.3 edu.ou.asgbook.usage.Assignment4 Class Reference

(1) Find optimal threshold on log(pop) Find distance of every grid point to a point < thresh Find optimal threshold of distance values Threshold image to keep only values < threshold

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.3.1 Detailed Description

(1) Find optimal threshold on log(pop) Find distance of every grid point to a point < thresh Find optimal threshold of distance values Threshold image to keep only values < threshold

Author:

v.lakshmanan

6.3.2 Member Function Documentation

6.3.2.1 static void edu.ou.asgbook.usage.Assignment4.main (String[ ] *args*) throws Exception [static]

```
                                                            {
        File outdir = OutputDirectory.getDefault("assignment4");

        // read input
        LatLon nwCorner = new LatLon(60, -130);
        LatLon seCorner = new LatLon(7, -52);
        // LatLonGrid conus = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new GlobalPopulatior
        LatLonGrid conus = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA);
        conus = conus.crop(conus.getRow(nwCorner),
                conus.getCol(nwCorner),
                conus.getRow(seCorner) - conus.getRow(nwCorner),
                conus.getCol(seCorner) - conus.getCol(nwCorner));
        KmlWriter.write(conus, outdir, "orig", PngWriter.createCoolToWarmColormap());

        // find threshold
        int popthresh = -1;
        {
            final int MIN = 0;
            final int MAX = 500;
            final int incr = 10;
            Histogram hist = new Histogram(MIN, incr, (MAX-MIN)/incr );
            hist.update(conus);
            popthresh = new OtsuThresholdSelector(hist).getOptimalThreshold();
            System.out.println("Optimal population threshold=" + popthresh);
        }

        // threshold
        //LatLonGrid threshed = new SimpleThresholder(popthresh).filter(conus);
        //KmlWriter.write(threshed, outdir, "thresh", PngWriter.createCoolToWarmColormap());

        // distance to points > thresh
        LatLonGrid distToCity = new EuclideanDTSaito().getDistanceTransform(conus, popthres
        KmlWriter.write(distToCity, outdir, "distToCity", PngWriter.createCoolToWarmColorma

        // optimal threshold on distance
        int distthresh = -1;
        {
            final int MIN = 0;
            final int MAX = 10000;
            final int incr = 10;
            Histogram hist = new Histogram(MIN, incr, (MAX-MIN)/incr );
            hist.update(distToCity);
            distthresh = new OtsuThresholdSelector(hist).getOptimalThreshold();
            System.out.println("Optimal distance threshold=" + distthresh);
        }

        // threshold by distance to find metropolitan areas
        LatLonGrid boondocks = new SimpleThresholder(distthresh/2).filter(distToCity);
        LatLonGrid metros = new Inverter(1).filter(boondocks);
        KmlWriter.write(metros, outdir, "metros", PngWriter.createCoolToWarmColormap());


    }
```

6.4 edu.ou.asgbook.rasterization.BoundingBox Class Reference

A rectangular bounding box of a polygon.

Public Member Functions

- BoundingBox (LatLon[ ] vertices)
- boolean contains (double x, double y)
- void update (BoundingBox a)

Static Public Member Functions

- static BoundingBox copyOf (BoundingBox a)

6.4.1 Detailed Description

A rectangular bounding box of a polygon.

It can sometimes be cheaper to use a bounding box instead of the accurate locations and do the real calcuation only if the bounding box passes.

Author:

valliappa.lakshmanan

6.4.2 Constructor & Destructor Documentation

6.4.2.1 edu.ou.asgbook.rasterization.BoundingBox.BoundingBox (LatLon[ ] *vertices*)

```
                                         {
        ScalarStatistic lat = new ScalarStatistic();
        ScalarStatistic lon = new ScalarStatistic();
        for (int i=0; i < vertices.length; ++i){
            lat.update(vertices[i].getLat());
            lon.update(vertices[i].getLon());
        }
        maxx = lat.getMax();
        maxy = lon.getMax();
        minx = lat.getMin();
        miny = lon.getMin();
    }
```

6.4.3 Member Function Documentation

6.4.3.1 boolean edu.ou.asgbook.rasterization.BoundingBox.contains (double *x*, double *y*)

```
                                              {
        return (x >= minx && x <= maxx && y >= miny && y <= maxy);
    }
```

6.4.3.2 static BoundingBox edu.ou.asgbook.rasterization.BoundingBox.copyOf (BoundingBox *a*) [static]

```
                                                  {
        BoundingBox b = new BoundingBox();
        b.minx = a.minx;
        b.maxx = a.maxx;
        b.miny = a.miny;
        b.maxy = a.maxy;
        return b;
    }
```

6.4.3.3 void edu.ou.asgbook.rasterization.BoundingBox.update (BoundingBox *a*)

```
                                        {
        minx = Math.min(minx, a.minx);
        miny = Math.min(miny, a.miny);
        maxx = Math.max(maxx, a.maxx);
        maxy = Math.max(maxy, a.maxy);
    }
```

6.5 edu.ou.asgbook.rasterization.CatmullRom Class Reference

A Catmull-Rom spline, a local spline.

Static Public Member Functions

- static double[ ] interpolate (double[ ] x1, double[ ] y1, double[ ] x2)

  Determines the y coordinates for the given x2 by interpolating the spline control points (x1,y1).

- static double[ ] sort_and_interpolate (double[ ] x1, double[ ] y1, double[ ] x2)
- static List< Pixel > getPositionIn (double[ ] controllat, double[ ] controllon, LatLonGrid grid)
- static void main (String args[ ]) throws Exception

Classes

- class **XtoY**

6.5.1 Detailed Description

A Catmull-Rom spline, a local spline.

Author:

valliappa.lakshmanan

6.5.2 Member Function Documentation

6.5.2.1 static List<Pixel> edu.ou.asgbook.rasterization.CatmullRom.getPositionIn (double[ ] *controllat*, double[ ] *controllon*, LatLonGrid *grid*) [static]

```

        List<Pixel> result = new ArrayList<Pixel>();
        // we want to find the intersection at all the lat of the grid
        double[] lat2 = new double[grid.getNumLat()];
        for (int i=0; i < lat2.length; ++i){
            lat2[i] = grid.getLocation(i, 0).getLat(); // lat of row
        }
        double[] lon2 = sort_and_interpolate(controllat, controllon, lat2);
        for (int i=0; i < lon2.length; ++i){
            int col = grid.getCol(new LatLon(lat2[i],lon2[i])); // col to fill in
            result.add(new Pixel(i,col,grid.getValue(i,col)));
        }
        return result;
    }
```

6.5.2.2 static double [ ] edu.ou.asgbook.rasterization.CatmullRom.interpolate (double[ ] *x1*, double[ ] *y1*, double[ ] *x2*) `[static]`

Determines the y coordinates for the given x2 by interpolating the spline control points (x1,y1).

The control points need to be sorted in x.

```
                                                                              {
        // result: initialize at lower-bound value
        if ( x1.length == 0 ) return new double[x2.length];
        double[] y2 = new double[x2.length];
        for (int i=0; i < y2.length; ++i){
            y2[i] = y1[0];
        }

        // every interval is p1 <= p2 <= p3 where p2 is resampling position
        double p3 = x2[0] - 1; // below first value
        for (int i=0; i < x2.length; ++i){
            // find interval which contains p2
            double p2 = x2[i];
            if ( p2 <= x1[0]     ){ y2[i] = y1[0]; continue; }
            if ( p2 >= back(x1) ){ y2[i] = back(y1); continue; }
            int j = 0;
            while (j < (int)x1.length && p2 > x1[j]){ ++j; }
            --j; //if ( p2 < x1[j] ) --j;

            double p1 = x1[j];
            p3 = x1[j+1];

            // j and j+1 will be in bounds but j-1 and j+2 may not be
            int j1 = j-1; if (j1 < 0) j1 = 0;
            int j2 = j+2; if (j2 > (x1.length-1)) j2 = x1.length-1;

            // spline
            double dx  = 1.0f / (p3 - p1);
            double dx1 = 1.0f / (p3 - x1[j1]);
            double dx2 = 1.0f / (x1[j2] - p1);
            double dy  = (y1[j+1] - y1[j]) * dx;
            double yd1 = (y1[j+1] - y1[j1]) * dx1;
            double yd2 = (y1[j2] - y1[j]) * dx2;
            double a0y =  y1[j];
            double a1y =  yd1;
            double a2y =  dx *  ( 3*dy - 2*yd1 - yd2);
            double a3y =  dx*dx*(-2*dy +   yd1 + yd2);

            // cubic polynomial
            double x = p2 - p1;
            y2[i] = ((a3y*x + a2y)*x + a1y)*x + a0y;
        }
        return y2;
    }
```

6.5.2.3 static void edu.ou.asgbook.rasterization.CatmullRom.main (String *args*[ ]) throws Exception [static]

```
                                                         {
         LatLonGrid grid = new LatLonGrid(100,100,0,new LatLon(100,-90),0.01,0.01);
         double[] controlx = new double[]{99.4,99.3,99.5,99.7};
         double[] controly = new double[]{-89.5,-89.3,-89.6,-89.4};

         List<Pixel> pixels = getPositionIn(controlx, controly, grid);
         for (Pixel p : pixels){
             grid.setValue(p.getRow(), p.getCol(), 10);
         }

         File out = OutputDirectory.getDefault("raster");
         KmlWriter.write(grid, out, "drawspline", PngWriter.createCoolToWarmColormap());
     }
```

6.5.2.4 static double [ ] edu.ou.asgbook.rasterization.CatmullRom.sort_-and_interpolate (double[ ] *x1*, double[ ] *y1*, double[ ] *x2*) [static]

```
                                                                                    {

    int N = x1.length;

    // create structure for sorting
    XtoY[] data = new XtoY[N];
    for (int i=0; i < N; ++i){
        data[i] = new XtoY();
        data[i].orig_index = i;
        data[i].scaledx = (int)Math.round(x1[i] * 1000 + 0.5);
         data[i].scaledy = (int)Math.round(y1[i] * 1000 + 0.5);
     }
     Arrays.sort(data);

     // create input data
     double[] x= new double[N];
     double[] y= new double[N];
     int curr = 0;
     for (int i=0; i < N; ++i){
         // if you have two or more y values for same x, then use avg
         int start_i = i;
         while ( (i+1) < N && data[i].scaledx == data[i+1].scaledx ){
             ++i;
         }
         x[curr] = x1[data[i].orig_index];
         double sumy = 0;
```

```
            for (int k=start_i; k <= i; ++k){
                sumy += y1[data[k].orig_index];
            }
            y[curr] = sumy/(i-start_i+1);
            ++curr;
        }

        x = Arrays.copyOf(x, curr);
        y = Arrays.copyOf(y, curr);

        // call interpolate
        return interpolate( x, y, x2 );
    }
```

6.6 edu.ou.asgbook.datamining.CityCategories Class Reference

Obtains city data for clustering.

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.6.1 Detailed Description

Obtains city data for clustering.

Author:

valliappa.lakshmanan

6.6.2 Member Function Documentation

6.6.2.1 static void edu.ou.asgbook.datamining.CityCategories.main (String[ ] *args*) throws Exception [static]

```
                                                            {
         // create output directory
         File out = OutputDirectory.getDefault("citycategories");
         final boolean SMALL = true;

         // read input (crop to cover China)
         LatLonGrid pop    = GlobalPopulation.read(GlobalPopulation.WORLD);
         if (SMALL){
             pop = pop.crop(900, 6000, 800, 1600); // China mainly
         }
         KmlWriter.write(pop, out, "modelpop", PngWriter.createRandomColormap());

         LatLonGrid nightTimeLights = NightimeLights.read(NightimeLights.WORLD).remapTo(pop);
         KmlWriter.write(nightTimeLights, out, "modellights", PngWriter.createCoolToWarmColormap());

         EnhancedWatershedSegmenter seg = new EnhancedWatershedSegmenter(10, 1, 600, 10, 5);
         LabelResult allcities = seg.label(pop);
         KmlWriter.write(allcities.label, out, "modelcities", PngWriter.createRandomColormap());

         // write out cluster file
         String filename = out.getAbsolutePath()+"/citydata.txt";
         PrintWriter writer = new PrintWriter(new FileWriter(filename));
         writer.println("Pop light");
         RegionProperty[] population = RegionProperty.compute(allcities, pop);
         RegionProperty[] lighting = RegionProperty.compute(allcities, nightTimeLights);
         for (int i=1; i < population.length; ++i){
             writer.println(population[i].getCval() + " " + lighting[i].getCval());
         }
         writer.close();
         System.out.println("Wrote " + filename);

         // compute the category of each (based on clustering result)
         int[] categories = new int[population.length];
         for (int i=1; i < categories.length; ++i){
             categories[i] = computeCategory( population[i].getCval(), lighting[i].getCval()
         }
         LatLonGrid result = LatLonGrid.copyOf(allcities.label);
         result.setMissing(0);
         result.fill(result.getMissing());
         for (int i=0; i < result.getNumLat(); ++i){
             for (int j=0; j < result.getNumLon(); ++j){
                 int cityno = allcities.label.getValue(i,j);
                 if ( cityno > 0 ){
                     result.setValue(i,j, categories[cityno]);
                 }
             }
         }
         KmlWriter.write(result, out, "citycategories", PngWriter.createCoolToWarmColormap()
     }
```

6.7 edu.ou.asgbook.datamining.CityGdiModels Class Reference

Applies different data mining models to each city.

Static Public Member Functions

- static double[ ][ ] findPatterns (LabelResult cities, LatLonGrid population, LatLonGrid nightTimeLights)
- static int[ ] applyLinearModel (double[ ][ ] pattern)
- static int[ ] applyDecisionTree (double[ ][ ] pattern)
- static int[ ] applyNeuralNetwork (double[ ][ ] pattern)
- static void main (String[ ] args) throws Exception

6.7.1 Detailed Description

Applies different data mining models to each city.

Author:

valliappa.lakshmanan

6.7.2 Member Function Documentation

6.7.2.1 static int [ ] edu.ou.asgbook.datamining.CityGdiModels.applyDecisionTree (double *pattern*[ ][ ]) [static]

```
                                                              {
         int[] result = new int[ pattern.length ];
         for (int i=0; i < pattern.length; ++i){
             double pop = pattern[i][0];
             double light = pattern[i][1];
             if (light < 48.91){
                 if (light < 17.61){
                     result[i] = 0;
                 } else {
                     result[i] = 1;
                 }
             } else {
                 if ( light < 81.25 ){
                     if ( pop >= 31.77 ){
                         result[i] = 1;
                     } else {
                         result[i] = 2;
                     }
```

```
                } else {
                    if ( pop >= 105.7 ){
                        result[i] = 2;
                    } else {
                        result[i] = 4;
                    }
                }
            }
        }
        return result;
    }
```

6.7.2.2 static int [ ] edu.ou.asgbook.datamining.CityGdiModels.applyLinearModel (double *pattern*[ ][ ]) [static]

```
                                                    {
        int[] result = new int[ pattern.length ];
        for (int i=0; i < pattern.length; ++i){
            result[i] = (int) Math.round(0.003494 + 0.034444 * pattern[i][1] - 0.005992 * pa
        }
        return result;
    }
```

6.7.2.3 static int [ ] edu.ou.asgbook.datamining.CityGdiModels.applyNeuralNetwork (double *pattern*[ ][ ]) [static]

```
                                                      {
        int[] result = new int[ pattern.length ];
        for (int i=0; i < pattern.length; ++i){
            result[i] = (int) Math.round(100 * probOfRichNN(pattern[i][0], pattern[i][1]));
        }
        return result;
    }
```

6.7.2.4 static double [ ][ ] edu.ou.asgbook.datamining.CityGdiModels.findPatterns (LabelResult *cities*, LatLonGrid *population*, LatLonGrid *nightTimeLights*) [static]

```

        // make sure that this is identical to GdiPattern.java
        RegionProperty[] pop = RegionProperty.compute(cities, population);
        RegionProperty[] lights = RegionProperty.compute(cities, nightTimeLights);

        double[][] patterns = new double[pop.length][2];
        for (int i=1; i < patterns.length; ++i){
            patterns[i][0] = pop[i].getCval();
            patterns[i][1] = lights[i].getCval();
        }
        return patterns;
    }
```

6.7.2.5 static void edu.ou.asgbook.datamining.CityGdiModels.main (String[ ] *args*) throws Exception [static]

```
                                                            {
        // create output directory
        File out = OutputDirectory.getDefault("gdimodels");
        final boolean SMALL = true;

        // read input (crop to cover China)
        LatLonGrid pop    = GlobalPopulation.read(GlobalPopulation.WORLD);
        if (SMALL){
            pop = pop.crop(900, 6000, 800, 1600); // China mainly
        }
        KmlWriter.write(pop, out, "modelpop", PngWriter.createRandomColormap());

        LatLonGrid nightTimeLights = NightimeLights.read(NightimeLights.WORLD).remapTo(pop);
        KmlWriter.write(nightTimeLights, out, "modellights", PngWriter.createCoolToWarmColormap());

        EnhancedWatershedSegmenter seg = new EnhancedWatershedSegmenter(10, 1, 600, 10, 5);
        LabelResult allcities = seg.label(pop);
        KmlWriter.write(allcities.label, out, "modelcities", PngWriter.createRandomColormap());

        // compute gdi for each city
        double[][] patterns = findPatterns(allcities, pop, nightTimeLights);
        String[] models = {"linear", "tree", "nn" };
        for (String model : models){
            int[] modelresult = null;
            if (model.equals("linear")){
                modelresult = applyLinearModel(patterns);
            } else if (model.equals("tree")){
                modelresult = applyDecisionTree(patterns);
            } else if (model.equals("nn")){
                modelresult = applyNeuralNetwork(patterns);
            }
            LatLonGrid result = LatLonGrid.copyOf(allcities.label);
            result.setMissing(WorldBankGDI.DevelopmentCategory.Unknown.ordinal());
            result.fill( result.getMissing() );
            for (int i=0; i < result.getNumLat(); ++i){
                for (int j=0; j < result.getNumLon(); ++j){
                    int cityno = allcities.label.getValue(i,j);
                    if (cityno > 0 ){
                        result.setValue(i, j, modelresult[cityno]);
                    }
                }
            }
            KmlWriter.write(result, out, model+"gdi", PngWriter.createCoolToWarmColormap());
        }
    }
```

6.8 edu.ou.asgbook.segmentation.ContiguityEnhancedKMeansSegmenter Class Reference

Objects consist of pixels that are grown from initial centers using K-means.

Public Member Functions

- ContiguityEnhancedKMeansSegmenter (int min_thresh, int seed_value, int max_data_difference, int max_cluster_size)

  KMeans is seeded from points > seed_value, so pass in a high enough value here Only pixels > min_thresh are eligible to be part of an object.

- List< LabelResult > label (LatLonGrid data)

  Creates a labeled grid where background pixels are set to 0 and labels for objects go 1,2,3.

Static Public Member Functions

- static void main (String[ ] args) throws Exception

Classes

- class **Cluster**

6.8.1 Detailed Description

Objects consist of pixels that are grown from initial centers using K-means.

Author:

v.lakshmanan

6.8.2 Constructor & Destructor Documentation

6.8.2.1 edu.ou.asgbook.segmentation.ContiguityEnhancedKMeansSegmenter.ContiguityEnhancedKMeansSegmenter (int *min_thresh*, int *seed_value*, int *max_data_difference*, int *max_cluster_size*)

KMeans is seeded from points > seed_value, so pass in a high enough value here Only pixels > min_thresh are eligible to be part of an object.

```

        this.START_THRESH = seed_value;
        this.MIN_THRESH = min_thresh;
        this.MAX_DATA_DIFFERENCE = max_data_difference;
        this.MAX_CLUSTER_SIZE = max_cluster_size;
    }
```

6.8.3 Member Function Documentation

6.8.3.1 List<LabelResult> edu.ou.asgbook.segmentation.Contiguity-EnhancedKMeansSegmenter.label (LatLonGrid *data*)

Creates a labeled grid where background pixels are set to 0 and labels for objects go 1,2,3.

.. All pixels > thresh are part of an object. Returns a LabelResult for each iteration, with the last one being the final result.

```
                                                   {
        List<LabelResult> result = new ArrayList<LabelResult>();
        final int nrows = data.getNumLat();
        final int ncols = data.getNumLon();

        // initialize based on simple thresholding at a high value
        final ThresholdSegmenter seeder = new ThresholdSegmenter(START_THRESH);
        LabelResult seed = seeder.label(data);
        result.add(seed); // first one

        // Start K-means
        int iter = 1;
        int n_changed = 0;
        do {
            // compute means
            Cluster[] clusters = findClusters(data, seed);
            // move pixels
            LabelResult next = new LabelResult(LatLonGrid.copyOf(seed.label), seed.maxlabel);
            n_changed = 0;
            for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
                if ( data.getValue(i,j) != data.getMissing() && data.getValue(i,j) > MIN_THRESH ){
                    int closest = findClosestCluster(data.getValue(i,j),i,j,seed.label, clusters);
                    if (closest != seed.label.getValue(i,j)){
                        // change the label to closest
                        next.label.setValue(i, j, closest);
                        ++n_changed;
                    }
                }
            }
            System.out.println("Changing " + n_changed + " at " + iter + " th iteration");
            // for next step
            seed = next;
            result.add(seed);
            ++iter;
        } while (iter < MAX_ITER && n_changed > 0);
        return result;
    }
```

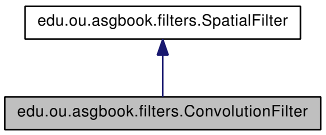
6.8.3.2 static void edu.ou.asgbook.segmentation.ContiguityEnhanced-KMeansSegmenter.main (String[ ] *args*) throws Exception [static]

```
                                                          {
        File out = OutputDirectory.getDefault("contigkmeans");

        LatLonGrid grid = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new GlobalPo
        KmlWriter.write(grid, out, "orig", PngWriter.createCoolToWarmColormap());

        // label image based on threshold
        List<LabelResult> labels =  new ContiguityEnhancedKMeansSegmenter(10,20,100,10).lab
        for (int i=0; i < labels.size(); ++i){
            LatLonGrid label = labels.get(i).label;
            KmlWriter.write(label, out, "label_" + i, PngWriter.createCoolToWarmColormap()
        }
    }
```

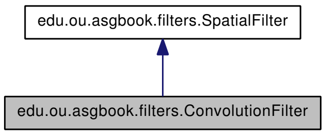
6.9 edu.ou.asgbook.filters.ConvolutionFilter Class Reference

Convolve an image by a window.

Inheritance diagram for edu.ou.asgbook.filters.ConvolutionFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.ConvolutionFilter

Collaboration diagram for edu.ou.asgbook.filters.ConvolutionFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.ConvolutionFilter

Public Member Functions

- ConvolutionFilter (double[ ][ ] coeffs)
- int getFilterNumRows ()
- int getFilterNumCols ()
- LatLonGrid smooth (final LatLonGrid input)

    Uses weights, but only at non-missing pixels, and divides by the total weight.

- LatLonGrid convolve (final LatLonGrid input)

    Uses the coefficients and returns the convolved value without dividing by sum of weights Use this for non-smoothing coefficients.

- Override LatLonGrid filter (LatLonGrid input)

Static Public Member Functions

- static double[ ][ ] boxcar (int numx, int numy)
- static double[ ][ ] gauss (int numx, int numy)
- static double[ ][ ] gauss (int numx, int numy, double sigmax, double sigmay)
- static void main (String[ ] args) throws Exception

6.9.1 Detailed Description

Convolve an image by a window.

Author:

Valliappa.Lakshmanan

6.9.2 Constructor & Destructor Documentation

6.9.2.1 edu.ou.asgbook.filters.ConvolutionFilter.ConvolutionFilter (double *coeffs*[ ][ ])

```
                                            {
       this.coeffs = coeffs;
       if ( coeffs.length % 2 == 0 || coeffs[0].length % 2 == 0 ){
           throw new IllegalArgumentException("Dimensions of coefficients array needs to be
       }
   }
```

6.9.3 Member Function Documentation

6.9.3.1 static double [ ][ ] edu.ou.asgbook.filters.ConvolutionFilter.boxcar (int *numx*, int *numy*) [static]

```
                                                    {
        double[][] coeffs = new double[numx][numy];

        double tot = numx * numy;
        for (int i=0; i < coeffs.length; ++i){
            for (int j=0; j < coeffs.length; ++j){
                coeffs[i][j] = 1 / tot;
            }
        }

        return coeffs;
    }
```

6.9.3.2 LatLonGrid edu.ou.asgbook.filters.ConvolutionFilter.convolve (final LatLonGrid *input*)

Uses the coefficients and returns the convolved value without dividing by sum of weights Use this for non-smoothing coefficients.

```
                                                  {
       LatLonGrid output = LatLonGrid.copyOf(input);
       output.fill(output.getMissing());
```

```
        int[][] outData = output.getData();
        int[][] inData = input.getData();
        final int hx = coeffs.length / 2;
        final int hy = coeffs[0].length / 2;
        final int nx = output.getNumLat();
        final int ny = output.getNumLon();
        for (int i=hx; i < (nx-hx); ++i){
            for (int j=hy; j < (ny-hy); ++j){
                double tot = 0;
                for (int m=-hx; m <= hx; ++m){
                    for (int n=-hy; n <= hy; ++n){
                        double coeff = coeffs[m+hx][n+hy];
                        int inval = inData[i+m][j+n];
                        if (inval != input.getMissing()){
                            tot += inval*coeff;
                        }
                    }
                }
                outData[i][j] = (int) Math.round(tot);
            }
        }
        return output;
    }
```

6.9.3.3 Override LatLonGrid edu.ou.asgbook.filters.ConvolutionFilter.filter (LatLonGrid *input*)

```
                                                    {
        return convolve(input);
    }
```

6.9.3.4 static double [ ][ ] edu.ou.asgbook.filters.ConvolutionFilter.gauss (int *numx*, int *numy*, double *sigmax*, double *sigmay*) [static]

```
                                                                                                   {
        double[][] coeffs = new double[numx][numy];

        for (int i=0; i < coeffs.length; ++i){
            for (int j=0; j < coeffs.length; ++j){
                double x = (i - coeffs.length/2.0)/sigmax;
                double y = (j - coeffs[0].length/2.0)/sigmay;
                coeffs[i][j] = Math.exp(-(x*x + y*y));
            }
        }

        return coeffs;
    }
```

6.9.3.5 static double [ ][ ] edu.ou.asgbook.filters.ConvolutionFilter.gauss (int *numx*, int *numy*) [static]

```
                                                      {
        return gauss(numx, numy, numx/6.0, numy/6.0); // 3-sigma on either side
    }
```

6.9.3.6 int edu.ou.asgbook.filters.ConvolutionFilter.getFilterNumCols ()

```
                                  {
        return coeffs[0].length;
    }
```

6.9.3.7 int edu.ou.asgbook.filters.ConvolutionFilter.getFilterNumRows ()

```
                                  {
        return coeffs.length;
    }
```

6.9.3.8 static void edu.ou.asgbook.filters.ConvolutionFilter.main (String[ ] *args*) throws Exception [static]

```
                                                            {
        // create output directory
        File out = OutputDirectory.getDefault("convolve");

        // read input
        LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new G
        KmlWriter.write(popdensity, out, "orig", PngWriter.createCoolToWarmColormap());

        // average
        {
            ConvolutionFilter filter = new ConvolutionFilter(ConvolutionFilter.boxcar(3, 3
            LatLonGrid sm = filter.smooth(popdensity);
            KmlWriter.write(sm, out, "boxcar1", PngWriter.createCoolToWarmColormap());
        }

        // boxcar
        {
            ConvolutionFilter filter = new ConvolutionFilter(ConvolutionFilter.boxcar(5, 5
            LatLonGrid sm = filter.smooth(popdensity);
            KmlWriter.write(sm, out, "boxcar", PngWriter.createCoolToWarmColormap());
        }

        // gauss
        {
            ConvolutionFilter filter = new ConvolutionFilter(ConvolutionFilter.gauss(11, 1
            LatLonGrid sm = filter.smooth(popdensity);
            KmlWriter.write(sm, out, "gauss", PngWriter.createCoolToWarmColormap());
        }
    }
```

6.9.3.9 LatLonGrid edu.ou.asgbook.filters.ConvolutionFilter.smooth (final LatLonGrid *input*)

Uses weights, but only at non-missing pixels, and divides by the total weight.

Use this for smoothing

```
                                                    {
        LatLonGrid output = LatLonGrid.copyOf(input);
        output.fill(output.getMissing());
        int[][] outData = output.getData();
        int[][] inData = input.getData();
        final int hx = coeffs.length / 2;
        final int hy = coeffs[0].length / 2;
        final int nx = output.getNumLat();
        final int ny = output.getNumLon();
        for (int i=hx; i < (nx-hx); ++i){
            for (int j=hy; j < (ny-hy); ++j){
                double tot = 0;
                double wt = 0;
                for (int m=-hx; m <= hx; ++m){
                    for (int n=-hy; n <= hy; ++n){
                        double coeff = coeffs[m+hx][n+hy];
                        int inval = inData[i+m][j+n];
                        if (inval != input.getMissing()){
                            tot += inval*coeff;
                            wt += coeff;
                        }
                    }
                }
                if ( wt > 0 ){
                    outData[i][j] = (int)( Math.round(tot / wt) );
                }
            }
        }
        return output;
    }
```

6.10 edu.ou.asgbook.dataset.CountryPolygons Class Reference

Reads country-by-country coordinates from a KML placemarks file.

Static Public Member Functions

- static Country[ ] readKml (File file) throws Exception

    reads data from a File.

- static LatLonGrid readGrid (File file) throws Exception
- static LatLonGrid asLatLonGrid (Country[ ] countries, double latres, double lon-res)
- static void main (String[ ] args) throws Exception

Static Public Attributes

- static File WORLD_KML = new File("data/countries/countries_world.kml")
- static File WORLD_GRID = new File("data/countries/countries_world.txt.gz")

Classes

- class **Country**

6.10.1 Detailed Description

Reads country-by-country coordinates from a KML placemarks file.

Author:

v.lakshmanan

6.10.2 Member Function Documentation

6.10.2.1 static LatLonGrid edu.ou.asgbook.dataset.CountryPolygons.as-LatLonGrid (Country[ ] *countries*, double *latres*, double *lonres*) `[static]`

```

        int nrows = (int) Math.round(180 / latres);
        int ncols = (int) Math.round(360 / lonres);
        LatLon nwCorner = new LatLon(90,-180);
        LatLonGrid result = new LatLonGrid(nrows, ncols, -1, nwCorner, latres, lonres);
        for (int i=0; i < nrows; ++i){
            for (int j=0; j < ncols; ++j){
                LatLon pt = result.getLocation(i, j);
                result.setValue(i,j, result.getMissing());
                for (int c = 0; c < countries.length; ++c){
                    if (countries[c].contains(pt)){
                        result.setValue(i, j, c);
                        break;
                    }
                }
            }
            System.out.println("row " + i + " computed.");
        }
        return result;
    }
```

6.10.2.2 static void edu.ou.asgbook.dataset.CountryPolygons.main (String[ ] *args*) throws Exception [static]

```
                                                              {
        CountryPolygons.Country[] countries = CountryPolygons.readKml(CountryPolygons.WORLD_KML);
        for (CountryPolygons.Country c : countries){
            System.out.println(c);
        }

        List<LatLon> cities = new ArrayList<LatLon>();
        cities.add(new LatLon(35, -97.1)); // Norman, Oklahoma
        cities.add(new LatLon(40,33)); // Istanbul, Turkey
        cities.add(new LatLon(-34,151)); // Sydney, Australia
        cities.add(new LatLon(-23.5,-46.5)); // Rio, Brazil
        for (LatLon city : cities){
            System.out.println("Looking for " + city);
            for (CountryPolygons.Country c : countries){
                if (c.contains(city)){
                    System.out.println(city + " is in " + c);
                    break;
                }
            }
        }
        System.out.println("Finished place search using country list");

        File out = OutputDirectory.getDefault("countries");
        LatLonGrid grid = asLatLonGrid(countries, 0.1, 0.1);
        for (LatLon city : cities){
            int country = grid.getValue(city);
            if (country >= 0){
                System.out.println("Location " + city + " is in " + countries[country]);
            } else {
                System.out.println("Location " + city + " is unclaimed");
            }
        }

        KmlWriter.write(grid, out, "countries", PngWriter.createRandomColormap());
        EsriGrid.write(grid, out, "countries.txt.gz");
        EsriGrid.write(grid, CountryPolygons.WORLD_GRID);


/*      // combine with GDI ...
        List<CountryPolygons.Country> notfound = new ArrayList<CountryPolygons.Country>();
        WorldBankGDI.Lookup gdicountries = WorldBankGDI.readAsMap(WorldBankGDI.WORLD);
        for (CountryPolygons.Country c : countries){
            WorldBankGDI.Country match = gdicountries.get(c.name);
            System.out.println(c + " " + match);
            if ( match == null ){
                notfound.add(c);
            }
        }
        for (CountryPolygons.Country c : notfound){
            System.out.println("Not found: " + c);
        }*/
    }
```

6.10.2.3 static LatLonGrid edu.ou.asgbook.dataset.CountryPolygons.readGrid (File *file*) throws Exception [static]

```
                                                                   {
        return EsriGrid.read(file, new LinearScaling(1));
    }
```

6.10.2.4 static Country [ ] edu.ou.asgbook.dataset.CountryPolygons.readKml (File *file*) throws Exception [static]

reads data from a File.

The File can be gzipped or uncompressed.

```
                                                                 {
        InputStream f = null;
        System.out.println("Reading " + file.getAbsolutePath());
        f = new FileInputStream(file);
        if (file.getAbsolutePath().endsWith(".gz")) {
            f = new GZIPInputStream(f);
        }
        return readKml(f);
    }
```

6.10.3 Member Data Documentation

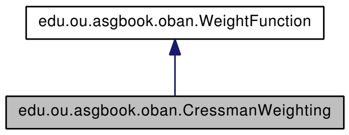
6.10.3.1 File edu.ou.asgbook.dataset.CountryPolygons.WORLD_GRID = new File("data/countries/countries_world.txt.gz") `[static]`

6.10.3.2 File edu.ou.asgbook.dataset.CountryPolygons.WORLD_KML = new File("data/countries/countries_world.kml") `[static]`

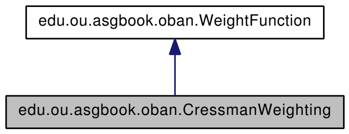
6.11 edu.ou.asgbook.oban.CressmanWeighting Class Reference

An interpolation method that uses 1/r^2.

Inheritance diagram for edu.ou.asgbook.oban.CressmanWeighting:


edu.ou.asgbook.oban.WeightFunction
edu.ou.asgbook.oban.CressmanWeighting

Collaboration diagram for edu.ou.asgbook.oban.CressmanWeighting:


edu.ou.asgbook.oban.WeightFunction
edu.ou.asgbook.oban.CressmanWeighting

Public Member Functions

- CressmanWeighting (double radiusOfInfluence)
- Override double computeWt (double latdist, double londist)

  Subclasses implement a weighting function.

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.11.1 Detailed Description

An interpolation method that uses 1/r^2.

Author:

Valliappa.Lakshmanan

6.11.2 Constructor & Destructor Documentation

6.11.2.1 edu.ou.asgbook.oban.CressmanWeighting.CressmanWeighting (double *radiusOfInfluence*)

Parameters:

radiusOfInfluence Set extent of influence in degrees

```
                                                         {
           this.R2 = radiusOfInfluence * radiusOfInfluence;
       }
```

6.11.3 Member Function Documentation

6.11.3.1 Override double edu.ou.asgbook.oban.Cressman-Weighting.computeWt (double *latdist*, double *londist*) [virtual]

Subclasses implement a weighting function.

If -ve value is returned, then the point will be considered too far away and not used in weighting.

Implements edu.ou.asgbook.oban.WeightFunction.

```
                                                             {
           double r2 = latdist * latdist + londist * londist;
           if ( r2 > R2 ){
               return INVALID_WEIGHT;
           }
           double factor = r2/R2;
           return (1 - factor)/(1 + factor);
       }
```

6.11.3.2 static void edu.ou.asgbook.oban.CressmanWeighting.main (String[ ] *args*) throws Exception [static]

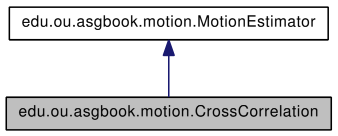
```
                                                              {
           PointObservations data = DailyRainfall.read(DailyRainfall.TN_Oct2010);

           double meansep = ObjectiveAnalysisUtils.computeMeanDistance(data);
           System.out.println("Objectively analyzing " + data.getPoints().length + " pts with a mean separ
           WeightFunction wtFunc = new CressmanWeighting(3*meansep);
           WeightedAverage analyzer = new WeightedAverage(wtFunc, 0.01, 0.01, 1);

           long startTime = System.nanoTime();
           final int numPasses = 2;
           LatLonGrid grid = analyzer.analyze(data, numPasses, 0, data.getMaxValue());
```

```
        System.out.println("Took " + (System.nanoTime() - startTime)/(1000*1000.0*1000) + "

        // write output
        File out = OutputDirectory.getDefault("cressman");
        KmlWriter.write(grid, out, "Precip24H", PngWriter.createCoolToWarmColormap());
    }
```

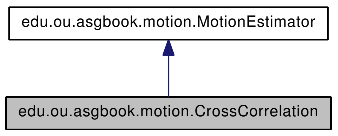
6.12 edu.ou.asgbook.motion.CrossCorrelation Class Reference

Estimates motion using cross-correlation.

Inheritance diagram for edu.ou.asgbook.motion.CrossCorrelation:


edu.ou.asgbook.motion.MotionEstimator
edu.ou.asgbook.motion.CrossCorrelation

Collaboration diagram for edu.ou.asgbook.motion.CrossCorrelation:


edu.ou.asgbook.motion.MotionEstimator
edu.ou.asgbook.motion.CrossCorrelation

Public Member Functions

- CrossCorrelation (int est_halfsize_x, int est_halfsize_y, int maxmotion_x, int maxmotion_y)

  Pass size of window to estimate motion of, and the maximum movement in the two directions.

- Override Pair< LatLonGrid, LatLonGrid > compute (LatLonGrid data0, LatLonGrid data1, File outdir)

  returns motion in the two directions.

Static Public Member Functions

- static void test () throws Exception
- static void main (String[ ] args) throws Exception

6.12.1 Detailed Description

Estimates motion using cross-correlation.

Author:

v.lakshmanan

6.12.2 Constructor & Destructor Documentation

6.12.2.1 edu.ou.asgbook.motion.CrossCorrelation.CrossCorrelation (int *est_halfsize_x*, int *est_halfsize_y*, int *maxmotion_x*, int *maxmotion_y*)

Pass size of window to estimate motion of, and the maximum movement in the two directions.

Parameters:

est_halfsize_x

est_halfsize_y

```
                                                    {
        super();
        EST_HALFSIZE_NS = est_halfsize_x;
        EST_HALFSIZE_EW = est_halfsize_y;
        MAX_U = maxmotion_x;
        MAX_V = maxmotion_y;
        MIN_FILL_PIXELS = (int) Math.round(MIN_FILL_RATIO * (2*EST_HALFSIZE_NS+1) * (2*EST_
    }
```

6.12.3 Member Function Documentation

6.12.3.1 Override Pair<LatLonGrid, LatLonGrid> edu.ou.asgbook.motion.CrossCorrelation.compute (LatLonGrid *data0*, LatLonGrid *data1*, File *outdir*)

returns motion in the two directions.

The first one is north to south and the second one is east to west. The data is aligned to second time frame. The output dir is used for intermediate products and may be null.

Implements edu.ou.asgbook.motion.MotionEstimator.

```

        // Grids we need. initialize all of them at zero
        final int nrows = data1.getNumLat();
        final int ncols = data1.getNumLon();
        LatLonGrid u = new LatLonGrid(nrows, ncols, 0, data1.getNwCorner(), data1.getLatRes
        LatLonGrid v = LatLonGrid.copyOf(u);

        System.out.println("Computing u,v using xcorr on " + nrows + "x" + ncols + " image;

        // compute u,v for every pixel
        double meanu = 0;
        double meanv = 0;
        int nestimates = 0;
        for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
            // at pixel, search best match for entire neighborhood
            // best_m, best_n are not changed from default unless < error_ratio
            double lse = MAX_ERROR_RATIO;
            int best_m = 0;
            int best_n = 0;
            for (int m=-MAX_U; m <= MAX_U; ++m){
                for (int n=-MAX_V; n <= MAX_V; ++n){
                    double error = compute_error(data0, data1, i, j, m, n);
                    if ( error < lse ){
                        lse = error;
                        best_m = m;
                        best_n = n;
                    }
                }
            }
            u.setValue(i,j, best_m);
            v.setValue(i,j, best_n);

            if ( lse != MAX_ERROR_RATIO ){
                meanu += best_m;
                meanv += best_n;
                ++nestimates;
            }

            if ( i%10 == 0 && j == 0){
                System.out.println( (100*i)/nrows + "% of pixels complete.");
            }
        }

        System.out.println("Mean motion vector: u=" + meanu/(nestimates) + " v=" + meanv/(nestimates));
        return new Pair<LatLonGrid,LatLonGrid>(u,v);
    }
```

6.12.3.2 static void edu.ou.asgbook.motion.CrossCorrelation.main (String[ ] *args*) throws Exception [static]

```
                                                                {
        // test();
        // create output directory
        File out = OutputDirectory.getDefault("xcorr");

        // read
        File f = new File("data/seviri");
        Pair<LatLonGrid,Date>[] grids = SeviriInfraredTemperature.readAll(f);

        // do alg
        CrossCorrelation alg = new CrossCorrelation(3,3,5,5);
        Pair<LatLonGrid,LatLonGrid> motion = alg.compute(grids[0].first, grids[1].first, out);

        // write
        KmlWriter.write(motion.first, out, "xcorr_u", PngWriter.createCoolToWarmColormap());
```

```
    KmlWriter.write(motion.second, out, "xcorr_v", PngWriter.createCoolToWarmColormap()

    // align and compute difference
    LatLonGrid diff = new AlignAndDifference().compute(grids[0].first, grids[1].first,
    KmlWriter.write(diff, out, "xcorr_diff", PngWriter.createCoolToWarmColormap());
}
```

6.12.3.3 static void edu.ou.asgbook.motion.CrossCorrelation.test () throws Exception [static]

```
                                                {
    // because the alignment doesn't really check lat-lon extents,
    // cropping from offset corners will look like translation ...
    LatLonGrid conus = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA);
    LatLonGrid[] grids = new LatLonGrid[2];
    grids[0] = conus.crop(900, 2500, 256, 256);
    grids[0].setMissing(0);
    int motx = -2; int moty = -3;
    grids[1] = conus.crop(900-motx, 2500-moty, 256, 256);
    grids[1].setMissing(0);
    CrossCorrelation alg = new CrossCorrelation(5,5,Math.abs(2*motx),Math.abs(2*moty));
    alg.compute(grids[0], grids[1], null);

    System.exit(0);
}
```

6.13 edu.ou.asgbook.histogram.Cumulative-DistributionFunction Class Reference

Forms a CDF from a Histogram.

Public Member Functions

- CumulativeDistributionFunction (Histogram hist)
- int getMin ()
- int getIncr ()
- float[ ] getProb ()
- Override String toString ()

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.13.1 Detailed Description

Forms a CDF from a Histogram.

Author:

v.lakshmanan

6.13.2 Constructor & Destructor Documentation

6.13.2.1 edu.ou.asgbook.histogram.CumulativeDistribution-Function.CumulativeDistributionFunction (Histogram *hist*)

```
                                                          {
           this.min = hist.getMin();
           this.incr = hist.getIncr();
           prob = new float[hist.getHist().length];
           int tot = 0;
           for (int i=0; i < hist.getHist().length; ++i){
               tot += hist.getHist()[i];
           }
           if ( tot == 0 ) return;

           int sofar = 0;
           for (int i=0; i < hist.getHist().length; ++i){
               sofar += hist.getHist()[i];
```

```
            prob[i] = sofar / (float) tot;
        }
    }
```

6.13.3 Member Function Documentation

6.13.3.1 int edu.ou.asgbook.histogram.CumulativeDistributionFunction.getIncr ()

```
                         {
        return incr;
    }
```

6.13.3.2 int edu.ou.asgbook.histogram.CumulativeDistributionFunction.getMin ()

```
                        {
        return min;
    }
```

6.13.3.3 float [ ] edu.ou.asgbook.histogram.CumulativeDistribution-Function.getProb ()

```
                             {
        return prob;
    }
```

6.13.3.4 static void edu.ou.asgbook.histogram.Cumulative-DistributionFunction.main (String[ ] *args*) throws Exception [static]

```
                                                             {
        // create output directory
        File outdir = OutputDirectory.getDefault("cdf");

        // read input
        LatLonGrid conus = SurfaceAlbedo.read(SurfaceAlbedo.CONUS, 100);

        // find histogram
        final int MIN = 0;
        final int MAX = 30;
        for (int incr=1; incr < 10; incr += 2){
            Histogram hist = new Histogram(MIN, incr, (MAX-MIN)/incr );
            hist.update(conus);
            CumulativeDistributionFunction cdf = new CumulativeDistributionFunction(hist);
            System.out.println("INCR=" + incr + " nbins=" + cdf.prob.length);
            System.out.println(cdf);
            String filename = outdir.getAbsolutePath() + "/cdf_" + incr + ".txt";
            PrintWriter writer = new PrintWriter(new FileWriter(filename));
            writer.println(cdf);
            writer.close();
            System.out.println("Wrote to " + filename);
        }
    }
```

6.13.3.5 Override String edu.ou.asgbook.histogram.CumulativeDistribution-Function.toString ()

```
                            {
        StringBuilder sb = new StringBuilder();
        for (int i=0; i < prob.length; ++i){
            int sval = min + i * incr;
            int eval = sval + incr;
            sb.append(sval);
            sb.append(" ");
            sb.append(eval);
            sb.append(" ");
            sb.append(prob[i]);
            sb.append("\n");
        }
        return sb.toString();
    }
```

6.14 edu.ou.asgbook.dataset.DailyRainfall Class Reference

Reads the ASCII precipitation data available at http://madis-data.noaa.gov/public/hydrodumpguest.html.

Static Public Member Functions

- static PointObservations read (File file) throws IOException
- static void main (String[ ] args) throws Exception

Static Public Attributes

- static final File TN_Oct2010 = new File("data/madishydro/tn_oct2010_-24hr.txt")

Package Functions

- SuppressWarnings ("unused") public static PointObservations read(Reader r) throws IOException

6.14.1 Detailed Description

Reads the ASCII precipitation data available at http://madis-data.noaa.gov/public/hydrodumpguest.html.

Author:

Valliappa.Lakshmanan

6.14.2 Member Function Documentation

6.14.2.1 static void edu.ou.asgbook.dataset.DailyRainfall.main (String[ ] *args*) throws Exception [static]

```
                                                        {
        PointObservations data = DailyRainfall.read(DailyRainfall.TN_Oct2010);
        for (int i=0; i < data.getPoints().length; ++i){
            System.out.println(data.getPoints()[i]);
        }
    }
```

6.14.2.2 static PointObservations edu.ou.asgbook.dataset.DailyRainfall.read (File *file*) throws IOException `[static]`

```
                                                                      {
        Reader f = null;
        if (file.getAbsolutePath().endsWith(".gz")) {
            f = new InputStreamReader(new GZIPInputStream(new FileInputStream(
                    file)));
        } else {
            f = new FileReader(file);
        }
        return read(f);
    }
```

6.14.2.3 edu.ou.asgbook.dataset.DailyRainfall.SuppressWarnings ("unused") throws IOException `[package]`

```
                                                                     {
        Scanner s = new Scanner(r);
        List<PointObservations.ObservationPoint> result = new ArrayList<PointObservations.ObservationPo

        final int FACTOR = 1000;
        final int MISSING = -9999 * FACTOR;
        s.nextLine(); // header
        while (s.hasNext()){
            String station = s.next();
            String date = s.next();
            String time = s.next();
            int precip = (int) Math.round( s.nextDouble() * FACTOR );
            double lat = s.nextDouble();
            double lon = s.nextDouble();
            result.add(new PointObservations.ObservationPoint(lat,lon,precip));
        }

        PointObservations.ObservationPoint[] pts = result.toArray(new PointObservations.ObservationPoir
        return new PointObservations(pts, MISSING);
    }
```

6.14.3 Member Data Documentation

6.14.3.1 final File edu.ou.asgbook.dataset.DailyRainfall.TN_Oct2010 = new File("data/madishydro/tn_oct2010_24hr.txt") `[static]`

6.15 edu.ou.asgbook.rbf.DataSimulator Class Reference

Simulates RBF data to be fit.

Static Public Member Functions

- static LatLonGrid simulateData (Pixel[ ] centers, double[ ] sigmax, double[ ] sigmay, int nrows, int ncols)
- static void simulateData (LatLonGrid result, Pixel[ ] centers, double[ ] sigmax, double[ ] sigmay)
- static void main (String[ ] args) throws Exception

6.15.1 Detailed Description

Simulates RBF data to be fit.

Author:

v.lakshmanan

6.15.2 Member Function Documentation

6.15.2.1 static void edu.ou.asgbook.rbf.DataSimulator.main (String[ ] *args*) throws Exception [static]

```
                                                          {
        int nrows = 100;
        int ncols = 100;
        Pixel[] centers = new Pixel[]{ new Pixel(nrows/4,ncols/3,20), new Pixel(nrows/3,nco
        double[] sigmax = new double[] { nrows/12, ncols/8 };
        double[] sigmay = new double[] { nrows/8, ncols/12 };
        LatLonGrid m = DataSimulator.simulateData(centers, sigmax, sigmay, nrows, ncols);

        // write out as image, for viewing
        File out = OutputDirectory.getDefault("rbf");
        KmlWriter.write(m, out, "simulated", PngWriter.createCoolToWarmColormap());

        System.out.println("Done");
    }
```

6.15.2.2 static void edu.ou.asgbook.rbf.DataSimulator.simulateData (LatLonGrid *result*, Pixel[ ] *centers*, double[ ] *sigmax*, double[ ] *sigmay*) [static]

```

        for (int i=0; i < result.getNumLat(); ++i) for (int j=0; j < result.getNumLon(); ++j){
            double tot = 0;
            for (int k=0; k < centers.length; ++k){
                double xdist = i - centers[k].getX();
                double ydist = j - centers[k].getY();
                double xnorm = (xdist*xdist) / (sigmax[k] * sigmax[k]);
                double ynorm = (ydist*ydist) / (sigmay[k] * sigmay[k]);
                double wt = Math.exp(-(xnorm + ynorm));
                tot += wt * centers[k].getValue();
            }
            if ( tot > 0 ){
                result.setValue(i, j, (int) Math.round(tot));
            } else {
                result.setValue(i, j, 0);
            }
        }
    }
```

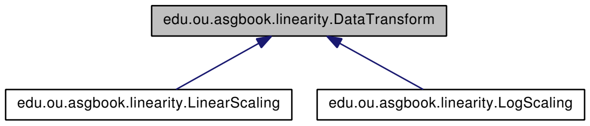
6.15.2.3 static LatLonGrid edu.ou.asgbook.rbf.DataSimulator.simulateData (Pixel[ ] *centers*, double[ ] *sigmax*, double[ ] *sigmay*, int *nrows*, int *ncols*) [static]

```

        LatLon nwCorner = new LatLon(38, -100);
        double latres = 0.01;
        double lonres = 0.01;
        LatLonGrid result = new LatLonGrid(nrows, ncols, -999, nwCorner, latres, lonres );
        simulateData(result, centers, sigmax, sigmay);
        return result;
    }
```

6.16 edu.ou.asgbook.linearity.DataTransform Class Reference

Transform pixel values, usually to meet linearity requirements.

Inheritance diagram for edu.ou.asgbook.linearity.DataTransform:


edu.ou.asgbook.linearity.DataTransform
edu.ou.asgbook.linearity.LinearScaling
edu.ou.asgbook.linearity.LogScaling

Public Member Functions

- int transformAndRoundoff (double value)
- abstract double transform (double value)
- abstract double inverse (double value)

6.16.1 Detailed Description

Transform pixel values, usually to meet linearity requirements.

Author:

valliappa.lakshmanan

6.16.2 Member Function Documentation

6.16.2.1 abstract double edu.ou.asgbook.linearity.DataTransform.inverse (double *value*) `[pure virtual]`

Implemented in edu.ou.asgbook.linearity.LinearScaling, and edu.ou.asgbook.linearity.LogScaling.

6.16.2.2 abstract double edu.ou.asgbook.linearity.DataTransform.transform (double *value*) `[pure virtual]`

Implemented in edu.ou.asgbook.linearity.LinearScaling, and edu.ou.asgbook.linearity.LogScaling.

6.16.2.3 int edu.ou.asgbook.linearity.DataTransform.transformAndRoundoff (double *value*)

```
                                                    {
            return (int) Math.round(transform(value));
        }
```

6.17 edu.ou.asgbook.motion.Differencer Class Reference

Just computes a pixel-by-pixel difference.

Public Member Functions

- LatLonGrid compute (LatLonGrid data0, LatLonGrid data1)

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.17.1 Detailed Description

Just computes a pixel-by-pixel difference.

This is really not a motion estimation, but is there just to show what happens when you do so.

Author:

v.lakshmanan

6.17.2 Member Function Documentation

6.17.2.1 LatLonGrid edu.ou.asgbook.motion.Differencer.compute (LatLonGrid *data0*, LatLonGrid *data1*)

```
                                                                    {
        LatLonGrid result = LatLonGrid.copyOf(data1);
        for (int i=0; i < result.getNumLat(); ++i){
            for (int j=0; j < result.getNumLon(); ++j){
                int diff = data1.getValue(i,j) - data0.getValue(i,j);
                result.setValue(i,j, diff);
            }
        }
        return result;
    }
```

6.17.2.2 static void edu.ou.asgbook.motion.Differencer.main (String[ ] *args*) throws Exception [static]

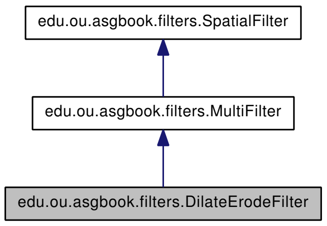
```
                                                            {
```

```
        File out = OutputDirectory.getDefault("difference");

        // seviri
        File f = new File("data/seviri");
        Pair<LatLonGrid,Date>[] grids = SeviriInfraredTemperature.readAll(f);
        KmlWriter.write(grids[0].first, out, "ir0", PngWriter.createCoolToWarmColormap());
        KmlWriter.write(grids[1].first, out, "ir1", PngWriter.createCoolToWarmColormap());
        LatLonGrid diff = new Differencer().compute(grids[0].first, grids[1].first);
        KmlWriter.write(diff, out, "irdiff", PngWriter.createCoolToWarmColormap());

        // popdensity
        DataTransform[] transforms = {new GlobalPopulation.LinearScaling(), new GlobalPopulation.LogSca
        String[] prefix = {"popdensity", "logpopdensity"};
        for (int i=0; i < transforms.length; ++i){
            LatLonGrid popdensity0 = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA1990, transform
            LatLonGrid popdensity1 = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, transforms[i]
            KmlWriter.write(popdensity0, out, prefix[i]+"0", PngWriter.createCoolToWarmColormap());
            KmlWriter.write(popdensity0, out, prefix[i]+"1", PngWriter.createCoolToWarmColormap());
            diff = new Differencer().compute(popdensity0, popdensity1);
            KmlWriter.write(diff, out, prefix[i]+"diff", PngWriter.createCoolToWarmColormap());
        }
    }
```

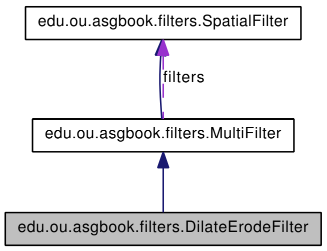
6.18 edu.ou.asgbook.filters.DilateErodeFilter Class Reference

Carries out paired dilation followed by erosion for filling in holes.

Inheritance diagram for edu.ou.asgbook.filters.DilateErodeFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.MultiFilter
edu.ou.asgbook.filters.DilateErodeFilter

Collaboration diagram for edu.ou.asgbook.filters.DilateErodeFilter:


edu.ou.asgbook.filters.SpatialFilter
filters
edu.ou.asgbook.filters.MultiFilter
edu.ou.asgbook.filters.DilateErodeFilter

Public Member Functions

- DilateErodeFilter (int halfSize, int numTimes)

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.18.1 Detailed Description

Carries out paired dilation followed by erosion for filling in holes.

Author:

Valliappa.Lakshmanan

6.18.2 Constructor & Destructor Documentation

6.18.2.1 edu.ou.asgbook.filters.DilateErodeFilter.DilateErodeFilter (int *halfSize*, int *numTimes*)

```
                                                          {
        super(new SpatialFilter[]{
            new DilationFilter(halfSize), new ErosionFilter(halfSize)
        }, numTimes);
    }
```

6.18.3 Member Function Documentation

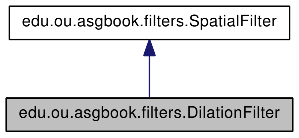
6.18.3.1 static void edu.ou.asgbook.filters.DilateErodeFilter.main (String[ ] *args*) throws Exception [static]

```
                                                             {
        // create output directory
        File out = OutputDirectory.getDefault("dilateerode");

        // read input
        LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new GlobalPopulati
        KmlWriter.write(popdensity, out, "orig", PngWriter.createCoolToWarmColormap());

        // erode
        LatLonGrid erode1 = new DilateErodeFilter(1,1).filter(popdensity);
        KmlWriter.write(erode1, out, "dilateerode_3_1", PngWriter.createCoolToWarmColormap());
        LatLonGrid erode3 = new DilateErodeFilter(1,3).filter(popdensity);
        KmlWriter.write(erode3, out, "dilateerode_3_3", PngWriter.createCoolToWarmColormap());
        LatLonGrid erode5 = new DilateErodeFilter(2,3).filter(popdensity);
        KmlWriter.write(erode5, out, "dilateerode_5_3", PngWriter.createCoolToWarmColormap());
    }
```

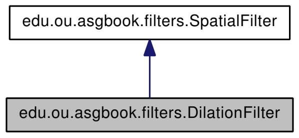
6.19 edu.ou.asgbook.filters.DilationFilter Class Reference

Expands entities by taking a local maximum.

Inheritance diagram for edu.ou.asgbook.filters.DilationFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.DilationFilter

Collaboration diagram for edu.ou.asgbook.filters.DilationFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.DilationFilter

Public Member Functions

- DilationFilter (int halfSize)
- Override LatLonGrid filter (LatLonGrid input)
- LatLonGrid dilate (final LatLonGrid input)

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.19.1 Detailed Description

Expands entities by taking a local maximum.

Author:

Valliappa.Lakshmanan

6.19.2 Constructor & Destructor Documentation

6.19.2.1 edu.ou.asgbook.filters.DilationFilter.DilationFilter (int *halfSize*)

```
                                    {
        this.halfSize = halfSize;
    }
```

6.19.3 Member Function Documentation

6.19.3.1 LatLonGrid edu.ou.asgbook.filters.DilationFilter.dilate (final LatLonGrid *input*)

```
                                                {
        LatLonGrid output = LatLonGrid.copyOf(input);
        output.fill(output.getMissing());
        int[][] outData = output.getData();
        int[][] inData = input.getData();
        int hx = halfSize;
        int hy = halfSize;
        int nx = inData.length;
        int ny = inData[0].length;
        for (int i=hx; i < (nx-hx); ++i){
            for (int j=hy; j < (ny-hy); ++j){
                int max = input.getMissing();
                boolean set = false;
                for (int m=-hx; m <= hx; ++m){
                    for (int n=-hy; n <= hy; ++n){
                        int inval = inData[i+m][j+n];
                        if (inval != input.getMissing()){
                            if ( !set || inval > max ){
                                max = inval;
                                set = true;
                            }
                        }
                    }
                }
                if ( set ){
                    outData[i][j] = max;
                }
            }
        }
        return output;
    }
```

6.19.3.2 Override LatLonGrid edu.ou.asgbook.filters.DilationFilter.filter (LatLonGrid *input*)

```
                                                {
        return dilate(input);
    }
```

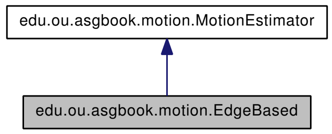
6.19.3.3 static void edu.ou.asgbook.filters.DilationFilter.main (String[ ] *args*) throws Exception [static]

```
                                                          {
       // create output directory
       File out = OutputDirectory.getDefault("dilate");

       // read input
       LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new Glo
       KmlWriter.write(popdensity, out, "orig", PngWriter.createCoolToWarmColormap());

       // dilate
       LatLonGrid dilate1 = new DilationFilter(1).dilate(popdensity);
       KmlWriter.write(dilate1, out, "dilate_3", PngWriter.createCoolToWarmColormap());
       LatLonGrid dilate3 = new DilationFilter(3).dilate(popdensity);
       KmlWriter.write(dilate3, out, "dilate_7", PngWriter.createCoolToWarmColormap());
       LatLonGrid dilate5 = new DilationFilter(5).dilate(popdensity);
       KmlWriter.write(dilate5, out, "dilate_11", PngWriter.createCoolToWarmColormap());
   }
```

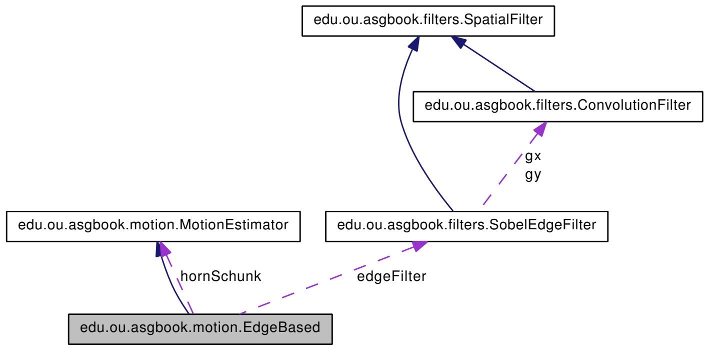
6.20 edu.ou.asgbook.motion.EdgeBased Class Reference

Estimates motion based on the displacement of edges.

Inheritance diagram for edu.ou.asgbook.motion.EdgeBased:


edu.ou.asgbook.motion.MotionEstimator
edu.ou.asgbook.motion.EdgeBased

Collaboration diagram for edu.ou.asgbook.motion.EdgeBased:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.ConvolutionFilter
gx
gy
edu.ou.asgbook.motion.MotionEstimator
edu.ou.asgbook.filters.SobelEdgeFilter
hornSchunk
edgeFilter
edu.ou.asgbook.motion.EdgeBased

Public Member Functions

- Override Pair< LatLonGrid, LatLonGrid > compute (LatLonGrid data0, LatLonGrid data1, File outdir)

    returns motion in the two directions.

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.20.1 Detailed Description

Estimates motion based on the displacement of edges.

Author:

v.lakshmanan

6.20.2 Member Function Documentation

6.20.2.1 Override Pair<LatLonGrid, LatLonGrid> edu.ou.asgbook.motion.EdgeBased.compute (LatLonGrid *data0*, LatLonGrid *data1*, File *outdir*)

returns motion in the two directions.

The first one is north to south and the second one is east to west. The data is aligned to second time frame. The output dir is used for intermediate products and may be null.

Implements edu.ou.asgbook.motion.MotionEstimator.

```

        // do an edge filter on the pair of images
        LatLonGrid edge0 = edgeFilter.edgeFilter(data0);
        LatLonGrid edge1 = edgeFilter.edgeFilter(data1);

        if (outdir != null){
            try {
                KmlWriter.write(edge0, outdir, "edge0", PngWriter.createCoolToWarmColormap()
                KmlWriter.write(edge1, outdir, "edge1", PngWriter.createCoolToWarmColormap()
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

        return hornSchunk.compute(edge0, edge1, outdir);
    }
```

6.20.2.2 static void edu.ou.asgbook.motion.EdgeBased.main (String[ ] *args*) throws Exception [static]

```
                                                                {
        // create output directory
        File out = OutputDirectory.getDefault("edgemotion");

        // read
        File f = new File("data/seviri");
        Pair<LatLonGrid,Date>[] grids = SeviriInfraredTemperature.readAll(f);

        // do alg
        MotionEstimator alg = new EdgeBased();
        Pair<LatLonGrid,LatLonGrid> motion = alg.compute(grids[0].first, grids[1].first, out

        // write
        SaturateFilter filter = new SaturateFilter(-150, 150);
        LatLonGrid u = filter.filter(motion.first);
        LatLonGrid v = filter.filter(motion.second);
        KmlWriter.write(u, out, "opticflow_u", PngWriter.createCoolToWarmColormap());
        KmlWriter.write(v, out, "opticflow_v", PngWriter.createCoolToWarmColormap());
    }
```

6.21 edu.ou.asgbook.projections.Ellipsoid Class Reference

An ellipsoidal approximation to the earth.

Public Member Functions

- Ellipsoid (double eqr, double eccsq)

Static Public Member Functions

- static Ellipsoid WGS84 ()
- static Ellipsoid NAD27 ()

Public Attributes

- final double eqr

  Equatorial radius (the semi-major axis) in meters.

- final double eccsq

  Square of the eccentricity.

6.21.1 Detailed Description

An ellipsoidal approximation to the earth.

Author:

v.lakshmanan

6.21.2 Constructor & Destructor Documentation

6.21.2.1 edu.ou.asgbook.projections.Ellipsoid.Ellipsoid (double *eqr*, double *eccsq*)

```
                                              {
        super();
        this.eqr = eqr;
        this.eccsq = eccsq;
    }
```

6.21.3 Member Function Documentation

6.21.3.1 static Ellipsoid edu.ou.asgbook.projections.Ellipsoid.NAD27 () [static]

```
                               {
        return new Ellipsoid(6378206, 0.006768658);
    }
```

6.21.3.2 static Ellipsoid edu.ou.asgbook.projections.Ellipsoid.WGS84 () [static]

```
                               {
        return new Ellipsoid(6378137, 0.00669438);
    }
```

6.21.4 Member Data Documentation

6.21.4.1 final double edu.ou.asgbook.projections.Ellipsoid.eccsq

Square of the eccentricity.

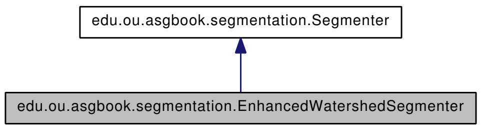
6.21.4.2 final double edu.ou.asgbook.projections.Ellipsoid.eqr

Equatorial radius (the semi-major axis) in meters.

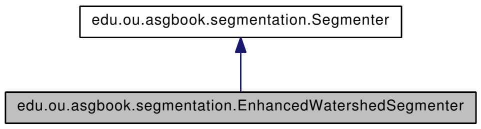
6.22 edu.ou.asgbook.segmentation.EnhancedWatershedSegmenter Class Reference

Enhanced watershed segmentation following Lakshmanan, Hondl and Rabin.

Inheritance diagram for edu.ou.asgbook.segmentation.EnhancedWatershedSegmenter:


edu.ou.asgbook.segmentation.Segmenter
edu.ou.asgbook.segmentation.EnhancedWatershedSegmenter

Collaboration diagram for edu.ou.asgbook.segmentation.EnhancedWatershed-Segmenter:


edu.ou.asgbook.segmentation.Segmenter
edu.ou.asgbook.segmentation.EnhancedWatershedSegmenter

Public Member Functions

- EnhancedWatershedSegmenter (int minThresh, int dataIncr, int maxThresh, int sizeThresholdInPixels, int deltaForCluster)
- LabelResult label (LatLonGrid dataval)

  Creates a labeled grid where background pixels are set to 0 and labels for objects go 1,2,3.

Static Public Member Functions

- static void main (String[ ] args) throws Exception

Package Functions

- SuppressWarnings ("serial") private static class Pixels extends ArrayList< Pixel >

Classes

- class **Glob**

6.22.1 Detailed Description

Enhanced watershed segmentation following Lakshmanan, Hondl and Rabin.

Author:

valliappa.lakshmanan

6.22.2 Constructor & Destructor Documentation

6.22.2.1 edu.ou.asgbook.segmentation.EnhancedWatershedSegmenter.EnhancedWatershedSegmenter (int *minThresh*, int *dataIncr*, int *maxThresh*, int *sizeThresholdInPixels*, int *deltaForCluster*)

Parameters:

minThresh,: minimum pixel value for a pixel to be part of a region

dataIncr,: quantization interval. Use 1 if you don't want to quantize

maxThresh,: values > maxThresh are treated as maxThresh

sizeThresholdInPixels,: Blobs smaller than the specified size will get ignored.

deltaForCluster,: Specify how many data-increments a cluster is allowed to range over. For example, if you specify 0, then a cluster will contain only values that fall in the same interval as the maximum. Larger values of D also yield clusters at larger scales.

```

        this.myDelta = deltaForCluster;
        this.myMinSize = sizeThresholdInPixels;
        this.minThresh = minThresh;
        this.maxThresh = maxThresh;
        this.dataIncr = dataIncr;
    }
```

6.22.3 Member Function Documentation

6.22.3.1 LabelResult edu.ou.asgbook.segmentation.EnhancedWatershedSegmenter.label (LatLonGrid *dataval*) [virtual]

Creates a labeled grid where background pixels are set to 0 and labels for objects go 1,2,3.

.. All pixels > thresh are part of an object.

Implements edu.ou.asgbook.segmentation.Segmenter.

```
                                            {
        LatLonGrid marked = findLocalMaxima(dataval);
        LabelResult initial = new ThresholdSegmenter(0).label(marked);
        LabelResult pruned = RegionProperty.pruneBySize(initial, dataval, myMinSize);
        return pruned;
    }
```

6.22.3.2 static void edu.ou.asgbook.segmentation.Enhanced-WatershedSegmenter.main (String[ ] *args*) throws Exception [static]

```
                                                        {
        File out = OutputDirectory.getDefault("ewshed");

        // data
        LatLonGrid grid = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new GlobalPo
        KmlWriter.write(grid, out, "orig", PngWriter.createCoolToWarmColormap());

        // int MIN = 200; int MAX = 500; int INCR = 10;// log scaling
        int MIN = 1; int MAX = 100; int INCR = 1;// linear scaling

        for (int sizethresh=5; sizethresh <= 20; sizethresh += 5){
            EnhancedWatershedSegmenter seg = new EnhancedWatershedSegmenter(MIN, INCR, MAX,
            LatLonGrid label = seg.label(grid).label;
            KmlWriter.write(label, out, "ewsheds_"+sizethresh, PngWriter.createRandomColorm
        }

        grid = new ConvolutionFilter(ConvolutionFilter.gauss(9, 9)).smooth(grid);
        for (int sizethresh=5; sizethresh <= 20; sizethresh += 5){
            EnhancedWatershedSegmenter seg = new EnhancedWatershedSegmenter(MIN, INCR, MAX,
            LatLonGrid label = seg.label(grid).label;
            KmlWriter.write(label, out, "smewsheds_"+sizethresh, PngWriter.createRandomColo
        }
    }
```

6.22.3.3 edu.ou.asgbook.segmentation.EnhancedWatershed-Segmenter.SuppressWarnings ("serial") [package]

```
                                                    {
    }
```

6.23 edu.ou.asgbook.histogram.Entropy Class Reference

Compute entropy from a histogram.

Static Public Member Functions

- static double computeEntropy (Histogram hist)
- static void main (String[ ] args) throws Exception

6.23.1 Detailed Description

Compute entropy from a histogram.

Author:

v.lakshmanan

6.23.2 Member Function Documentation

6.23.2.1 static double edu.ou.asgbook.histogram.Entropy.computeEntropy (Histogram *hist*) [static]

```
                                                   {
        float[] prob = hist.calcProb();
        double entropy = 0;
        for (int i=0; i < prob.length; ++i){
            if ( prob[i] > 0 ){
                double plogp = prob[i] * Math.log(prob[i]);
                entropy -= plogp;
            }
        }
        // to base 2
        return (entropy / Math.log(2.0));
    }
```

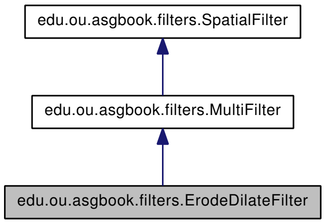
6.23.2.2 static void edu.ou.asgbook.histogram.Entropy.main (String[ ] *args*) throws Exception [static]

```
                                                        {
        LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA);
        Histogram hist = new Histogram(0, 100, 100 );
        hist.update(popdensity);
        double e1 = Entropy.computeEntropy(hist);
        System.out.println("Population density entropy = " + e1);

        LatLonGrid albedo = SurfaceAlbedo.read(SurfaceAlbedo.CONUS, 100);
        hist = new Histogram(0, 30, 30 );
        hist.update(albedo);
        double e2 = Entropy.computeEntropy(hist);
        System.out.println("surface albedo entropy = " + e2);
    }
```

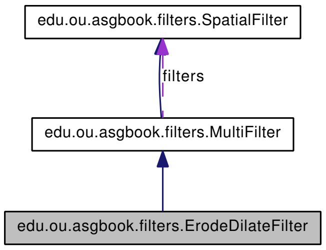
6.24 edu.ou.asgbook.filters.ErodeDilateFilter Class Reference

Carries out paired erosion followed by dilation for denoising.

Inheritance diagram for edu.ou.asgbook.filters.ErodeDilateFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.MultiFilter
edu.ou.asgbook.filters.ErodeDilateFilter

Collaboration diagram for edu.ou.asgbook.filters.ErodeDilateFilter:


edu.ou.asgbook.filters.SpatialFilter
filters
edu.ou.asgbook.filters.MultiFilter
edu.ou.asgbook.filters.ErodeDilateFilter

Public Member Functions

- ErodeDilateFilter (int halfSize, int numTimes)

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.24.1 Detailed Description

Carries out paired erosion followed by dilation for denoising.

Author:

Valliappa.Lakshmanan

6.24.2 Constructor & Destructor Documentation

6.24.2.1 edu.ou.asgbook.filters.ErodeDilateFilter.ErodeDilateFilter (int *halfSize*, int *numTimes*)

```
                                                   {
        super(new SpatialFilter[]{
            new ErosionFilter(halfSize), new DilationFilter(halfSize)
        }, numTimes);
    }
```

6.24.3 Member Function Documentation

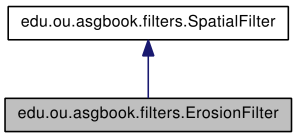
6.24.3.1 static void edu.ou.asgbook.filters.ErodeDilateFilter.main (String[ ] *args*) throws Exception [static]

```
                                                        {
        // create output directory
        File out = OutputDirectory.getDefault("dilateerode");

        // read input
        LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new Gl
        KmlWriter.write(popdensity, out, "orig", PngWriter.createCoolToWarmColormap());

        // erode
        LatLonGrid erode1 = new ErodeDilateFilter(1,1).filter(popdensity);
        KmlWriter.write(erode1, out, "erodedilate_3_1", PngWriter.createCoolToWarmColormap(
        LatLonGrid erode3 = new ErodeDilateFilter(1,3).filter(popdensity);
        KmlWriter.write(erode3, out, "erodedilate_3_3", PngWriter.createCoolToWarmColormap(
        LatLonGrid erode5 = new ErodeDilateFilter(2,3).filter(popdensity);
        KmlWriter.write(erode5, out, "erodedilate_5_3", PngWriter.createCoolToWarmColormap(
    }
```

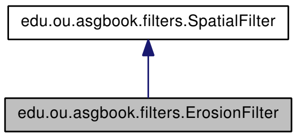
6.25 edu.ou.asgbook.filters.ErosionFilter Class Reference

Reduces the size of entities by taking a local mininum.

Inheritance diagram for edu.ou.asgbook.filters.ErosionFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.ErosionFilter

Collaboration diagram for edu.ou.asgbook.filters.ErosionFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.ErosionFilter

Public Member Functions

- ErosionFilter (int halfSize)
- Override LatLonGrid filter (LatLonGrid input)
- LatLonGrid erode (final LatLonGrid input)

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.25.1 Detailed Description

Reduces the size of entities by taking a local mininum.

Author:

Valliappa.Lakshmanan

6.25.2 Constructor & Destructor Documentation

6.25.2.1 edu.ou.asgbook.filters.ErosionFilter.ErosionFilter (int *halfSize*)

```
                                    {
        this.halfSize = halfSize;
    }
```

6.25.3 Member Function Documentation

6.25.3.1 LatLonGrid edu.ou.asgbook.filters.ErosionFilter.erode (final LatLonGrid *input*)

```
                                                    {
        LatLonGrid output = LatLonGrid.copyOf(input);
        output.fill(output.getMissing());
        int[][] outData = output.getData();
        int[][] inData = input.getData();
        int hx = halfSize;
        int hy = halfSize;
        int nx = inData.length;
        int ny = inData[0].length;
        for (int i=hx; i < (nx-hx); ++i){
            for (int j=hy; j < (ny-hy); ++j){
                int min = input.getMissing();
                boolean set = false;
                for (int m=-hx; m <= hx; ++m){
                    for (int n=-hy; n <= hy; ++n){
                        int inval = inData[i+m][j+n];
                        if (inval != input.getMissing()){
                            if ( !set || inval < min ){
                                min = inval;
                                set = true;
                            }
                        }
                    }
                }
                if ( set ){
                    outData[i][j] = min;
                }
            }
        }
        return output;
    }
```

6.25.3.2 Override LatLonGrid edu.ou.asgbook.filters.ErosionFilter.filter (LatLonGrid *input*)

```
                                                {
        return erode(input);
    }
```

6.25.3.3 static void edu.ou.asgbook.filters.ErosionFilter.main (String[ ] *args*) throws Exception [static]

```
                                                        {
        // create output directory
        File out = OutputDirectory.getDefault("erode");

        // read input
        LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new GlobalPopulati
        KmlWriter.write(popdensity, out, "orig", PngWriter.createCoolToWarmColormap());

        // erode
        LatLonGrid erode1 = new ErosionFilter(1).erode(popdensity);
        KmlWriter.write(erode1, out, "erode_3", PngWriter.createCoolToWarmColormap());
        LatLonGrid erode3 = new ErosionFilter(3).erode(popdensity);
        KmlWriter.write(erode3, out, "erode_7", PngWriter.createCoolToWarmColormap());
        LatLonGrid erode5 = new ErosionFilter(5).erode(popdensity);
        KmlWriter.write(erode5, out, "erode_11", PngWriter.createCoolToWarmColormap());
    }
```

6.26 edu.ou.asgbook.io.EsriGrid Class Reference

Read an ESRI grid.

Static Public Member Functions

- static LatLonGrid read (File file, DataTransform t) throws IOException, FileNotFoundException
- static LatLonGrid read (Reader inputFile, DataTransform t)
- static void write (LatLonGrid data, File outdir, String fname) throws IOException
- static void write (LatLonGrid data, File out) throws IOException

6.26.1 Detailed Description

Read an ESRI grid.

Author:

valliappa.lakshmanan

6.26.2 Member Function Documentation

6.26.2.1 static LatLonGrid edu.ou.asgbook.io.EsriGrid.read (Reader *inputFile*, DataTransform *t*) [static]

```
                                                                    {
           BufferedReader reader = null;
           try {
               reader = new BufferedReader(inputFile);
               // fields separated by spaces. This is the regular expression for spaces
               final String sep = " +";
               // read header
               int ncols = Integer.parseInt( reader.readLine().split(sep)[1] );
               int nrows = Integer.parseInt( reader.readLine().split(sep)[1] );
               double cornerlon = Double.parseDouble( reader.readLine().split(sep)[1] );
               double cornerlat = Double.parseDouble( reader.readLine().split(sep)[1] );
               double latres = Double.parseDouble( reader.readLine().split(sep)[1] );
               double lonres = latres;
               String missingValue = reader.readLine().split(sep)[1];
               int missing = Integer.parseInt(missingValue);

               // read in data
               int[][] data = new int[nrows][ncols];
               int numvalid = 0;
               int nummissing = 0;
               int numzero = 0;
```

```
            int minval = Integer.MAX_VALUE;
            int maxval = 0;
            int i = 0;
            int j = 0;
            String line = null;
            while ( (line = reader.readLine()) != null ){
                for (String field : line.split(sep)){
                    if (field.equals(missingValue)){
                        data[i][j] = missing;
                        ++nummissing;
                    } else {
                        double value = Double.parseDouble(field);
                        data[i][j] = t.transformAndRoundoff(value);
                        if ( data[i][j] != 0 ){
                            ++numvalid;
                            minval = Math.min(minval, data[i][j]);
                            maxval = Math.max(maxval, data[i][j]);
                        } else {
                            ++numzero;
                        }
                    }
                    ++j; // next column
                    if ( j == ncols ){
                        j = 0; // next row
                        ++i;
                    }
                }
            }
            System.out.println(numvalid + " valid pixels; " + numzero + " zero; " + nummissing + " " +
            LatLon nwCorner = new LatLon(cornerlat + latres*nrows, cornerlon);
            return new LatLonGrid(data, missing, nwCorner, latres, lonres);
        } catch (Exception e){
            System.err.println("Error reading file: " + e);
            throw new IllegalArgumentException(e);
        } finally {
            if (reader != null) {
                try{
                    reader.close();
                } catch (Exception e){
                    // okay
                }
            }
        }
    }
```

6.26.2.2 static LatLonGrid edu.ou.asgbook.io.EsriGrid.read (File *file*, DataTransform *t*) throws IOException, FileNotFoundException [static]

```

        Reader f = null;
        if (file.getAbsolutePath().endsWith(".gz")) {
            f = new InputStreamReader(new GZIPInputStream(new FileInputStream(
                    file)));
        } else {
            f = new FileReader(file);
        }
        return read(f, t);
    }
```

6.26.2.3 static void edu.ou.asgbook.io.EsriGrid.write ([LatLonGrid](#) *data*, File *out*) throws IOException [static]

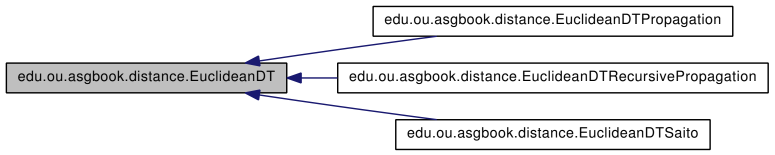
```
                                                                                    {
        PrintWriter writer = null;
        try {
            writer = new PrintWriter(new GZIPOutputStream(new FileOutputStream(out)));
            int nrows = data.getNumLat();
            int ncols = data.getNumLon();
            writer.println("ncols " + ncols);
            writer.println("nrows " + nrows);
            writer.println("xllcorner " + data.getNwCorner().getLon());
            writer.println("yllcorner " + (data.getNwCorner().getLat() - data.getLatRes()*
            writer.println("cellsize " + data.getLatRes());
            writer.println("NODATA_value " + data.getMissing());

            final String sep = " ";
            for (int i=0; i < nrows; ++i){
                for (int j=0; j < ncols; ++j){
                    writer.print(data.getValue(i,j));
                    if (j != (ncols-1)){
                        writer.print(sep);
                    }
                }
                writer.println();
            }
        } finally {
            if (writer != null){
                System.out.println("Successfully wrote " + out);
                writer.close();
            }
        }
    }
```

6.26.2.4 static void edu.ou.asgbook.io.EsriGrid.write ([LatLonGrid](#) *data*, File *outdir*, String *fname*) throws IOException [static]

```

        File f = new File(outdir.getAbsolutePath() + "/" + fname);
        write(data, f);
    }
```

6.27 edu.ou.asgbook.distance.EuclideanDT Interface Reference

Inheritance diagram for edu.ou.asgbook.distance.EuclideanDT:


edu.ou.asgbook.distance.EuclideanDTPropagation
edu.ou.asgbook.distance.EuclideanDT
edu.ou.asgbook.distance.EuclideanDTRecursivePropagation
edu.ou.asgbook.distance.EuclideanDTSaito

Public Member Functions

- abstract LatLonGrid getDistanceTransform (LatLonGrid data, int thresh)

  At every pixel, finds the square of the Euclidean distance to the nearest pixel > thresh.

6.27.1 Member Function Documentation

6.27.1.1 abstract LatLonGrid edu.ou.asgbook.distance.EuclideanDT.getDistanceTransform (LatLonGrid *data*, int *thresh*) `[pure virtual]`

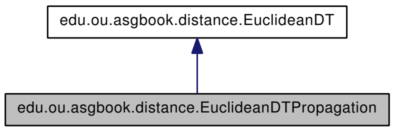
At every pixel, finds the square of the Euclidean distance to the nearest pixel > thresh.

Implemented in edu.ou.asgbook.distance.EuclideanDTPropagation, edu.ou.asgbook.distance.EuclideanDTRecursivePropagation, and edu.ou.asgbook.distance.EuclideanDTSaito.

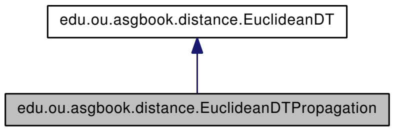
6.28 edu.ou.asgbook.distance.EuclideanDTPropagation Class Reference

Implementation of Euclidean distance that updates the distance instead of computing it afresh each time.

Inheritance diagram for edu.ou.asgbook.distance.EuclideanDTPropagation:


edu.ou.asgbook.distance.EuclideanDT
edu.ou.asgbook.distance.EuclideanDTPropagation

Collaboration diagram for edu.ou.asgbook.distance.EuclideanDTPropagation:


edu.ou.asgbook.distance.EuclideanDT
edu.ou.asgbook.distance.EuclideanDTPropagation

Public Member Functions

- Override LatLonGrid getDistanceTransform (LatLonGrid data, int thresh)

  At every pixel, finds the square of the Euclidean distance to the nearest pixel > thresh.

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.28.1 Detailed Description

Implementation of Euclidean distance that updates the distance instead of computing it afresh each time.

Author:

v.lakshmanan

6.28.2 Member Function Documentation

6.28.2.1 Override LatLonGrid edu.ou.asgbook.distance.Euclidean-DTPropagation.getDistanceTransform (LatLonGrid *data*, int *thresh*) [virtual]

At every pixel, finds the square of the Euclidean distance to the nearest pixel > thresh.

Implements edu.ou.asgbook.distance.EuclideanDT.

```
                                                                          {
        int nrows = data.getNumLat();
        int ncols = data.getNumLon();
        final int MAXDIST = nrows * nrows + ncols * ncols;
        LatLonGrid dist = new LatLonGrid(nrows, ncols, MAXDIST,
                data.getNwCorner(), data.getLatRes(), data.getLonRes());
        dist.fill(dist.getMissing());
        for (int i = 0; i < nrows; ++i)
            for (int j = 0; j < ncols; ++j) {
                if (data.getValue(i, j) > thresh) {
                    dist.setValue(i, j, 0);
                    propagate(dist, i, j, 250*250);
                }
            }
        return dist;
    }
```

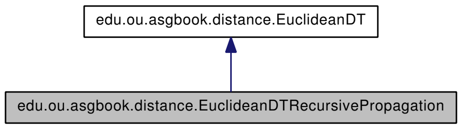
6.28.2.2 static void edu.ou.asgbook.distance.EuclideanDTPropagation.main (String[ ] *args*) throws Exception [static]

```
                                                              {
        File out = OutputDirectory.getDefault("distance");
        LatLonGrid popdensity = GlobalPopulation
                .read(GlobalPopulation.NORTHAMERICA);

        EuclideanDT transform = new EuclideanDTPropagation();
        LatLonGrid edt = transform.getDistanceTransform(popdensity, 50);

        // write it clamped out at a reasonable distance
        final int maxdist = 250 * 250;
        for (int i=0; i < edt.getNumLat(); ++i){
            for (int j=0; j < edt.getNumLon(); ++j){
                if ( edt.getValue(i,j) > maxdist){
                    edt.setValue(i,j, edt.getMissing() );
                }
            }
        }
        KmlWriter.write(edt, out, "edtupdate", PngWriter.createCoolToWarmColormap());
    }
```

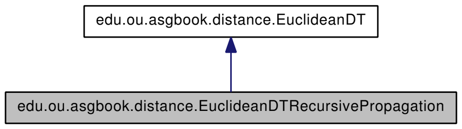
6.29 edu.ou.asgbook.distance.EuclideanDTRecursivePropagation Class Reference

Note that this class is only for illustrative purposes.

Inheritance diagram for edu.ou.asgbook.distance.EuclideanDTRecursivePropagation:


edu.ou.asgbook.distance.EuclideanDT
edu.ou.asgbook.distance.EuclideanDTRecursivePropagation

Collaboration diagram for edu.ou.asgbook.distance.EuclideanDTRecursivePropagation:


edu.ou.asgbook.distance.EuclideanDT
edu.ou.asgbook.distance.EuclideanDTRecursivePropagation

Public Member Functions

- Override LatLonGrid getDistanceTransform (LatLonGrid data, int thresh)

  At every pixel, finds the square of the Euclidean distance to the nearest pixel $>$ thresh.

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.29.1 Detailed Description

Note that this class is only for illustrative purposes.

It will not work because of stack overflow. Use the EuclideanDTPropagation implementation that replaces the recursion by a list.

Author:

v.lakshmanan

6.29.2 Member Function Documentation

6.29.2.1 Override LatLonGrid edu.ou.asgbook.distance.EuclideanDTRecursivePropagation.getDistanceTransform (LatLonGrid *data*, int *thresh*) [virtual]

At every pixel, finds the square of the Euclidean distance to the nearest pixel > thresh.

Implements edu.ou.asgbook.distance.EuclideanDT.

```
                                                                         {
        int nrows = data.getNumLat();
        int ncols = data.getNumLon();
        final int MAXDIST = nrows * nrows + ncols * ncols;
        LatLonGrid dist = new LatLonGrid(nrows, ncols, MAXDIST,
                data.getNwCorner(), data.getLatRes(), data.getLonRes());
        dist.fill(dist.getMissing());
        for (int i = 0; i < nrows; ++i)
            for (int j = 0; j < ncols; ++j) {
                if (data.getValue(i, j) > thresh) {
                    dist.setValue(i, j, 0);
                    propagate(dist, i, j, i, j);
                }
            }
        return dist;
    }
```

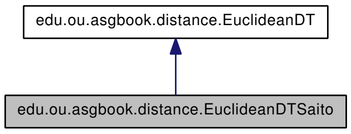
6.29.2.2 static void edu.ou.asgbook.distance.EuclideanDTRecursivePropagation.main (String[ ] *args*) throws Exception [static]

```
                                                             {
        File out = OutputDirectory.getDefault("distance");
        LatLonGrid popdensity = GlobalPopulation
                .read(GlobalPopulation.NORTHAMERICA);

        EuclideanDT transform = new EuclideanDTRecursivePropagation();
        LatLonGrid edt = transform.getDistanceTransform(popdensity, 50);

        // write it clamped out at a reasonable distance
        final int maxdist = 250 * 250;
        for (int i=0; i < edt.getNumLat(); ++i){
            for (int j=0; j < edt.getNumLon(); ++j){
                if ( edt.getValue(i,j) > maxdist){
                    edt.setValue(i,j, edt.getMissing() );
                }
            }
        }
        KmlWriter.write(edt, out, "edt", PngWriter.createCoolToWarmColormap());
    }
```

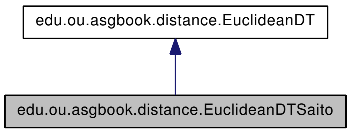
6.30 edu.ou.asgbook.distance.EuclideanDTSaito Class Reference

The Saito technique of computing the distance transform by calculating in the two directions separately.

Inheritance diagram for edu.ou.asgbook.distance.EuclideanDTSaito:


edu.ou.asgbook.distance.EuclideanDT
edu.ou.asgbook.distance.EuclideanDTSaito

Collaboration diagram for edu.ou.asgbook.distance.EuclideanDTSaito:


edu.ou.asgbook.distance.EuclideanDT
edu.ou.asgbook.distance.EuclideanDTSaito

Public Member Functions

- Override LatLonGrid getDistanceTransform (LatLonGrid data, int thresh)

  At every pixel, finds the square of the Euclidean distance to the nearest pixel > thresh.

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.30.1 Detailed Description

The Saito technique of computing the distance transform by calculating in the two directions separately.

Author:

v.lakshmanan

6.30.2 Member Function Documentation

6.30.2.1 Override LatLonGrid edu.ou.asgbook.distance.EuclideanDTSaito.getDistanceTransform (LatLonGrid *data*, int *thresh*) [virtual]

At every pixel, finds the square of the Euclidean distance to the nearest pixel > thresh.

Implements edu.ou.asgbook.distance.EuclideanDT.

```
                                                                       {
           try {
               return getDistanceTransform(data, thresh, null);
           } catch (Exception e) {
               throw new IllegalStateException();
           }
       }
```

6.30.2.2 static void edu.ou.asgbook.distance.EuclideanDTSaito.main (String[ ] *args*) throws Exception [static]

```
                                                               {
           File out = OutputDirectory.getDefault("euclideandt");
           LatLonGrid popdensity = GlobalPopulation
                   .read(GlobalPopulation.NORTHAMERICA);

           EuclideanDTSaito transform = new EuclideanDTSaito();
           LatLonGrid edt = transform.getDistanceTransform(popdensity, 50, out);
           writeClamped(edt, out, "edt");
       }
```

6.31 edu.ou.asgbook.transforms.FFT Class Reference

FFT based on Sedgewick and Wayne.

Static Public Member Functions

- static Complex[ ] fft (Complex[ ] x)

  Computes FFT of array whose length is a power of 2.

- static Complex[ ] ifft (Complex[ ] x)

  compute inverse FFT of array whose length is a power of 2

- static void main (String[ ] args)

Classes

- class **Complex**

6.31.1 Detailed Description

FFT based on Sedgewick and Wayne.

Author:

v.lakshmanan

6.31.2 Member Function Documentation

6.31.2.1 static Complex [ ] edu.ou.asgbook.transforms.FFT.fft (Complex[ ] *x*) `[static]`

Computes FFT of array whose length is a power of 2.

```
                                             {
           int N = x.length;

           if (N == 1){
               return new Complex[] { x[0] };
           } else if (N % 2 != 0) {
               throw new IllegalArgumentException("N is not a power of 2");
           }

           // Break the array down into two parts and perform FFT of each piece
           Complex[] part = new Complex[N / 2];
```

```
        for (int k = 0; k < N / 2; k++) {
            part[k] = x[2 * k]; // even terms
        }
        Complex[] evenfft = fft(part);
        for (int k = 0; k < N / 2; k++) {
            part[k] = x[2 * k + 1]; // odd terms
        }
        Complex[] oddfft = fft(part);

        // combine
        Complex[] y = new Complex[N];
        for (int k = 0; k < N / 2; k++) {
            double kth = -2 * k * Math.PI / N;
            Complex wk = new Complex(Math.cos(kth), Math.sin(kth));
            y[k] = evenfft[k].add(wk.multiply(oddfft[k]));
            y[k + N / 2] = evenfft[k].subtract(wk.multiply(oddfft[k]));
        }
        return y;
    }
```

6.31.2.2 static Complex [ ] edu.ou.asgbook.transforms.FFT.ifft (Complex[ ] *x*) [static]

compute inverse FFT of array whose length is a power of 2

```
                                          {
        int N = x.length;

        // Conjugate x
        Complex[] y = new Complex[N];
        for (int i = 0; i < N; i++) {
            y[i] = x[i].conjugate();
        }

        // compute forward FFT
        y = fft(y);

        // Conjugate result and divide by N
        for (int i = 0; i < N; i++) {
             y[i] = y[i].conjugate().divide(N);
         }

         return y;

    }
```

6.31.2.3 static void edu.ou.asgbook.transforms.FFT.main (String[ ] *args*) [static]

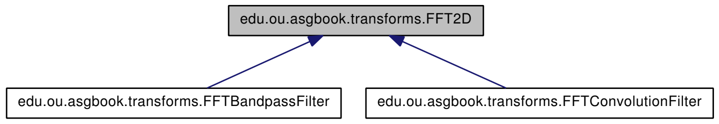
```
                                      {
         DecimalFormat df = new DecimalFormat("0.0");
```

```
        // FFT( rect ) should be a sinc function
        FFT.Complex[] input = new FFT.Complex[32];
        for (int i=0; i < input.length; ++i){
            input[i] = new FFT.Complex(0,0);
        }
        for (int i=input.length/3; i < 2*input.length/3; ++i){
            input[i] = new FFT.Complex(1, 0);
        }
        FFT.Complex[] output = fft(input);
        for (int i=0; i < output.length; ++i){
            System.out.print(df.format(output[i].norm()) + " ");
        }
        System.out.println();
        FFT.Complex[] reverse = ifft(output);
        for (int i=0; i < reverse.length; ++i){
            System.out.print(df.format(reverse[i].norm()) + " ");
        }
        System.out.println();
    }
```

6.32 edu.ou.asgbook.transforms.FFT2D Class Reference

Two-dimensional FFT.

Inheritance diagram for edu.ou.asgbook.transforms.FFT2D:


edu.ou.asgbook.transforms.FFT2D
edu.ou.asgbook.transforms.FFTBandpassFilter
edu.ou.asgbook.transforms.FFTConvolutionFilter

Static Public Member Functions

- static Complex[ ][ ] ifft (Complex[ ][ ] input)
- static Complex[ ][ ] zeropad (LatLonGrid data)
- static Complex[ ][ ] zeropad (double[ ][ ] data)
- static Complex[ ][ ] zeropad (double[ ][ ] data, int nrows, int ncols)
- static Complex[ ][ ] fft (Complex[ ][ ] data)
- static void main (String[ ] args) throws Exception

6.32.1 Detailed Description

Two-dimensional FFT.

Author:

valliappa.lakshmanan

6.32.2 Member Function Documentation

6.32.2.1 static Complex [ ][ ] edu.ou.asgbook.transforms.FFT2D.fft (Complex *data*[ ][ ]) [static]

```
                                                      {
           int nrows = data.length;
           int ncols = data[0].length;

           // compute fft row-by-row
           Complex[][] rowwise = new Complex[nrows][];
           for (int i=0; i < nrows; ++i){
               rowwise[i] = FFT.fft(data[i]);
           }

```

```
        // on the result, compute fft column by column
        Complex[][] result = new Complex[nrows][ncols];
        {
            Complex[] tmp = new Complex[nrows];
            for (int j=0; j < ncols; ++j){
                for (int i=0; i < nrows; ++i){
                    tmp[i] = rowwise[i][j];
                }
                Complex[] tmp2 = FFT.fft(tmp);
                for (int i=0; i < nrows; ++i){
                    result[i][j] = tmp2[i];
                }
            }
        }

        return result;
    }
```

6.32.2.2 static Complex [ ][ ] edu.ou.asgbook.transforms.FFT2D.ifft (Complex *input*[ ][ ]) [static]

```
                                                                  {
        // compute ifft row-wise, then column-wise
        int nrows = input.length;
        int ncols = input[0].length;
        Complex[][] rowwise = new Complex[nrows][];
        for (int i=0; i < nrows; ++i){
            rowwise[i] = FFT.ifft(input[i]);
        }
        Complex[][] result = new Complex[nrows][ncols];
        Complex[] tmp = new Complex[nrows];
        for (int j=0; j < ncols; ++j){
            for (int i=0; i < nrows; ++i){
                tmp[i] = rowwise[i][j];
            }
            Complex[] tmp2 = FFT.ifft(tmp);
            for (int i=0; i < nrows; ++i){
                result[i][j] = tmp2[i];
            }
        }
        return result;
    }
```

6.32.2.3 static void edu.ou.asgbook.transforms.FFT2D.main (String[ ] *args*) throws Exception [static]

Reimplemented in edu.ou.asgbook.transforms.FFTBandpassFilter, and edu.ou.asgbook.transforms.FFTConvolutionFilter.

```
                                                                          {
         DecimalFormat df = new DecimalFormat("0.0");
```

```
        // FFT( rect ) should be a sinc function
        FFT.Complex[][] input = new FFT.Complex[8][8];
        for (int i=0; i < input.length; ++i){
            for (int j=0; j < input[i].length; ++j){
                input[i][j] = new FFT.Complex(0,0);
            }
        }
        for (int i=input.length/3; i < 2*input.length/3; ++i){
            for (int j=input[i].length/3; j < 2*input[i].length/3; ++j){
                input[i][j] = new FFT.Complex(1, 0);
            }
        }
        FFT.Complex[][] output = fft(input);
        for (int i=0; i < output.length; ++i){
            for (int j=0; j < input[i].length; ++j){
                System.out.print(df.format(output[i][j].norm()) + " ");
            }
            System.out.println();
        }
        System.out.println();
        FFT.Complex[][] reverse = ifft(output);
        for (int i=0; i < reverse.length; ++i){
            for (int j=0; j < reverse[i].length; ++j){
                System.out.print(df.format(reverse[i][j].norm()) + " ");
            }
            System.out.println();
        }
        System.out.println();
    }
```

6.32.2.4 static Complex [ ][ ] edu.ou.asgbook.transforms.FFT2D.zeropad (double *data*[ ][ ], int *nrows*, int *ncols*) [static]

```
                                                                        {
       Complex[][] result = new Complex[nrows][ncols];
       Complex ZERO = new Complex(0,0);
       for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
           result[i][j] = ZERO;
       }
       for (int i=0; i < data.length; ++i){
           for (int j=0; j < data[0].length; ++j){
               result[i][j] = new Complex(data[i][j], 0);
           }
       }
       return result;
   }
```

6.32.2.5 static Complex [ ][ ] edu.ou.asgbook.transforms.FFT2D.zeropad (double *data*[ ][ ]) [static]

```
                                                 {
       int nrows = getNextPowerOf2(data.length);
```

```
        int ncols = getNextPowerOf2(data[0].length);
        return zeropad(data, nrows, ncols);
    }
```

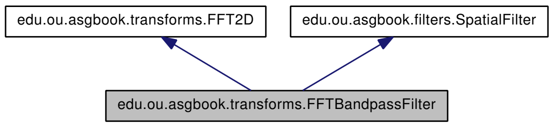
6.32.2.6 static Complex [ ][ ] edu.ou.asgbook.transforms.FFT2D.zeropad (LatLonGrid *data*) [static]

```
                                                            {
        int nrows = getNextPowerOf2(data.getNumLat());
        int ncols = getNextPowerOf2(data.getNumLon());
        Complex[][] result = new Complex[nrows][ncols];
        Complex ZERO = new Complex(0,0);
        for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
            result[i][j] = ZERO;
        }
        for (int i=0; i < data.getNumLat(); ++i){
            for (int j=0; j < data.getNumLon(); ++j){
                result[i][j] = new Complex(data.getValue(i,j), 0);
            }
        }
        return result;
    }
```

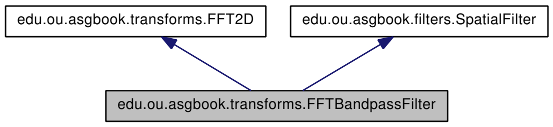
6.33 edu.ou.asgbook.transforms.FFTBandpassFilter Class Reference

Removes noise (high frequencies) and the gross signal (low frequencies).

Inheritance diagram for edu.ou.asgbook.transforms.FFTBandpassFilter:


edu.ou.asgbook.transforms.FFT2D
edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.transforms.FFTBandpassFilter

Collaboration diagram for edu.ou.asgbook.transforms.FFTBandpassFilter:


edu.ou.asgbook.transforms.FFT2D
edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.transforms.FFTBandpassFilter

Public Member Functions

- FFTBandpassFilter (double minr, double maxr)

  Supply numbers in the range (0,1) where 1 is the full dynamic range.

- Override LatLonGrid filter (LatLonGrid input)
- LatLonGrid convolve (LatLonGrid data)

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.33.1 Detailed Description

Removes noise (high frequencies) and the gross signal (low frequencies).

Author:

valliappa.lakshmanan

6.33.2 Constructor & Destructor Documentation

6.33.2.1 edu.ou.asgbook.transforms.FFTBandpassFilter.FFTBandpassFilter (double *minr*, double *maxr*)

Supply numbers in the range (0,1) where 1 is the full dynamic range.

```
                                                     {
        this.minr = minr;
        this.maxr = maxr;
    }
```

6.33.3 Member Function Documentation

6.33.3.1 LatLonGrid edu.ou.asgbook.transforms.FFTBandpassFilter.convolve (LatLonGrid *data*)

```
                                              {
        Complex[][] in1 = zeropad(data);
        final int nrows = in1.length;
        final int ncols = in1[0].length;

        // compute the fft
        in1 = fft(in1);

        // the fft is arranged in quadrants, so we need to be careful
        // to remove the corresponding data in all the quadrants
        Complex zero = new Complex(0,0);
        double diag = Math.sqrt(nrows*nrows+ncols*ncols)/4;
        for (int i=0; i < nrows/2; ++i){
            for (int j=0; j < ncols/2; ++j){
                double r = Math.sqrt(i*i + j*j)/diag;
                if (r < minr || r > maxr){
                    in1[i][j] = zero; // 1st quadrant
                    in1[nrows-i-1][j] = zero; // 3rd quadrant
                    in1[i][ncols-j-1] = zero; // 2nd quandrant
                    in1[nrows-i-1][ncols-i-1] = zero; // 4th quadrant
                }
            }
        }

        // take ifft
        Complex[][] result = ifft(in1);

        // return real part, rounded off
        LatLonGrid out = LatLonGrid.copyOf(data);
        for (int i=0; i < out.getNumLat(); ++i) for (int j=0; j < out.getNumLon(); ++j){
            out.setValue(i,j, (int)Math.round(result[i][j].real));
        }
        return out;
    }
```

6.33.3.2 Override LatLonGrid edu.ou.asgbook.transforms.FFTBandpassFilter.filter (LatLonGrid *input*)

```
                                                {
        return convolve(input);
    }
```

6.33.3.3 static void edu.ou.asgbook.transforms.FFTBandpassFilter.main (String[ ] *args*) throws Exception [static]

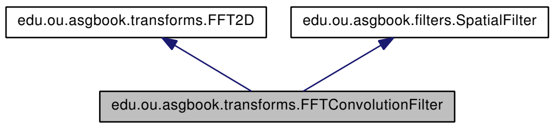
Reimplemented from edu.ou.asgbook.transforms.FFT2D.

```
                                                             {
        // create output directory
        File out = OutputDirectory.getDefault("fftbandpass");

        // read input
        LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new GlobalPopulati
        KmlWriter.write(popdensity, out, "orig", PngWriter.createCoolToWarmColormap());
        LatLonGrid sm = new FFTBandpassFilter(0,0.1).convolve(popdensity);
        KmlWriter.write(sm, out, "bp0_10", PngWriter.createCoolToWarmColormap());
        sm = new FFTBandpassFilter(0,0.2).convolve(popdensity);
        KmlWriter.write(sm, out, "bp0_20", PngWriter.createCoolToWarmColormap());
        sm = new FFTBandpassFilter(0.2,0.8).convolve(popdensity);
        KmlWriter.write(sm, out, "bp20_80", PngWriter.createCoolToWarmColormap());
        sm = new FFTBandpassFilter(0.8,1.0).convolve(popdensity);
        KmlWriter.write(sm, out, "bp80_100", PngWriter.createCoolToWarmColormap());
    }
```

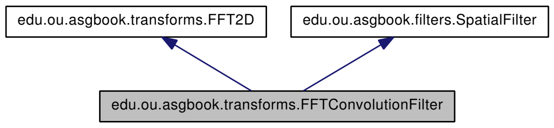
6.34 edu.ou.asgbook.transforms.FFTConvolutionFilter Class Reference

An optimization for convolution using FFTs.

Inheritance diagram for edu.ou.asgbook.transforms.FFTConvolutionFilter:


edu.ou.asgbook.transforms.FFT2D
edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.transforms.FFTConvolutionFilter

Collaboration diagram for edu.ou.asgbook.transforms.FFTConvolutionFilter:


edu.ou.asgbook.transforms.FFT2D
edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.transforms.FFTConvolutionFilter

Public Member Functions

- FFTConvolutionFilter (double[ ][ ] coeffs)
- LatLonGrid convolve (LatLonGrid data)
- Override LatLonGrid filter (LatLonGrid input)

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.34.1 Detailed Description

An optimization for convolution using FFTs.

Author:

valliappa.lakshmanan

6.34.2 Constructor & Destructor Documentation

6.34.2.1 edu.ou.asgbook.transforms.FFTConvolutionFilter.FFTConvolution-Filter (double *coeffs*[ ][ ])

```
                                                          {
          this.coeffs = coeffs;
      }
```

6.34.3 Member Function Documentation

6.34.3.1 LatLonGrid edu.ou.asgbook.transforms.FFTConvolution-Filter.convolve (LatLonGrid *data*)

```
                                                   {
          Complex[][] in1 = zeropad(data);
          int nrows = in1.length;
          int ncols = in1[0].length;
          Complex[][] in2 = zeropad(coeffs, nrows, ncols );

          // compute their ffts
          in1 = fft(in1);
          in2 = fft(in2);

          // multiply point by point (this by the conjugate of other)
          for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
              in1[i][j] = in1[i][j].multiply(in2[i][j].conjugate());
          }
          // take ifft
          Complex[][] result = ifft(in1);

          // return real part, rounded off
          LatLonGrid out = LatLonGrid.copyOf(data);
          for (int i=0; i < out.getNumLat(); ++i) for (int j=0; j < out.getNumLon(); ++j){
              out.setValue(i,j, (int)Math.round(result[i][j].real));
          }
          return out;
      }
```

6.34.3.2 Override LatLonGrid edu.ou.asgbook.transforms.FFTConvolution-Filter.filter (LatLonGrid *input*)

```
                                                   {
          return convolve(input);
      }
```

6.34.3.3 static void edu.ou.asgbook.transforms.FFTConvolutionFilter.main (String[ ] *args*) throws Exception [static]

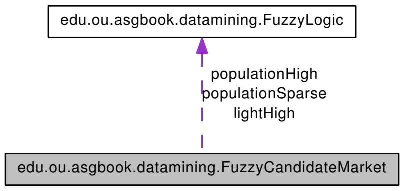
Reimplemented from edu.ou.asgbook.transforms.FFT2D.

```
                                                            {
          // create output directory
          File out = OutputDirectory.getDefault("fftconv");

          // read input
          LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new Glo
          System.out.println("Grid size=" + popdensity.getNumLat() + "x" + popdensity.getNumLo
          KmlWriter.write(popdensity, out, "fullgrid", PngWriter.createCoolToWarmColormap());
          double[][] coeffs = ConvolutionFilter.gauss(301, 301);
          long timer = System.currentTimeMillis();
          LatLonGrid sm = new FFTConvolutionFilter(coeffs).convolve(popdensity);
          long ffttime = System.currentTimeMillis() - timer;
          KmlWriter.write(sm, out, "fftgauss", PngWriter.createCoolToWarmColormap());

          // do it in spatial domain
          timer = System.currentTimeMillis();
          LatLonGrid sm2 = new ConvolutionFilter(coeffs).convolve(popdensity);
          long spatialtime = System.currentTimeMillis() - timer;
          KmlWriter.write(sm2, out, "spgauss", PngWriter.createCoolToWarmColormap());

          double improvement = 100*((double)(spatialtime - ffttime))/spatialtime;
          System.out.println("The FFT technique took " + ffttime + "ms whereas the spatial tec
      }
```

6.35 edu.ou.asgbook.datamining.FuzzyCandidateMarket Class Reference

Uses heuristic rules to choose the next market to enter.

Collaboration diagram for edu.ou.asgbook.datamining.FuzzyCandidateMarket:


edu.ou.asgbook.datamining.FuzzyLogic
populationHigh
populationSparse
lightHigh
edu.ou.asgbook.datamining.FuzzyCandidateMarket

Public Member Functions

- FuzzyCandidateMarket ()
- int isGoodCandidate (double population, double lightIntensity)

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.35.1 Detailed Description

Uses heuristic rules to choose the next market to enter.

Author:

valliappa.lakshmanan

6.35.2 Constructor & Destructor Documentation

6.35.2.1 edu.ou.asgbook.datamining.FuzzyCandidateMarket.FuzzyCandidateMarket ()

```
                                   {
        lightHigh = new FuzzyLogic.IsHigh(30, 70);
        populationSparse = new FuzzyLogic.IsLow(5,10);
        populationHigh = new FuzzyLogic.IsHigh(30,80);
    }
```

6.35.3 Member Function Documentation

6.35.3.1 int edu.ou.asgbook.datamining.FuzzyCandidateMarket.isGoodCandidate (double *population*, double *lightIntensity*)

```
                                                                                {
        // apply the basic rules
        FuzzyLogic.Fuzzy highlight = lightHigh.apply(lightIntensity);
        FuzzyLogic.Fuzzy popSparse = populationSparse.apply(population);
        FuzzyLogic.Fuzzy popHigh = populationHigh.apply(population);

        // if high light and moderate population density ...
        FuzzyLogic.Fuzzy popModerate = popSparse.not().and( popHigh.not() );
        FuzzyLogic.Fuzzy result = popModerate.and(highlight);

        return (int) Math.round(result.getValue()*10);
    }
```

6.35.3.2 static void edu.ou.asgbook.datamining.FuzzyCandidateMarket.main (String[ ] *args*) throws Exception [static]

```
                                                                        {
        // create output directory
        File out = OutputDirectory.getDefault("fuzzy");

        // read input (crop to cover Spain)
        LatLonGrid lights = NightimeLights.read(NightimeLights.WORLD).crop(980, 4080, 220, 3
        LatLonGrid pop    = GlobalPopulation.read(GlobalPopulation.WORLD).crop(980, 4080, 22

        // sanity check: are both grids correctly geolocated?
        System.out.println("Lights nwcorner: " + lights.getNwCorner());
        System.out.println("Population nwcorner: " + pop.getNwCorner());

        // apply fuzzy logic
        FuzzyCandidateMarket rules = new FuzzyCandidateMarket();
        LatLonGrid result = LatLonGrid.copyOf(lights);
        result.fill(0);
        result.setMissing(0);
        for (int i=0; i < result.getNumLat(); ++i){
            for (int j=0; j < result.getNumLon(); ++j){
                result.setValue(i,j, rules.isGoodCandidate(pop.getValue(i,j), lights.getValu
            }
        }

        // write out as image, for viewing
        KmlWriter.write(lights, out, "fzlights", PngWriter.createCoolToWarmColormap());
        KmlWriter.write(pop, out, "fzpop", PngWriter.createCoolToWarmColormap());
        KmlWriter.write(result, out, "candidatepixels", PngWriter.createCoolToWarmColormap()

        // find cities from population data using watershed
        EnhancedWatershedSegmenter seg = new EnhancedWatershedSegmenter(10, 1, 130, 10, 5);
        LabelResult label = seg.label(pop);
        RegionProperty[] popProps = RegionProperty.compute(label, pop);
        RegionProperty[] lightProps = RegionProperty.compute(label, lights);

        List<LatLon> points = new ArrayList<LatLon>();
        List<String> names = new ArrayList<String>();
        int[] howgood = new int[popProps.length];
        for (int i=1; i < howgood.length; ++i){
            howgood[i] = rules.isGoodCandidate(popProps[i].getCval(), lightProps[i].getCval());
            if (howgood[i] > 5){
                points.add( result.getLocation(popProps[i].getCx(), popProps[i].getCy()) );
                names.add( " " + howgood[i]);
                System.out.println( points.get(points.size()-1) + " " + howgood[i]);
            }
        }
        KmlWriter.write(points, names, out, "candidates");

         LatLonGrid candidateCities = LatLonGrid.copyOf(result);
         for (int i=0; i < candidateCities.getNumLat(); ++i){
             for (int j=0; j < candidateCities.getNumLon(); ++j){
                 int regno = label.label.getValue(i,j);
                 if (regno > 0){
                     candidateCities.setValue(i, j, howgood[regno]);
                 } else {
                     candidateCities.setValue(i, j, 0);
                 }
             }
         }
         KmlWriter.write(candidateCities, out, "candidateCities", PngWriter.createCoolToWarmColormap())
    }
```

6.36 edu.ou.asgbook.datamining.FuzzyLogic Class Reference

A simple fuzzy logic engine.

Classes

- class **Aggregate**

  Simply applies an equal weighting to all of the values.

- class **Fuzzy**
- class **IsAbout**
- class **IsHigh**
- class **IsLow**
- interface Rule

6.36.1 Detailed Description

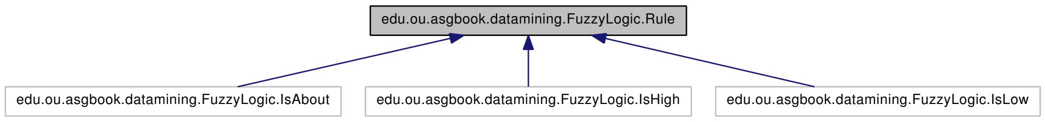
A simple fuzzy logic engine.

Author:

valliappa.lakshmanan

6.37 edu.ou.asgbook.datamining.FuzzyLogic.Rule Interface Reference

Inheritance diagram for edu.ou.asgbook.datamining.FuzzyLogic.Rule:


edu.ou.asgbook.datamining.FuzzyLogic.Rule
edu.ou.asgbook.datamining.FuzzyLogic.IsAbout
edu.ou.asgbook.datamining.FuzzyLogic.IsHigh
edu.ou.asgbook.datamining.FuzzyLogic.IsLow

Public Member Functions

- Fuzzy apply (double value)

6.37.1 Member Function Documentation

6.37.1.1 Fuzzy edu.ou.asgbook.datamining.FuzzyLogic.Rule.apply (double *value*)

6.38 edu.ou.asgbook.gmm.GaussianComponent Class Reference

Component of a Gaussian Mixture Model.

Public Member Functions

- GaussianComponent (double cx, double cy, double varx, double vary, double sigmaxy, double inwt)

    Initialize with known values.

- GaussianComponent (double cx, double cy, double varx, double vary)
- double getWeight ()
- double getCx ()
- double getCy ()
- double getSigmax ()
- double getSigmay ()
- double getSigmaxy ()
- GaussianComponent (Pixel[ ] pixels, double[ ] wts)

    Finds best fit (the M-step in E-M).

- double computeProbabilityDensityAt (Pixel p)
- double computeProbabilityDensityAt (double x, double y)

    Value of Normal function at x,y given these parameters.

- boolean isValid ()
- Override String toString ()

Classes

- class **Expectation**

6.38.1 Detailed Description

Component of a Gaussian Mixture Model.

Author:

valliappa.lakshmanan

6.38.2 Constructor & Destructor Documentation

6.38.2.1 edu.ou.asgbook.gmm.GaussianComponent.GaussianComponent (double *cx*, double *cy*, double *varx*, double *vary*, double *sigmaxy*, double *inwt*)

Initialize with known values.

```
                                    {
        mux = cx;
        muy = cy;
        sxx = varx;
        syy = vary;
        sxy = sigmaxy;
        wt = inwt;
        det = Math.abs(sxx * syy - sxy * sxy);
        denom = 2 * Math.PI * Math.sqrt(det);
    }
```

6.38.2.2 edu.ou.asgbook.gmm.GaussianComponent.GaussianComponent (double *cx*, double *cy*, double *varx*, double *vary*)

```
                                                                        {
        this(cx, cy, varx, vary, 0, 1);
    }
```

6.38.2.3 edu.ou.asgbook.gmm.GaussianComponent.GaussianComponent (Pixel[ ] *pixels*, double[ ] *wts*)

Finds best fit (the M-step in E-M).

```
                                                            {
        int n_pts = pixels.length;
        if (wts.length != pixels.length){
            throw new IllegalArgumentException("Array lengths have to match");
        }
        // compute pi_k (wt)
        wt = 0;
        for (int i = 0; i < n_pts; ++i) {
            wt += wts[i];
        }
        wt /= n_pts;

        // mean
        Expectation wm_x = new Expectation(), wm_y = new Expectation();
        for (int i = 0; i < n_pts; ++i) {
            wm_x.update(pixels[i].getX(), wts[i]);
            wm_y.update(pixels[i].getY(), wts[i]);
        }
        mux = wm_x.result();
        muy = wm_y.result();

        // covariance matrix
         Expectation wv_x = new Expectation();
         Expectation wv_y = new Expectation();
         Expectation cv_xy = new Expectation();
         for (int i = 0; i < n_pts; ++i) {
             double dx = pixels[i].getX() - mux;
             double dy = pixels[i].getY() - muy;
             wv_x.update(dx * dx, wts[i]);
             wv_y.update(dy * dy, wts[i]);
             cv_xy.update(dx * dy, wts[i]);
         }
         sxx = wv_x.result();
         syy = wv_y.result();
         sxy = cv_xy.result();

         final double EPSILON = 0.01; // at-least 1/10 pixel of variance ...
         if (sxx < EPSILON || syy < EPSILON) {
             det = denom = 0;
             return;
         }

         // normalizing constant
         det = (sxx * syy - sxy * sxy);
         denom = 2 * Math.PI * Math.sqrt(Math.abs(det)); // always positive

    }
```

6.38.3 Member Function Documentation

6.38.3.1 double edu.ou.asgbook.gmm.GaussianComponent.compute-ProbabilityDensityAt (double *x*, double *y*)

Value of Normal function at x,y given these parameters.

You typically want to weight this contribution by getWeight() This goes into the E-step in E-M.

```
                                                                     {
         if (denom < 0.00001) {
             return 0;
         } // singular
         double dx = x - mux;
         double dy = y - muy;
         double term = -(syy * dx * dx - 2 * sxy * dx * dy + sxx * dy * dy);
         double num = Math.exp(term / (2 * det));
         double result = num / denom;
         if (result > 1) {
             // usually because of numerical instability
             return 0;
         }
         return result;
```

```

    }
```

6.38.3.2 double edu.ou.asgbook.gmm.GaussianComponent.compute-ProbabilityDensityAt (Pixel *p*)

```
                                                        {
         return computeProbabilityDensityAt(p.getX(), p.getY());
    }
```

6.38.3.3 double edu.ou.asgbook.gmm.GaussianComponent.getCx ()

```
                           {
        return mux;
    }
```

6.38.3.4 double edu.ou.asgbook.gmm.GaussianComponent.getCy ()

```
                           {
        return muy;
    }
```

6.38.3.5 double edu.ou.asgbook.gmm.GaussianComponent.getSigmax ()

```
                               {
        return Math.sqrt(sxx);
    }
```

6.38.3.6 double edu.ou.asgbook.gmm.GaussianComponent.getSigmaxy ()

```
                                {
        return sxy;
    }
```

6.38.3.7 double edu.ou.asgbook.gmm.GaussianComponent.getSigmay ()

```
                               {
        return Math.sqrt(syy);
    }
```

6.38.3.8 double edu.ou.asgbook.gmm.GaussianComponent.getWeight ()

```
                            {
        return wt;
    }
```

6.38.3.9 boolean edu.ou.asgbook.gmm.GaussianComponent.isValid ()

```
                              {
        return denom > 0;
    }
```

6.38.3.10 Override String edu.ou.asgbook.gmm.GaussianComponent.toString ()

```
                              {
        return "center=(" + mux + "," + muy + ") covar=(" + sxx + "," + sxy
                + "," + syy + ") wt=" + wt;
    }
```

6.39 edu.ou.asgbook.gmm.GaussianMixtureModel Class Reference

A parametric approximation of a spatial grid as a sum of Gaussians.

Public Member Functions

- GaussianMixtureModel (LatLonGrid input, int numModels)
- GaussianComponent[ ] getMixture ()
- void setValueInGrid (LatLonGrid data, double scale)

  Estimates the value at each pixel in the grid based on GMM The values will be 0-1, so provide a scale in order to make them integers.

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.39.1 Detailed Description

A parametric approximation of a spatial grid as a sum of Gaussians.

Author:

valliappa.lakshmanan

6.39.2 Constructor & Destructor Documentation

6.39.2.1 edu.ou.asgbook.gmm.GaussianMixtureModel.GaussianMixtureModel (LatLonGrid *input*, int *numModels*)

```
                                                                {
          final int MIN_DISTSQ = 100; // distance between initial centers in px
          initGMM(LevelSet.newInstance(input), numModels, MIN_DISTSQ);
          final int MAX_ITER = 10;
          final double MIN_IMPROVEMENT = 0.01; // 1 percent
          tuneGMM(input.asPixels(), MAX_ITER, MIN_IMPROVEMENT);
      }
```

6.39.3 Member Function Documentation

6.39.3.1 GaussianComponent [ ] edu.ou.asgbook.gmm.GaussianMixture-Model.getMixture ()

```
                                         {
        return mixture.toArray(new GaussianComponent[0]);
    }
```

6.39.3.2 static void edu.ou.asgbook.gmm.GaussianMixtureModel.main (String[ ] *args*) throws Exception [static]

```
                                                             {
        LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new Gl

        File out = OutputDirectory.getDefault("gmmpopdensity");
        KmlWriter.write(popdensity, out, "original", PngWriter.createCoolToWarmColormap());

        GaussianMixtureModel gmm = new GaussianMixtureModel(popdensity, 10);
        GaussianComponent[] fit = gmm.getMixture();
        List<LatLon> locs = new ArrayList<LatLon>();
        List<String> names = new ArrayList<String>();
        for (int i=0; i < fit.length; ++i){
            LatLon loc = popdensity.getLocation( fit[i].getCx(), fit[i].getCy() );
            String name = ("GMM#" + i + " ampl=" +  fit[i].getWeight() + " sigmax=" + fit[i
            System.out.println(" loc: " + loc + name);
            locs.add(loc);
            names.add(name);
        }
        KmlWriter.write(locs, names, out, "gmmcities");

        // write out the approximation
        gmm.setValueInGrid(popdensity, 500);
        KmlWriter.write(popdensity, out, "gmmapprox", PngWriter.createCoolToWarmColormap()
    }
```

6.39.3.3 void edu.ou.asgbook.gmm.GaussianMixtureModel.setValueInGrid (LatLonGrid *data*, double *scale*)

Estimates the value at each pixel in the grid based on GMM The values will be 0-1, so provide a scale in order to make them integers.

Parameters:

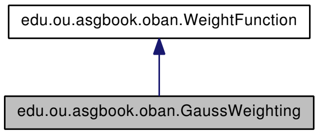
data

```
                                                             {
        data.setMissing(0);
        double peakval = 0;
        for (int m=0; m < mixture.size(); ++m){
            peakval = Math.max( peakval, mixture.get(m).getWeight() );
        }
        for (int i=0; i < data.getNumLat(); ++i){
            for (int j=0; j < data.getNumLon(); ++j){
                double raw = 0;
                for (int m=0; m < mixture.size(); ++m){
                    raw += mixture.get(m).computeProbabilityDensityAt(i,j);
                }
                data.setValue(i,j, (int) Math.round(raw*scale/peakval));
            }
        }
    }
```

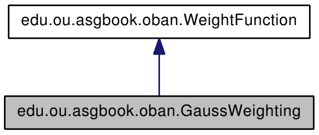
6.40 edu.ou.asgbook.oban.GaussWeighting Class Reference

An interpolation method that uses exp(-1/r^2).

Inheritance diagram for edu.ou.asgbook.oban.GaussWeighting:


edu.ou.asgbook.oban.WeightFunction
edu.ou.asgbook.oban.GaussWeighting

Collaboration diagram for edu.ou.asgbook.oban.GaussWeighting:


edu.ou.asgbook.oban.WeightFunction
edu.ou.asgbook.oban.GaussWeighting

Public Member Functions

- GaussWeighting (double sigma)
- Override double computeWt (double latdist, double londist)

  Subclasses implement a weighting function.

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.40.1 Detailed Description

An interpolation method that uses exp(-1/r^2).

Author:

Valliappa.Lakshmanan

6.40.2 Constructor & Destructor Documentation

6.40.2.1 edu.ou.asgbook.oban.GaussWeighting.GaussWeighting (double *sigma*)

Parameters:

sigma Set extent of gaussian in degrees

```
                                        {
        this.sigmasq = sigma * sigma;
        this.epsilonDistSq = (3*3)*(2*sigmasq); // at distance of 3*sigma in both directions
    }
```

6.40.3 Member Function Documentation

6.40.3.1 Override double edu.ou.asgbook.oban.GaussWeighting.computeWt (double *latdist*, double *londist*) `[virtual]`

Subclasses implement a weighting function.

If -ve value is returned, then the point will be considered too far away and not used in weighting.

Implements edu.ou.asgbook.oban.WeightFunction.

```
                                                             {
        double r2 = latdist * latdist + londist * londist;
        if ( r2 < epsilonDistSq ){
            return Math.exp(-r2 / (2*sigmasq));
        } else {
            return INVALID_WEIGHT;
        }
    }
```

6.40.3.2 static void edu.ou.asgbook.oban.GaussWeighting.main (String[ ] *args*) throws Exception `[static]`

```
                                                              {
        PointObservations data = DailyRainfall.read(DailyRainfall.TN_Oct2010);

        double sigma = ObjectiveAnalysisUtils.computeMeanDistance(data);
        System.out.println("Objectively analyzing " + data.getPoints().length + " pts with a mean separ
        WeightFunction wtFunc = new GaussWeighting(sigma);
        WeightedAverage analyzer = new WeightedAverage(wtFunc, 0.01, 0.01, 1);
        LatLonGrid grid = analyzer.analyze(data);

        // write output
        File out = OutputDirectory.getDefault("gaussoban");
        KmlWriter.write(grid, out, "Precip24H", PngWriter.createCoolToWarmColormap());
    }
```

6.41 edu.ou.asgbook.datamining.GdiPattern Class Reference

The training pattern for each city.

Public Member Functions

- Override String toString ()
- String toString (String colsep, String linesep)

Static Public Member Functions

- static GdiPattern[ ] findTrainingPattern (LabelResult cities, LatLonGrid population, LatLonGrid nightTimeLights, LatLonGrid gdiGrid)
- static void write (GdiPattern[ ] patterns, File outdir) throws IOException
- static void main (String[ ] args) throws Exception

6.41.1 Detailed Description

The training pattern for each city.

Author:

valliappa.lakshmanan

6.41.2 Member Function Documentation

6.41.2.1 static GdiPattern [ ] edu.ou.asgbook.datamining.GdiPattern.findTrainingPattern (LabelResult *cities*, LatLonGrid *population*, LatLonGrid *nightTimeLights*, LatLonGrid *gdiGrid*) [static]

```

        // for each city, compute the other properties
        RegionProperty[] pop = RegionProperty.compute(cities, population);
        RegionProperty[] lights = RegionProperty.compute(cities, nightTimeLights);
        RegionProperty[] gdi = RegionProperty.compute(cities, gdiGrid);
        GdiPattern[] patterns = new GdiPattern[pop.length];
        for (int i=1; i < patterns.length; ++i){
            patterns[i] = new GdiPattern();
            patterns[i].data[0] = pop[i].getCval();
            patterns[i].data[1] = lights[i].getCval();
            patterns[i].data[2] = gdi[i].getCval();
        }
        return patterns;
    }
```

6.41.2.2 static void edu.ou.asgbook.datamining.GdiPattern.main (String[ ] *args*) throws Exception [static]

```
                                                           {
       // create output directory
       File out = OutputDirectory.getDefault("gdipattern");
       final boolean SMALL = true;

       // read input (crop to cover Spain)
       LatLonGrid pop    = GlobalPopulation.read(GlobalPopulation.WORLD);
       if (SMALL){
           pop = pop.crop(980, 4080, 220, 350);
       }
       KmlWriter.write(pop, out, "pop", PngWriter.createRandomColormap());

       LatLonGrid nightTimeLights = NightimeLights.read(NightimeLights.WORLD).remapTo(pop);
       KmlWriter.write(nightTimeLights, out, "nighttimelights", PngWriter.createCoolToWarmColormap());

       LatLonGrid countries = CountryPolygons.readGrid(CountryPolygons.WORLD_GRID).remapTo(pop);
       KmlWriter.write(countries, out, "countries", PngWriter.createRandomColormap());

        LabelResult primary = PrimaryCities.findPrimaryCities(pop, countries, out);
        KmlWriter.write(primary.label, out, "primarycities", PngWriter.createRandomColormap());

        LatLonGrid gdiGrid = WorldBankGDI.readGrid(WorldBankGDI.WORLD_GRID).remapTo(pop);
        KmlWriter.write(gdiGrid, out, "gdism", PngWriter.createCoolToWarmColormap());

        // obtain pattern
        GdiPattern[] patterns = GdiPattern.findTrainingPattern(primary, pop, nightTimeLights, gdiGrid)
        System.out.println("Population & Lighting & GDI \\\\");
        for (int i=1; i < patterns.length; ++i){
            System.out.println(patterns[i]);
        }

        if (!SMALL){
            write(patterns, out); // for R
        }
    }
```

6.41.2.3 String edu.ou.asgbook.datamining.GdiPattern.toString (String *colsep*, String *linesep*)

```
                                                         {
        String result = "";
        for (int i=0; i < data.length; ++i){
            String sep = (i == data.length-1)? linesep : colsep;
            result += format(data[i]) + sep;
        }
        return result;
    }
```

6.41.2.4 Override String edu.ou.asgbook.datamining.GdiPattern.toString ()

```
                          {
        return toString(" & ", "\\\\"); // for LaTeX
    }
```

6.41.2.5 static void edu.ou.asgbook.datamining.GdiPattern.write (GdiPattern[ ] *patterns*, File *outdir*) throws IOException `[static]`

```
                                                                                          {
        PrintWriter writer = null;
        try {
            String name = outdir.getAbsolutePath() + "/gdipatterns.txt";
            writer = new PrintWriter( new FileWriter(name) );
            for (int i=1; i < patterns.length; ++i){
                writer.println(patterns[i].toString(" ","")); // for R
            }
        } finally {
            if (writer != null){
                writer.close();
            }
        }
    }
```

6.42 edu.ou.asgbook.dataset.GlobalPopulation Class Reference

Reads the ASCII population data available at http://sedac.ciesin.columbia.edu/gpw.

Static Public Member Functions

- static LatLonGrid read (Reader inputFile, DataTransform t)
- static LatLonGrid read (File file, DataTransform t) throws IOException

  reads data from a File.

- static LatLonGrid read (File file) throws IOException
- static void main (String[ ] args) throws Exception

Static Public Attributes

- static File WORLD = new File("data/popdensity/glp10ag.asc.gz")
- static File NORTHAMERICA = new File("data/popdensity/nap10ag.asc.gz")
- static File NORTHAMERICA1990 = new File("data/popdensity/nap90ag.asc.gz")

Classes

- class **LinearScaling**
- class **LogScaling**

6.42.1 Detailed Description

Reads the ASCII population data available at http://sedac.ciesin.columbia.edu/gpw.

Author:

Valliappa.Lakshmanan

6.42.2 Member Function Documentation

6.42.2.1 static void edu.ou.asgbook.dataset.GlobalPopulation.main (String[ ] *args*) throws Exception [static]

```
                                                                {
        // create output directory
        File out = OutputDirectory.getDefault("globalpop");

        // read input
        LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA).crop(9

        // write out as image, for viewing
        KmlWriter.write(popdensity, out, "popdensity", PngWriter.createCoolToWarmColormap()
        KmlWriter.write(GlobalPopulation.read(GlobalPopulation.WORLD, new LogScaling()), out

        // show impact of colormap and log scaling
        KmlWriter.write(popdensity, out, "rainbow", PngWriter.createHotColormap());
        popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new LogScaling())
        KmlWriter.write(popdensity, out, "logdensity", PngWriter.createCoolToWarmColormap()
        KmlWriter.write(popdensity, out, "lograinbow", PngWriter.createHotColormap());
    }
```

6.42.2.2 static LatLonGrid edu.ou.asgbook.dataset.GlobalPopulation.read (File *file*) throws IOException [static]

```
                                                                        {
        return read(file, new LinearScaling());
    }
```

6.42.2.3 static LatLonGrid edu.ou.asgbook.dataset.GlobalPopulation.read (File *file*, DataTransform *t*) throws IOException [static]

reads data from a File.

The File can be gzipped or uncompressed.

```
                                                                                    {
        return EsriGrid.read(file, t);
    }
```

6.42.2.4 static LatLonGrid edu.ou.asgbook.dataset.GlobalPopulation.read (Reader *inputFile*, DataTransform *t*) [static]

```
                                                                          {
        return EsriGrid.read(inputFile, t);
    }
```

6.42.3 Member Data Documentation

6.42.3.1 File edu.ou.asgbook.dataset.GlobalPopulation.NORTHAMERICA = new File("data/popdensity/nap10ag.asc.gz") `[static]`

6.42.3.2 File edu.ou.asgbook.dataset.Global-Population.NORTHAMERICA1990 = new File("data/popdensity/nap90ag.asc.gz") `[static]`

6.42.3.3 File edu.ou.asgbook.dataset.GlobalPopulation.WORLD = new File("data/popdensity/glp10ag.asc.gz") `[static]`

6.43 edu.ou.asgbook.imgstat.GraylevelCooccurence-Matrix Class Reference

Computes texture properties from a GLCM.

Public Types

- EASTWARD
- SOUTHWARD
- NORTHEAST
- SOUTHEAST
- enum Direction {

  EASTWARD, SOUTHWARD, NORTHEAST, SOUTHEAST,

  xadd, yadd = xadd yadd }

Public Member Functions

- GraylevelCooccurenceMatrix (LatLonGrid input, int x, int y, Direction dir, int hx, int hy, int min, int incr, int bins)
- double computeUniformity ()
- double computeEntropy ()
- double computeMaximumProbability ()
- double computeDifferenceMoment (int order)

Package Functions

- int findBin (int val, int missing, int min, int incr, int bins)

6.43.1 Detailed Description

Computes texture properties from a GLCM.

Author:

valliappa.lakshmanan

6.43.2 Member Enumeration Documentation

6.43.2.1 enum edu::ou::asgbook::imgstat::GraylevelCooccurence-Matrix::Direction

Enumerator:

EASTWARD

SOUTHWARD

NORTHEAST

SOUTHEAST

xadd

yadd

```
                         {
        EASTWARD(0,1),
        SOUTHWARD(1,0), // downward
        NORTHEAST(-1,1),
        SOUTHEAST(1,-1);
        public final int xadd;
        public final int yadd;
        private Direction(int xadd, int yadd){
            this.xadd = xadd;
            this.yadd = yadd;
        }
    }
```

6.43.3 Constructor & Destructor Documentation

6.43.3.1 edu.ou.asgbook.imgstat.GraylevelCooccurenceMatrix.Graylevel-CooccurenceMatrix (LatLonGrid *input*, int *x*, int *y*, Direction *dir*, int *hx*, int *hy*, int *min*, int *incr*, int *bins*)

```

        p = new double[bins+1][bins+1]; // last bin is for missing data
        int N = 0;
        for (int m=-hx; m <= hx; ++m){
            for (int n=-hy; n <= hy; ++n){
                int value1 = input.getMissing();
                if ( input.isValid(x+m,y+n) ){
                    value1 = input.getValue(x+m,y+n);
                }
                int value2 = input.getMissing();
                if ( input.isValid(x+m+dir.xadd,y+n+dir.yadd) ){
                    value2 = input.getValue(x+m+dir.xadd,y+n+dir.yadd);
                }
                int bin1 = findBin(value1,input.getMissing(),min,incr,bins);
                int bin2 = findBin(value2,input.getMissing(),min,incr,bins);
                p[bin1][bin2]++;
                ++N;
            }
        }
        for (int i=0; i < p.length; ++i) for (int j=0; j < p.length; ++j){
            p[i][j] /= N;
        }
    }
```

6.43.4 Member Function Documentation

6.43.4.1 double edu.ou.asgbook.imgstat.GraylevelCooccurence-Matrix.computeDifferenceMoment (int *order*)

```
                                                    {
        double result = 0;
        int N = p.length;
        for (int i=0; i < N; ++i) for (int j=0; j < N; ++j){
            if (i != j){
                result += p[i][j]*p[i][j]/Math.pow(Math.abs(i-j),order);
            }
        }
        return result;
    }
```

6.43.4.2 double edu.ou.asgbook.imgstat.GraylevelCooccurence-Matrix.computeEntropy ()

```
                                  {
        double result = 0;
        int N = p.length;
        for (int i=0; i < N; ++i) for (int j=0; j < N; ++j){
            if (p[i][j] > 0){
                result += p[i][j]*Math.log(p[i][j])/Math.log(2);
            }
        }
        return result;
    }
```

6.43.4.3 double edu.ou.asgbook.imgstat.GraylevelCooccurence-Matrix.computeMaximumProbability ()

```
                                             {
        double result = 0;
        int N = p.length;
        for (int i=0; i < N; ++i) for (int j=0; j < N; ++j){
            result = Math.max(result, p[i][j]);
        }
        return result;
    }
```

6.43.4.4 double edu.ou.asgbook.imgstat.GraylevelCooccurence-Matrix.computeUniformity ()

```
                                    {
        double result = 0;
        int N = p.length;
        for (int i=0; i < N; ++i) for (int j=0; j < N; ++j){
            result += p[i][j]*p[i][j];
        }
        return result;
    }
```

6.43.4.5 int edu.ou.asgbook.imgstat.GraylevelCooccurenceMatrix.findBin (int *val*, int *missing*, int *min*, int *incr*, int *bins*) [package]

```
                                                                {
        if (val != missing && val >= min) {
            int bin_no = (val - min) / incr;
            // last bin is unbounded
            if (bin_no >= bins)
                bin_no = bins - 1;
            return bin_no;
        }
        return bins; // for missing data
    }
```

6.44 edu.ou.asgbook.thinning.HilditchSkeletonization Class Reference

Hilditch method of skeletonizing a grid.

Static Public Member Functions

- static LatLonGrid findSkeleton (LatLonGrid input, int thresh, File out) throws Exception
- static void main (String[ ] args) throws Exception

Classes

- class **State**

6.44.1 Detailed Description

Hilditch method of skeletonizing a grid.

Author:

v.lakshmanan

6.44.2 Member Function Documentation

6.44.2.1 static LatLonGrid edu.ou.asgbook.thinning.Hilditch-Skeletonization.findSkeleton (LatLonGrid *input*, int *thresh*, File *out*) throws Exception [static]

```

        // threshold.  object=1 and background=0
        LatLonGrid binaryImage = new SimpleThresholder(thresh).threshold(input);
        if (out != null){
            KmlWriter.write(binaryImage, out, "thresh", PngWriter.createCoolToWarmColormap()
        }

        final int nx = binaryImage.getNumLat();
        final int ny = binaryImage.getNumLon();
        int numChanges;
        do {
            // compute ap, bp
            LatLonGrid ap = new LatLonGrid(nx,ny,-1,binaryImage.getNwCorner(),binaryImage.ge
            LatLonGrid bp = new LatLonGrid(nx,ny,-1,binaryImage.getNwCorner(),binaryImage.ge
            for (int i=1; i < (nx-1); ++i) for (int j=1; j < (ny-1); ++j){
                if ( binaryImage.getValue(i, j) > 0){
                    // find A(p) and B(p)
                    State state = new State( binaryImage.getValue(i-1,j-1) );
                    state.update( binaryImage.getValue(i-1, j) );
                    state.update( binaryImage.getValue(i-1, j+1) );
                    state.update( binaryImage.getValue(i, j+1) );
                    state.update( binaryImage.getValue(i+1, j+1) );
                    state.update( binaryImage.getValue(i+1, j) );
                    state.update( binaryImage.getValue(i+1, j-1) );
                    state.update( binaryImage.getValue(i, j-1) );
                    state.update( binaryImage.getValue(i-1, j-1) );
                    ap.setValue(i,j, state.ap);
                    bp.setValue(i,j, state.bp);
                }
            }

            // peel off pixel?
            numChanges = 0;
            LatLonGrid after = LatLonGrid.copyOf(binaryImage);
            for (int i=1; i < (nx-1); ++i) for (int j=1; j < (ny-1); ++j){
                if ( ap.getValue(i,j) == 1 && bp.getValue(i,j) >= 2 && bp.getValue(i,j) <= 6){
                    if ( ap.getValue(i-1,j) == 0 ||
                            binaryImage.getValue(i-1,j) == 0 ||
                            binaryImage.getValue(i,j+1) == 0 ||
                            binaryImage.getValue(i,j-1) == 0 ){
                        if ( ap.getValue(i,j+1) == 0 ||
                                binaryImage.getValue(i-1,j) == 0 ||
                                binaryImage.getValue(i,j+1) == 0 ||
                                binaryImage.getValue(i+1,j) == 0){
                            // peel
                            after.setValue(i,j, 0);
                            ++numChanges;
                        }
                    }
                }
            }
            binaryImage = after;
            System.out.println(numChanges + " pixels peeled off in iteration");
        } while (numChanges > 0);

        return binaryImage;
    }
```

6.44.2.2 static void edu.ou.asgbook.thinning.HilditchSkeletonization.main (String[ ] *args*) throws Exception [static]

```
                                                                {
        File out = OutputDirectory.getDefault("hilditchskeleton");
         LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new GlobalPopulat
         KmlWriter.write(popdensity, out, "orig", PngWriter.createCoolToWarmColormap());

         popdensity = new DilateErodeFilter(2,3).filter(popdensity);
         popdensity = new ErodeDilateFilter(2,3).filter(popdensity);
         KmlWriter.write(popdensity, out, "filledin", PngWriter.createCoolToWarmColormap());

         LatLonGrid result = findSkeleton(popdensity, 300, out);
         result.setMissing(0); // to make the 1s pop out
         KmlWriter.write(result, out, "skel", PngWriter.createCoolToWarmColormap());
     }
```

6.45 edu.ou.asgbook.histogram.Histogram Class Reference

A histogram is an empirical probability distribution.

Public Member Functions

- Histogram (int min, int incr, int nbins)

  Values below min are ignored but the last bin is unbounded.

- int getMin ()
- int getIncr ()
- int[ ] getHist ()
- void update (LatLonGrid data)
- int getCenterValue (int bin_no, int missing)
- int getBinNumber (int val, int missing)

  points outside the histogram have bin number of -1

- Override String toString ()
- float[ ] calcProb ()
- int getNumBins ()

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.45.1 Detailed Description

A histogram is an empirical probability distribution.

Author:

v.lakshmanan

6.45.2 Constructor & Destructor Documentation

6.45.2.1 edu.ou.asgbook.histogram.Histogram.Histogram (int *min*, int *incr*, int *nbins*)

Values below min are ignored but the last bin is unbounded.

Parameters:

min

incr

nbins

```
                                                    {
       super();
       this.min = min;
       this.incr = incr;
       this.hist = new int[nbins];
   }
```

6.45.3 Member Function Documentation

6.45.3.1 float [ ] edu.ou.asgbook.histogram.Histogram.calcProb ()

```
                            {
       float[] prob = new float[hist.length];
       int tot = 0;
       for (int i = 0; i < hist.length; ++i) {
           tot += hist[i];
       }
       if (tot > 0) {
           for (int i = 0; i < hist.length; ++i) {
               prob[i] = hist[i] / (float) tot;
           }
       }
       return prob;
   }
```

6.45.3.2 int edu.ou.asgbook.histogram.Histogram.getBinNumber (int *val*, int *missing*)

points outside the histogram have bin number of -1

```
                                                    {
       if (val != missing && val >= min) {
           int bin_no = (val - min) / incr;
           // last bin is unbounded
           if (bin_no >= hist.length)
               bin_no = hist.length - 1;
           return bin_no;
       }
       return -1;
   }
```

6.45.3.3 int edu.ou.asgbook.histogram.Histogram.getCenterValue (int *bin_no*, int *missing*)

```
                                                    {
        if (bin_no < 0){
            return missing;
        }
        return min + bin_no*incr + incr/2;
    }
```

6.45.3.4 int [ ] edu.ou.asgbook.histogram.Histogram.getHist ()

```
                          {
        return hist;
    }
```

6.45.3.5 int edu.ou.asgbook.histogram.Histogram.getIncr ()

```
                        {
        return incr;
    }
```

6.45.3.6 int edu.ou.asgbook.histogram.Histogram.getMin ()

```
                       {
        return min;
    }
```

6.45.3.7 int edu.ou.asgbook.histogram.Histogram.getNumBins ()

```
                           {
        return hist.length;
    }
```

6.45.3.8 static void edu.ou.asgbook.histogram.Histogram.main (String[ ] *args*) throws Exception [static]

```
                                                             {
        // create output directory
        File outdir = OutputDirectory.getDefault("hist");

        // read input
        LatLonGrid conus = SurfaceAlbedo.read(SurfaceAlbedo.CONUS, 100);

        // find histogram
        final int MIN = 0;
        final int MAX = 30;
        for (int incr = 1; incr < 10; incr += 2) {
            Histogram hist = new Histogram(MIN, incr, (MAX - MIN) / incr);
            hist.update(conus);
            System.out.println("INCR=" + incr + " nbins=" + hist.hist.length);
            System.out.println(hist);
            String filename = outdir.getAbsolutePath() + "/hist_" + incr
                    + ".txt";
            PrintWriter writer = new PrintWriter(new FileWriter(filename));
            writer.println(hist);
            writer.close();
            System.out.println("Wrote to " + filename);
        }
    }
```

6.45.3.9 Override String edu.ou.asgbook.histogram.Histogram.toString ()

```
                          {
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < hist.length; ++i) {
            int sval = min + i * incr;
            int eval = sval + incr;
            sb.append(sval);
            sb.append(" ");
            sb.append(eval);
            sb.append(" ");
            sb.append(hist[i]);
            sb.append("\n");
        }
        return sb.toString();
    }
```

6.45.3.10 void edu.ou.asgbook.histogram.Histogram.update (LatLonGrid *data*)

```
                                     {
        final int nrows = data.getNumLat();
        final int ncols = data.getNumLon();
        for (int i = 0; i < nrows; ++i)
            for (int j = 0; j < ncols; ++j) {
                int val = data.getValue(i, j);
                int bin_no = getBinNumber(val, data.getMissing());
                if (bin_no != -1 ){
                    hist[bin_no]++;
                }
            }
    }
```

6.46 edu.ou.asgbook.histogram.HistogramBinSelection Class Reference

Tries out different values for the number of bins and replaces each pixel value by the center of its bin.

Static Public Member Functions

- static LatLonGrid band (LatLonGrid data, Histogram hist)

  replaces each pixel by the center of its bin

- static Histogram createBasedOnRange (LatLonGrid data)

  Based on range.

- static Histogram createHighestResolution (LatLonGrid data)

  Highest resolution possible.

- static Histogram createBasedOnStdDev (LatLonGrid data)

  Based on range.

- static Histogram createBasedOnNumSamples (LatLonGrid data)

  Based on range.

- static void main (String[ ] args) throws Exception

6.46.1 Detailed Description

Tries out different values for the number of bins and replaces each pixel value by the center of its bin.

Author:

valliappa.lakshmanan

6.46.2 Member Function Documentation

6.46.2.1 static LatLonGrid edu.ou.asgbook.histogram.HistogramBinSelection.band (LatLonGrid *data*, Histogram *hist*) `[static]`

replaces each pixel by the center of its bin

```
                                                                 {
        LatLonGrid result = LatLonGrid.copyOf(data);
        int nrows = result.getNumLat();
        int ncols = result.getNumLon();
        for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
            int bin_no = hist.getBinNumber(data.getValue(i,j), data.getMissing());
            int cval = hist.getCenterValue(bin_no, data.getMissing());
            result.setValue(i,j, cval);
        }
        return result;
    }
```

6.46.2.2 static Histogram edu.ou.asgbook.histogram.HistogramBinSelection.createBasedOnNumSamples (LatLonGrid *data*) `[static]`

Based on range.

```
                                                                   {
        int min = data.getMissing();
        int max = data.getMissing();
        int N = 0;
        int nrows = data.getNumLat();
        int ncols = data.getNumLon();
        for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
            int val = data.getValue(i,j);
            if ( val != data.getMissing() ){
                ++N;
                if (min == data.getMissing() || val < min){
                    min = val;
                }
                if (max == data.getMissing() || val > max){
                    max = val;
                }
            }
        }
        int nbins = 1 + (int) Math.round(Math.sqrt(N));
        System.out.println("Based on N="+ N + ", nbins=" + nbins);
        int incr = (max-min)/nbins;
        if (incr == 0) incr = 1;
        Histogram hist = new Histogram(min,incr,nbins);
        hist.update(data);
        return hist;
    }
```

6.46.2.3 static Histogram edu.ou.asgbook.histogram.HistogramBinSelection.createBasedOnRange (LatLonGrid *data*) `[static]`

Based on range.

```
                                                                {
        // find the range
        int min = data.getMissing();
        int max = data.getMissing();
        int nrows = data.getNumLat();
        int ncols = data.getNumLon();
        for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
            int val = data.getValue(i,j);
            if ( val != data.getMissing() ){
                if (min == data.getMissing() || val < min){
                    min = val;
                }
                if (max == data.getMissing() || val > max){
                    max = val;
                }
            }
        }
        int nbins = 1 + (int) Math.round(Math.log(max-min)/Math.log(2));
        System.out.println("Based on range: min=" + min + " max="+ max + ", nbins=" + nbins);
        int incr = (max-min)/nbins;
        if (incr == 0) incr = 1;
        Histogram hist = new Histogram(min,incr,nbins);
        hist.update(data);
        return hist;
    }
```

6.46.2.4 static Histogram edu.ou.asgbook.histogram.Histogram-BinSelection.createBasedOnStdDev (LatLonGrid *data*) [static]

Based on range.

```
                                                                {
        ScalarStatistic stat = new ScalarStatistic();
        int min = data.getMissing();
        int max = data.getMissing();
        int nrows = data.getNumLat();
        int ncols = data.getNumLon();
        for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
            int val = data.getValue(i,j);
            if ( val != data.getMissing() ){
                stat.update(val);
                if (min == data.getMissing() || val < min){
                    min = val;
                }
                if (max == data.getMissing() || val > max){
                    max = val;
                }
            }
        }
        double sigma = stat.getStdDeviation();
        int N = stat.getNumSamples();
        int nbins = 1 + (int) Math.round(3.5*sigma/Math.pow(N, 1.0/3));
        System.out.println("Based on sigma=" + sigma + " N="+ N + ", nbins=" + nbins);
        int incr = (max-min)/nbins;
        if (incr == 0) incr = 1;
        Histogram hist = new Histogram(min,incr,nbins);
        hist.update(data);
        return hist;
    }
```

6.46.2.5 static Histogram edu.ou.asgbook.histogram.HistogramBinSelection.createHighestResolution (LatLonGrid *data*) `[static]`

Highest resolution possible.

```
                                                                     {
        // find the range
        int min = data.getMissing();
        int max = data.getMissing();
        int nrows = data.getNumLat();
        int ncols = data.getNumLon();
        for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
            int val = data.getValue(i,j);
            if ( val != data.getMissing() ){
                if (min == data.getMissing() || val < min){
                    min = val;
                }
                if (max == data.getMissing() || val > max){
                    max = val;
                }
            }
        }
        System.out.println("full resolution: min=" + min + " max="+ max + ", incr=1");
        final int incr = 1;
        int nbins = (max-min)+1;
        Histogram hist = new Histogram(min,incr,nbins);
        hist.update(data);
        return hist;
    }
```

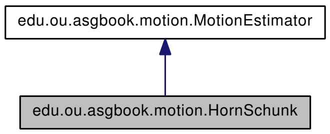
6.46.2.6 static void edu.ou.asgbook.histogram.HistogramBinSelection.main (String[ ] *args*) throws Exception `[static]`

```
                                                             {
        // create output directory
        File outdir = OutputDirectory.getDefault("histbin");

        // read input
        LatLonGrid conus = SurfaceAlbedo.read(SurfaceAlbedo.CONUS, 100);

        // find histogram in three different ways
        Map<String, Histogram> map = new HashMap<String,Histogram>();
        map.put("range", HistogramBinSelection.createBasedOnRange(conus));
        map.put("numsamples", HistogramBinSelection.createBasedOnNumSamples(conus));
        map.put("stddev", HistogramBinSelection.createBasedOnStdDev(conus));

        for (Map.Entry<String, Histogram> entry : map.entrySet()) {
            Histogram hist = entry.getValue();
            String name = entry.getKey();

            String filename = outdir.getAbsolutePath() + "/hist_" + name
                    + ".txt";
            PrintWriter writer = new PrintWriter(new FileWriter(filename));
            writer.println(hist);
            writer.close();
            System.out.println("Wrote to " + filename);

            LatLonGrid banded = HistogramBinSelection.band(conus, hist);
            KmlWriter.write(banded, outdir, name, PngWriter.createCoolToWarmColormap());
        }
    }
```

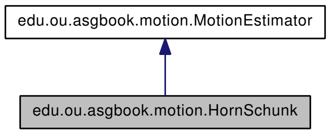
6.47 edu.ou.asgbook.motion.HornSchunk Class Reference

Horn-Schunk optical flow method of motion estimation.

Inheritance diagram for edu.ou.asgbook.motion.HornSchunk:


edu.ou.asgbook.motion.MotionEstimator
edu.ou.asgbook.motion.HornSchunk

Collaboration diagram for edu.ou.asgbook.motion.HornSchunk:


edu.ou.asgbook.motion.MotionEstimator
edu.ou.asgbook.motion.HornSchunk

Public Member Functions

- Override Pair< LatLonGrid, LatLonGrid > compute (LatLonGrid data0, LatLonGrid data1, File outdir)

  returns motion in the two directions.

Static Public Member Functions

- static void main (String[ ] args) throws Exception

Static Public Attributes

- static final int MOT_SCALE = 10

6.47.1 Detailed Description

Horn-Schunk optical flow method of motion estimation.

Author:

v.lakshmanan

6.47.2 Member Function Documentation

6.47.2.1 Override Pair<LatLonGrid, LatLonGrid> edu.ou.asgbook.motion.HornSchunk.compute (LatLonGrid *data0*, LatLonGrid *data1*, File *outdir*)

returns motion in the two directions.

The first one is north to south and the second one is east to west. The data is aligned to second time frame. The output dir is used for intermediate products and may be null.

Implements edu.ou.asgbook.motion.MotionEstimator.

```
                                                                                        {
        // Grids we need. initialize all of them at zero
        final int nrows = data1.getNumLat();
        final int ncols = data1.getNumLon();
        LatLonGrid I_x = new LatLonGrid(nrows, ncols, 0, data1.getNwCorner(), data1.getLatRes(), data1.
        LatLonGrid I_y = LatLonGrid.copyOf(I_x);
        LatLonGrid I_t = LatLonGrid.copyOf(I_x);
        LatLonGrid u = LatLonGrid.copyOf(I_x);
        LatLonGrid v = LatLonGrid.copyOf(I_x);

        // compute gradient of intensity in x, y and t directions
        for (int i=1; i < nrows-1; ++i) for (int j=1; j < ncols-1; ++j){
            int i_t = data1.getValue(i,j) - data0.getValue(i,j);   // time
            int i_x = data1.getValue(i,j) - data1.getValue(i-1,j); // lat
            int i_y = data1.getValue(i,j) - data1.getValue(i,j-1); // lon
            I_x.setValue(i,j, i_x);
            I_y.setValue(i,j, i_y);
            I_t.setValue(i,j, i_t);
        }

        // write intermediates
        if (outdir != null){
            try {
                KmlWriter.write(I_x, outdir, "I_x", PngWriter.createCoolToWarmColormap());
                KmlWriter.write(I_y, outdir, "I_y", PngWriter.createCoolToWarmColormap());
                KmlWriter.write(I_t, outdir, "I_t", PngWriter.createCoolToWarmColormap());
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

        // now iterate
        for (int iter=0; iter < MAX_ITER; ++iter){
            // compute meanu, meanv
            LatLonGrid meanu, meanv;
            if ( iter == 0 ){
                meanu = LatLonGrid.copyOf(u);
                meanv = LatLonGrid.copyOf(v);
            } else {
                ConvolutionFilter boxcar = new ConvolutionFilter(ConvolutionFilter.boxcar(2*SM_HALFSIZE
                meanu = boxcar.smooth(u);
                meanv = boxcar.smooth(v);
            }

            for (int i=1; i < nrows-1; ++i) for (int j=1; j < ncols-1; ++j){
                double u_k = meanu.getValue(i, j)/(double)MOT_SCALE;
                double v_k = meanv.getValue(i, j)/(double)MOT_SCALE;
                int i_x = I_x.getValue(i,j);
                int i_y = I_y.getValue(i,j);
                int i_t = I_t.getValue(i,j);
                double corr = (i_x*u_k + i_y*v_k + i_t) / (ALPHASQ + i_x*i_x + i_y*i_y);
                u.setValue(i,j, (int) Math.round((u_k - i_x*corr)*MOT_SCALE));
                v.setValue(i,j, (int) Math.round((v_k - i_y*corr)*MOT_SCALE));
            }

            if (outdir != null && iter == 0 || iter == 1 || iter == MAX_ITER/2){
                try {
                    KmlWriter.write(u, outdir, "motionNS_"+iter, PngWriter.createCoolToWarmC
                    KmlWriter.write(v, outdir, "motionEW_"+iter, PngWriter.createCoolToWarmC
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        }

        return new Pair<LatLonGrid,LatLonGrid>(u,v);
    }
```

6.47.2.2 static void edu.ou.asgbook.motion.HornSchunk.main (String[ ] *args*) throws Exception [static]

```
                                                            {
    // create output directory
    File out = OutputDirectory.getDefault("hornschunk");

    // read
    File f = new File("data/seviri");
    Pair<LatLonGrid,Date>[] grids = SeviriInfraredTemperature.readAll(f);

    // do alg
    MotionEstimator alg = new HornSchunk();
    Pair<LatLonGrid,LatLonGrid> motion = alg.compute(grids[0].first, grids[1].first, ou

    // write
    KmlWriter.write(motion.first, out, "opticflow_u", PngWriter.createCoolToWarmColorma
    KmlWriter.write(motion.second, out, "opticflow_v", PngWriter.createCoolToWarmColorm

    // align and compute difference
    LatLonGrid diff = new AlignAndDifference().compute(grids[0].first, grids[1].first,
    KmlWriter.write(diff, out, "opticflow_diff", PngWriter.createCoolToWarmColormap());
}
```

6.47.3 Member Data Documentation

6.47.3.1 final int edu.ou.asgbook.motion.HornSchunk.MOT_SCALE = 10 [static]

6.48 edu.ou.asgbook.transforms.HoughTransform Class Reference

Finds lines in image.

Public Member Functions

- Line[ ] findLines (LatLonGrid grid, int datathresh)

  Find best lines that connect points > thresh.

Static Public Member Functions

- static void main (String[ ] args) throws Exception

Classes

- class **Line**

6.48.1 Detailed Description

Finds lines in image.

Author:

v.lakshmanan

6.48.2 Member Function Documentation

6.48.2.1 Line [ ] edu.ou.asgbook.transforms.HoughTransform.findLines (LatLonGrid *grid*, int *datathresh*)

Find best lines that connect points > thresh.

Parameters:

grid

datathresh

Returns:

```
                                                             {
        int maxr = grid.getNumLat() + grid.getNumLon();
        int numr = (int) Math.round(maxr / DELTA_RHO);
        int numtheta = (int) Math.round(360 / DELTA_THETA);

        // update vote
        Line[] lines = new Line[numr * numtheta];
        for (int i=0; i < lines.length; ++i){
            lines[i] = new Line();
        }
        for (int i = 0; i < grid.getNumLat(); ++i) {
            for (int j = 0; j < grid.getNumLon(); ++j) {
                if (grid.getValue(i, j) > datathresh) {
                    // use this point to cast votes ...
                    for (int theta = 0; theta < numtheta; ++theta) {
                        double theta_radians = (theta * DELTA_THETA * Math.PI) / 180.0;
                        double rho = i * Math.cos(theta_radians) + j
                                * Math.sin(theta_radians);
                        int r = (int) Math.round(rho / DELTA_RHO);
                        if (r >= 0 && r < maxr) {
                            Line line = lines[r * numtheta + theta];
                            line.rho = rho;
                            line.theta = theta_radians;
                            line.numVotes++;
                            line.x1 = Math.min(line.x1, i);
                            line.x2 = Math.max(line.x2, i);
                            line.y1 = Math.min(line.y1, j);
                            line.y2 = Math.max(line.y2, j);
                        }
                    }
                }
            }
        }

        // sort the lines by vote
        Arrays.sort(lines);
        return lines;
    }
```

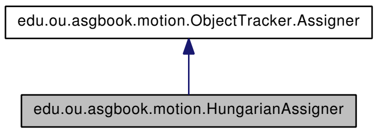
6.48.2.2 static void edu.ou.asgbook.transforms.HoughTransform.main (String[ ] *args*) throws Exception `[static]`

```
                                                             {
        // create output directory
        File out = OutputDirectory.getDefault("hough");

        // read input
        LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new GlobalPopulat
        KmlWriter.write(popdensity, out, "orig", PngWriter.createCoolToWarmColormap());

        // fill in
        popdensity = new DilateErodeFilter(2,3).filter(popdensity);
        popdensity = new ErosionFilter(3).filter(popdensity);
        KmlWriter.write(popdensity, out, "filledin", PngWriter.createCoolToWarmColormap());

        // skeletonize
        LatLonGrid skel = HilditchSkeletonization.findSkeleton(popdensity, 300, out);
        KmlWriter.write(skel, out, "skel", PngWriter.createCoolToWarmColormap());

        // find lines
        HoughTransform hough = new HoughTransform();
        HoughTransform.Line[] lines = hough.findLines(skel, 0);
        final int NBEST = 3;
        for (int i=0; i < Math.min(lines.length,NBEST); ++i){ // NBEST lines
            HoughTransform.Line line = lines[i];
            System.out.println(line);
            List<Pixel> pixels = line.computePixels(popdensity.getNumLat(), popdensity.get
            for (Pixel p : pixels){
                popdensity.setValue(p.getX(), p.getY(), 1000);
            }
        }
        KmlWriter.write(popdensity, out, "lines",
                PngWriter.createCoolToWarmColormap());
    }
```

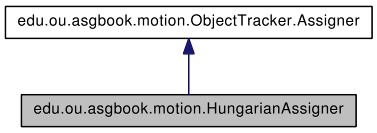
6.49 edu.ou.asgbook.motion.HungarianAssigner Class Reference

Optimal assignment algorithm.

Inheritance diagram for edu.ou.asgbook.motion.HungarianAssigner:


edu.ou.asgbook.motion.ObjectTracker.Assigner
edu.ou.asgbook.motion.HungarianAssigner

Collaboration diagram for edu.ou.asgbook.motion.HungarianAssigner:


edu.ou.asgbook.motion.ObjectTracker.Assigner
edu.ou.asgbook.motion.HungarianAssigner

Public Member Functions

- Override int[ ] getAssignments (int[ ][ ] cost, int maxcost)

Static Public Member Functions

- static void main (String[ ] args) throws Exception

Classes

- class **HungarianMatch**

6.49.1 Detailed Description

Optimal assignment algorithm.

Author:

valliappa.lakshmanan

6.49.2 Member Function Documentation

6.49.2.1 Override int [ ] edu.ou.asgbook.motion.HungarianAssigner.getAssignments (int *cost*[ ][ ], int *maxcost*)

Implements edu.ou.asgbook.motion.ObjectTracker.Assigner.

```
                                                         {
        // intialize result to be all unassigned
        int[] result = new int[cost.length];
        for (int i=0; i < result.length; ++i){
            result[i] = -1;
        }

        // if number of objects is zero, then can’t do any assignment
        if (cost.length == 0 || cost[0].length == 0){
            return result;
        }

        if (cost[0].length < cost.length){
            // rotate so that we have more columns than rows
            int[][] rot = new int[ cost[0].length ][ cost.length ];
            for (int i=0; i < cost.length; ++i){
                for (int j=0; j < cost[i].length; ++j){
                    rot[j][i] = cost[i][j];
                }
            }
            // do the assignment process on rotated cost function
            int[] col_to_row = getAssignments(rot, maxcost);
            // fix result: we need row_to_col
            for (int col = 0; col < col_to_row.length; ++col){
                int row = col_to_row[col];
                if (row >= 0){
                    result[row] = col;
                }
            }
            return result;
        }

        // threshold just in case some cost > maxcost
        for (int i=0; i < cost.length; ++i){
            for (int j=0; j < cost[i].length; ++j){
                if (cost[i][j] > maxcost){
                    cost[i][j] = maxcost;
                }
            }
        }
        HungarianMatch match = new HungarianMatch(cost);
        match.do_step1();
        match.do_step2();
        match.do_step3();
        for (int i=0; i < cost.length; ++i){
            for (int j=0; j < cost[i].length; ++j){
                if (match.starred_zero[i][j] && cost[i][j] < maxcost ){
                    result[i] = j;
                }
            }
        }
        return result;
    }
```

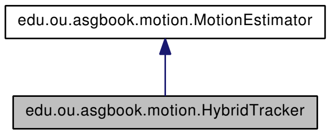
6.49.2.2 static void edu.ou.asgbook.motion.HungarianAssigner.main (String[ ] *args*) throws Exception [static]

```
                                                                {
        // create output directory
        File out = OutputDirectory.getDefault("hungarian");

        // read
        File f = new File("data/seviri");
        Pair<LatLonGrid,Date>[] grids = SeviriInfraredTemperature.readAll(f);

        // do alg
        Segmenter seg = new ObjectTracker.SimpleSegmenter(100, 110, 1000);
        ObjectTracker alg = new ObjectTracker(seg, new ObjectTracker.CentroidDistance(), new Hungariar
        MedianFilter smoother = new MedianFilter(10);
        LatLonGrid grid0 = smoother.filter(grids[0].first);
        LatLonGrid grid1 = smoother.filter(grids[1].first);
        Pair<LatLonGrid,LatLonGrid> motion = alg.compute(grid0, grid1, out);

        // write
        SaturateFilter filter = new SaturateFilter(-150, 150);
        LatLonGrid u = filter.filter(motion.first);
        LatLonGrid v = filter.filter(motion.second);
        KmlWriter.write(u, out, "hungarian_u", PngWriter.createCoolToWarmColormap());
        KmlWriter.write(v, out, "hungarian_v", PngWriter.createCoolToWarmColormap());
    }
```

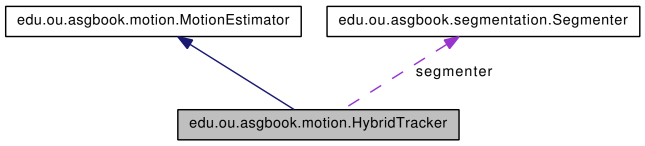
6.50 edu.ou.asgbook.motion.HybridTracker Class Reference

Estimates motion by finding cross-correlation of objects in one frame to the pixels in the previous frame.

Inheritance diagram for edu.ou.asgbook.motion.HybridTracker:


edu.ou.asgbook.motion.MotionEstimator
edu.ou.asgbook.motion.HybridTracker

Collaboration diagram for edu.ou.asgbook.motion.HybridTracker:


edu.ou.asgbook.motion.MotionEstimator
edu.ou.asgbook.segmentation.Segmenter
segmenter
edu.ou.asgbook.motion.HybridTracker

Public Member Functions

- HybridTracker (Segmenter seg, int maxmotionx, int maxmotiony)
- Override Pair< LatLonGrid, LatLonGrid > compute (LatLonGrid data0, LatLonGrid data1, File outdir)

  returns motion in the two directions.

Static Public Member Functions

- static void main (String[ ] args) throws Exception

Classes

- class **Centroid**

6.50.1 Detailed Description

Estimates motion by finding cross-correlation of objects in one frame to the pixels in the previous frame.

Author:

v.lakshmanan

6.50.2 Constructor & Destructor Documentation

6.50.2.1 edu.ou.asgbook.motion.HybridTracker.HybridTracker (Segmenter *seg*, int *maxmotionx*, int *maxmotiony*)

```
                                                                              {
        MAX_U = maxmotionx;
        MAX_V = maxmotiony;
        segmenter = seg;
    }
```

6.50.3 Member Function Documentation

6.50.3.1 Override Pair<LatLonGrid, LatLonGrid> edu.ou.asgbook.motion.HybridTracker.compute (LatLonGrid *data0*, LatLonGrid *data1*, File *outdir*)

returns motion in the two directions.

The first one is north to south and the second one is east to west. The data is aligned to second time frame. The output dir is used for intermediate products and may be null.

Implements edu.ou.asgbook.motion.MotionEstimator.

```
                                                                                                                      {
        LabelResult objects1 = segmenter.label(data1);
        if (outdir != null){
            try {
                KmlWriter.write(objects1.label, outdir, "hybobjects1", PngWriter.createRandomColormap()
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

        // find motion for each region and apply it to all pixels for that region
        Pixel[][] regions = RegionProperty.getPixelsInRegions(data1, objects1);
        LatLonGrid u = new LatLonGrid(data0.getNumLat(), data0.getNumLon(), 0, data0.getNwCorner(), dat
        LatLonGrid v = LatLonGrid.copyOf(u);
        RegionProperty[] regprop = RegionProperty.compute(objects1, data1);
        List<Centroid> centroids = new ArrayList<Centroid>();
        for (int reg=1; reg < regions.length; ++reg){
            Pair<Integer,Integer> motion = computeMotion(regions[reg], data0);
            int motx = motion.first;
            int moty = motion.second;
            Centroid c = new Centroid();
            c.cx = regprop[reg].getCx();
            c.cy = regprop[reg].getCy();
            c.motx = motx;
            c.moty = moty;
            c.size = regprop[reg].getSize();
            centroids.add(c);
            for (Pixel p : regions[reg]){
                u.setValue(p.getX(), p.getY(), motx);
                v.setValue(p.getX(), p.getY(), moty);
            }
        }

        if (outdir != null){
            try {
                KmlWriter.write(u, outdir, "u_beforeinterp", PngWriter.createCoolToWarmColor
                KmlWriter.write(v, outdir, "v_beforeinterp", PngWriter.createCoolToWarmColor
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

        // interpolate inbetween regions if you have enough of them ...
        if ( centroids.size() > 1 ){
            LatLonGrid interpu = LatLonGrid.copyOf(u);
            LatLonGrid interpv = LatLonGrid.copyOf(v);
            for (int i=0; i < interpu.getNumLat(); ++i) for (int j=0; j < interpu.getNumLon
                double totu = 0;
                 double totv = 0;
                 double totwt = 0;
                 for (Centroid c : centroids){
                     double distx = c.cx - i;
                     double disty = c.cy - j;
                     double distsq = distx*distx + disty*disty;
                     double wt = c.size * 1.0/(distsq*distsq + 0.0001); // 1/r^2
                     totu += c.motx * wt;
                     totv += c.moty * wt;
                     totwt += wt;
                 }
                 interpu.setValue(i, j, (int) Math.round(totu/totwt));
                 interpv.setValue(i, j, (int) Math.round(totv/totwt));
             }

             if (outdir != null){
                 try {
                     KmlWriter.write(interpu, outdir, "u_interp", PngWriter.createCoolToWarm
                     KmlWriter.write(interpv, outdir, "v_interp", PngWriter.createCoolToWarm
                 } catch (Exception e) {
                     e.printStackTrace();
                 }
             }

             return new Pair<LatLonGrid,LatLonGrid>(interpu,interpv);
         }

        return new Pair<LatLonGrid,LatLonGrid>(u,v);
    }
```

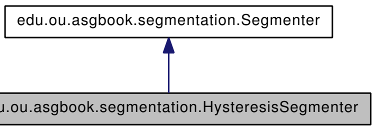
6.50.3.2 static void edu.ou.asgbook.motion.HybridTracker.main (String[ ] *args*) throws Exception [static]

```
                                                          {
        // create output directory
        File out = OutputDirectory.getDefault("hybridtracker");

        // read
        File f = new File("data/seviri");
        Pair<LatLonGrid,Date>[] grids = SeviriInfraredTemperature.readAll(f);

        // do alg
        Segmenter seg = new ObjectTracker.SimpleSegmenter(100, 110, 100);
        MotionEstimator alg = new HybridTracker( seg, 20, 20 );
        MedianFilter smoother = new MedianFilter(10);
        LatLonGrid grid0 = smoother.filter(grids[0].first);
        LatLonGrid grid1 = smoother.filter(grids[1].first);
        Pair<LatLonGrid,LatLonGrid> motion = alg.compute(grid0, grid1, out);

        LatLonGrid diff = new AlignAndDifference().compute(grids[0].first, grids[1].first, motion);
        KmlWriter.write(diff, out, "hybriddiff", PngWriter.createCoolToWarmColormap());

        // write
        SaturateFilter filter = new SaturateFilter(-15, 15);
        LatLonGrid u = filter.filter(motion.first);
        LatLonGrid v = filter.filter(motion.second);
        KmlWriter.write(u, out, "opticflow_u", PngWriter.createCoolToWarmColormap());
        KmlWriter.write(v, out, "opticflow_v", PngWriter.createCoolToWarmColormap());
    }
```

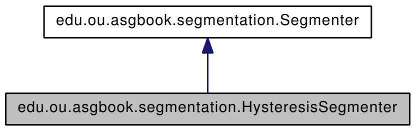
6.51 edu.ou.asgbook.segmentation.Hysteresis-Segmenter Class Reference

Objects consist of pixels that are > thresh2 but have at least one pixel > thresh1.

Inheritance diagram for edu.ou.asgbook.segmentation.HysteresisSegmenter:


edu.ou.asgbook.segmentation.Segmenter
edu.ou.asgbook.segmentation.HysteresisSegmenter

Collaboration diagram for edu.ou.asgbook.segmentation.HysteresisSegmenter:


edu.ou.asgbook.segmentation.Segmenter
edu.ou.asgbook.segmentation.HysteresisSegmenter

Public Member Functions

- HysteresisSegmenter (int thresh1, int thresh2)
- Override LabelResult label (LatLonGrid data)

  Creates a labeled grid where background pixels are set to 0 and labels for objects go 1,2,3.

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.51.1 Detailed Description

Objects consist of pixels that are > thresh2 but have at least one pixel > thresh1.

Author:

v.lakshmanan

6.51.2 Constructor & Destructor Documentation

6.51.2.1 edu.ou.asgbook.segmentation.HysteresisSegmenter.Hysteresis-Segmenter (int *thresh1*, int *thresh2*)

```
                                                          {
        super();
        this.t1 = thresh1;
        this.t2 = thresh2;
        if (t1 < t2){
            // swap
            int t = t1;
            t1 = t2;
            t2 = t;
        }
    }
```

6.51.3 Member Function Documentation

6.51.3.1 Override LabelResult edu.ou.asgbook.segmentation.Hysteresis-Segmenter.label (LatLonGrid *data*) [virtual]

Creates a labeled grid where background pixels are set to 0 and labels for objects go 1,2,3.

.. All pixels > thresh are part of an object.

Implements edu.ou.asgbook.segmentation.Segmenter.

```
                                              {
        final int UNSET = 0;
        int nrows = data.getNumLat();
        int ncols = data.getNumLon();
        LatLonGrid label = new LatLonGrid(nrows,ncols,0,data.getNwCorner(),data.getLatRes(),data.getLo
        // label.fill(UNSET); java default is to zero-out arrays
        int regno = 0;
        for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
            if ( data.getValue(i, j) > t1 && label.getValue(i, j) == UNSET ){
                ++regno;
                RegionGrowing.growRegion(i,j, data, t2, label, regno);
            }
        }
        System.out.println("Found " + (regno+1) + " objects");
        return new LabelResult(label, regno);
    }
```

6.51.3.2 static void edu.ou.asgbook.segmentation.HysteresisSegmenter.main (String[ ] *args*) throws Exception [static]

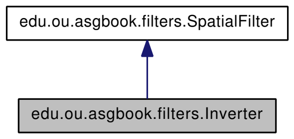
```
                                                              {
```

```
        File out = OutputDirectory.getDefault("hysteresis");

        // data
        LatLonGrid grid = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new GlobalPop
        KmlWriter.write(grid, out, "orig", PngWriter.createCoolToWarmColormap());

        // hysteresis thresh
        for (int thresh = 10; thresh <= 30; thresh += 10){
            int t1 = thresh;
            int t2 = thresh-5;
            Segmenter seg = new HysteresisSegmenter(t1, t2);
            LatLonGrid label = seg.label(grid).label;
            // label.setMissing(-1); // so background is present
            KmlWriter.write(label, out, "cities_"+t1+"_"+t2, PngWriter.createRandomColormap
        }
    }
```

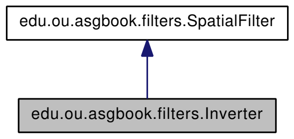
6.52 edu.ou.asgbook.filters.Inverter Class Reference

at every pixel, replaces its value (val) by (A - val)

Inheritance diagram for edu.ou.asgbook.filters.Inverter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.Inverter

Collaboration diagram for edu.ou.asgbook.filters.Inverter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.Inverter

Public Member Functions

- Inverter (int A)
- Override LatLonGrid filter (LatLonGrid input)
- LatLonGrid invert (final LatLonGrid input)

6.52.1 Detailed Description

at every pixel, replaces its value (val) by (A - val)

Author:

Valliappa.Lakshmanan

6.52.2 Constructor & Destructor Documentation

6.52.2.1 edu.ou.asgbook.filters.Inverter.Inverter (int *A*)

```
                            {
        this.A = A;
    }
```

6.52.3 Member Function Documentation

6.52.3.1 Override LatLonGrid edu.ou.asgbook.filters.Inverter.filter (LatLonGrid *input*)

```
                                             {
        return invert(input);
    }
```

6.52.3.2 LatLonGrid edu.ou.asgbook.filters.Inverter.invert (final LatLonGrid *input*)

```
                                                  {
        LatLonGrid output = LatLonGrid.copyOf(input);
        int[][] outData = output.getData();
        int[][] inData = input.getData();
        for (int i=0; i < output.getNumLat(); ++i){
            for (int j=0; j < output.getNumLon(); ++j){
                if ( inData[i][j] != input.getMissing() ){
                    outData[i][j] = A - inData[i][j];
                }
            }
        }
        return output;
    }
```

6.53 edu.ou.asgbook.motion.KalmanFilter Class Reference

For the time smoothing of motion vectors.

Public Member Functions

- KalmanFilter (double x_0, double dx_0)

  Start off with an initial estimate for the position and velocity.

- void init (double x_0, double dx_0)
- boolean updated ()
- void update (double z_k)
- double getValue ()

  get the smoothed centroid position

- double getRateOfChange ()

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.53.1 Detailed Description

For the time smoothing of motion vectors.

Author:

valliappa.lakshmanan

6.53.2 Constructor & Destructor Documentation

6.53.2.1 edu.ou.asgbook.motion.KalmanFilter.KalmanFilter (double *x_0*, double *dx_0*)

Start off with an initial estimate for the position and velocity.

```
                                                  {
        init(x_0, dx_0);
    }
```

6.53.3 Member Function Documentation

6.53.3.1 double edu.ou.asgbook.motion.KalmanFilter.getRateOfChange ()

```
                                    {
        return x_k.get(1,0);
    }
```

6.53.3.2 double edu.ou.asgbook.motion.KalmanFilter.getValue ()

get the smoothed centroid position

```
                             {
        return x_k.get(0, 0);
    }
```

6.53.3.3 void edu.ou.asgbook.motion.KalmanFilter.init (double *x_0*, double *dx_0*)

```
                                                 {
        k = 0;
        // x_k
        x_k = new Matrix(2,1);
        x_k.set(0,0, x_0);
        x_k.set(1,0, dx_0);

        // p_k
        p_k = new Matrix( 2, 2 ); // all zero

        // assume unit white noise for errors before we see any observations.
        R_k = 1;
        Q_k = Matrix.identity(2,2);
    }
```

6.53.3.4 static void edu.ou.asgbook.motion.KalmanFilter.main (String[ ] *args*) throws Exception [static]

```
                                                               {
        double[] truex = new double[20];
        double[] trueu = new double[truex.length];
        double[] obsx = new double[truex.length];
        truex[0] = 5;
        trueu[0] = 3;
        obsx[0]= truex[0] + noise();

        KalmanFilter kalman = new KalmanFilter(obsx[0], trueu[0]); // assume that we have a
        double true_acc = 0.2;

        System.out.println("true x & true velocity & observed x & estimate of x & estimate of velocity
        for (int i=1; i < truex.length; ++i){
            trueu[i] = trueu[i-1] + true_acc;
            truex[i] = truex[i-1] + trueu[i-1];
            obsx[i] = noise() + truex[i];
            kalman.update(obsx[i]);
            System.out.println( df(truex[i]) + " & " + df(trueu[i]) + " & " + df(obsx[i]) + " & " + df
        }



    }
```

6.53.3.5 void edu.ou.asgbook.motion.KalmanFilter.update (double *z_k*)

```
                                   {
        ++k; // observation number ...
        if ( MAX_HISTORY > 0 && k > MAX_HISTORY ){
            k = MAX_HISTORY; // k is used in computing Q_k and R_k
        }

        // P_k+1 and x_k+1 will be computed on next turn around so that getValue()
        // works correctly ...
        p_k = phi.copy().times(p_k).times(phiT).plus(Q_k);
        x_k = phi.copy().times(x_k);


        // Kalman gain
        double inv = H.copy().times(p_k).times(HT).get(0,0) + R_k;
        final Matrix K_k = p_k.copy().times(HT).times( 1.0 / inv );

        // observation error
        final double v_k = z_k - H.copy().times(x_k).get(0,0);

        // update x_k
        final Matrix update = K_k.copy().times(v_k);
        x_k = x_k.plus( update );

        // estimate R_k, covariance of observation error to use next time 'round
        R_k = ( (k-1) * R_k + v_k * v_k ) / k;

        // estimate Q_k, covariance of model error to use in P_k+1 computation
        if ( k != 1 ){ // when k is 1, x_k=old_x_k and so Q_k would become 0
            final Matrix wkT = update.copy().transpose();
            final Matrix wk_wkT = update.copy().times(wkT);
            Q_k = Q_k.times(k-1).plus(wk_wkT).times(1.0/k);
        }

        // update error covariance for updated estimate
        p_k = Matrix.identity(2,2).minus(K_k.copy().times(H)).times(p_k);

        if ( finite(getValue()) == false || finite(getRateOfChange()) == false ){
            double new_val = getValue();
            if ( finite(new_val) == false ) new_val = 0;
            double new_rate = getRateOfChange();
            if ( finite(new_rate) == false ) new_rate = 0;
            init( new_val, new_rate );
        }
    }
```

6.53.3.6 boolean edu.ou.asgbook.motion.KalmanFilter.updated ()

```
                            {
        return ( k > 0 );
    }
```

6.54 edu.ou.asgbook.io.KmlWriter Class Reference

Writes data out in KML form, for display in Google Earth or similar program.

Static Public Member Functions

- static void write (LatLonGrid grid, File outputDir, String dataName, ColorModel colormap) throws Exception
- static void write (List< LatLon > points, File outputDir, String dataName) throws Exception
- static void write (List< LatLon > points, List< String > names, File outputDir, String dataName) throws Exception
- static void debugWrite (LatLonGrid grid, File out, String name)
- static void main (String[ ] args) throws Exception

6.54.1 Detailed Description

Writes data out in KML form, for display in Google Earth or similar program.

Author:

Valliappa.Lakshmanan

6.54.2 Member Function Documentation

6.54.2.1 static void edu.ou.asgbook.io.KmlWriter.debugWrite (LatLonGrid *grid*, File *out*, String *name*) [static]

```
                                                                        {
            if (out != null){
                try {
                    KmlWriter.write(grid, out, name, PngWriter.createCoolToWarmColormap());
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        }
```

6.54.2.2 static void edu.ou.asgbook.io.KmlWriter.main (String[ ] *args*) throws Exception [static]

```
                                                                 {
            LatLonGrid grid = new LatLonGrid(100, 200, -1, new LatLon(35,-97), 0.1, 0.1);
            for (int i=0; i < grid.getNumLat(); ++i){
```

```
            for (int j=0; j < grid.getNumLon(); ++j){
                grid.getData()[i][j] = i + j;
                if ( i%10 == 0 || j%20 == 0){
                    grid.getData()[i][j] = grid.getMissing();
                }
            }
        }
        File outputDir = OutputDirectory.getDefault("kmlwriter");
        KmlWriter.write(grid, outputDir, "kmlwriter", PngWriter.createHotColormap());
    }
```

6.54.2.3 static void edu.ou.asgbook.io.KmlWriter.write (List< LatLon > *points*, List< String > *names*, File *outputDir*, String *dataName*) throws Exception `[static]`

```

       // create KML doc
       Document doc = DocumentBuilderFactory.newInstance().newDocumentBuilder().newDocument
       Element root = doc.createElement("kml");
       doc.appendChild(root);
       Element docE = doc.createElement("Document");
       root.appendChild(docE);
       for (int i=0; i < points.size(); ++i){
           Element placemark = doc.createElement("Placemark");
           docE.appendChild(placemark);
           Element name = doc.createElement("name");
           placemark.appendChild(name);
            if (names != null && i < names.size()){
                name.setTextContent(names.get(i));
            } else {
                name.setTextContent(dataName + "#" + (i+1));
            }
            Element point = doc.createElement("Point");
            placemark.appendChild(point);
            Element coords = doc.createElement("coordinates");
            point.appendChild(coords);
            coords.setTextContent(points.get(i).getLon() + "," + points.get(i).getLat() +
        }

        // write out
        File filename = new File(outputDir.getAbsolutePath() + "/" + dataName + ".kml");
        Transformer t = TransformerFactory.newInstance().newTransformer();
        t.setOutputProperty(OutputKeys.INDENT, "yes");
        t.setOutputProperty("{http://xml.apache.org/xslt}indent-amount", "2");
        t.transform(new DOMSource(doc), new StreamResult(filename));
        System.out.println("Wrote " + filename + " to refer to " + names.size() + " placema
    }
```

6.54.2.4 static void edu.ou.asgbook.io.KmlWriter.write (List< LatLon > *points*, File *outputDir*, String *dataName*) throws Exception `[static]`

```

```

```
        List<String> names = new ArrayList<String>();
        for (int i=0; i < points.size(); ++i){
            names.add(dataName + " " + (i+1) );
        }
        write(points, names, outputDir, dataName);
    }
```

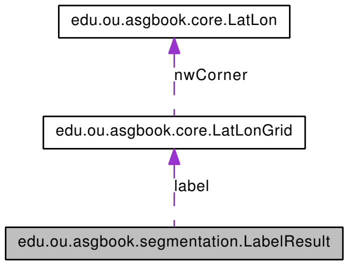
6.54.2.5 static void edu.ou.asgbook.io.KmlWriter.write (LatLonGrid *grid*, File *outputDir*, String *dataName*, ColorModel *colormap*) throws Exception [static]

```

        // write image
        File imgFileName = new File(outputDir.getAbsolutePath() + "/" + dataName + ".png");
        PngWriter.writeAutoScaled(grid, imgFileName, colormap);

        // create KML
        Document doc = DocumentBuilderFactory.newInstance().newDocumentBuilder().newDocument();
        Element root = doc.createElement("kml");
        doc.appendChild(root);
        Element folder = doc.createElement("Folder");
        root.appendChild(folder);
        Element folderName = doc.createElement("name");
        folder.appendChild(folderName);
        folderName.setTextContent(dataName);
        Element folderDesc = doc.createElement("description");
        folderDesc.setTextContent(dataName + " created by " + KmlWriter.class.getCanonicalName() + " fo
        Element goverlay = doc.createElement("GroundOverlay");
        folder.appendChild(goverlay);
        Element icon = doc.createElement("Icon");
        goverlay.appendChild(icon);
        Element href = doc.createElement("href");
        icon.appendChild(href);
        href.setTextContent(dataName + ".png");
        Element box = doc.createElement("LatLonBox");
        goverlay.appendChild(box);
        Element north = doc.createElement("north");
        north.setTextContent("" + grid.getNwCorner().getLat());
        box.appendChild(north);
        Element south = doc.createElement("south");
        south.setTextContent("" + grid.getSeCorner().getLat());
        box.appendChild(south);
        Element east = doc.createElement("east");
        east.setTextContent("" + grid.getSeCorner().getLon());
        box.appendChild(east);
        Element west = doc.createElement("west");
        west.setTextContent("" + grid.getNwCorner().getLon());
        box.appendChild(west);
        box.appendChild(north);

        // write KML
        File kmlFileName = new File(outputDir.getAbsolutePath() + "/" + dataName + ".kml");
        Transformer t = TransformerFactory.newInstance().newTransformer();
        t.setOutputProperty(OutputKeys.INDENT, "yes");
        t.setOutputProperty("{http://xml.apache.org/xslt}indent-amount", "2");
        t.transform(new DOMSource(doc), new StreamResult(kmlFileName));
        System.out.println("Wrote " + kmlFileName + " to refer to image");
    }
```

6.55 edu.ou.asgbook.segmentation.LabelResult Class Reference

Result of segmentation.

Collaboration diagram for edu.ou.asgbook.segmentation.LabelResult:


edu.ou.asgbook.core.LatLon
nwCorner
edu.ou.asgbook.core.LatLonGrid
label
edu.ou.asgbook.segmentation.LabelResult

Public Member Functions

- LabelResult (LatLonGrid label, int maxlabel)

Public Attributes

- final LatLonGrid label
- final int maxlabel

6.55.1 Detailed Description

Result of segmentation.

Each pixel holds the region number that it belongs to. Zero is the background value.

Author:

valliappa.lakshmanan

6.55.2 Constructor & Destructor Documentation

6.55.2.1 edu.ou.asgbook.segmentation.LabelResult.LabelResult (LatLonGrid *label*, int *maxlabel*)

```
                                                             {
        this.label = label;
        this.maxlabel = maxlabel;
    }
```

6.55.3 Member Data Documentation

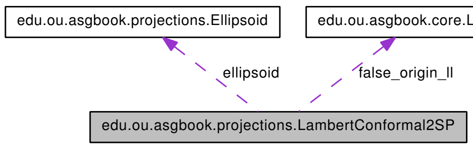
6.55.3.1 final LatLonGrid edu.ou.asgbook.segmentation.LabelResult.label

6.55.3.2 final int edu.ou.asgbook.segmentation.LabelResult.maxlabel

6.56 edu.ou.asgbook.projections.LambertConformal2SP Class Reference

Lambert Conformation 2 Standard Parallels map projection.

Collaboration diagram for edu.ou.asgbook.projections.LambertConformal2SP:


edu.ou.asgbook.projections.Ellipsoid
edu.ou.asgbook.core.LatLon
ellipsoid
false_origin_ll
edu.ou.asgbook.projections.LambertConformal2SP

Public Member Functions

- LambertConformal2SP (Ellipsoid ellipsoid, LatLon falseOriginLl, double lat_1, double lat_2, Coord falseOriginLam)
- Coord getLambert (LatLon in)
- LatLon getLatLon (Coord lam)

Static Public Member Functions

- static void main (String[ ] args)

Classes

- class **Coord**

6.56.1 Detailed Description

Lambert Conformation 2 Standard Parallels map projection.

Author:

v.lakshmanan

6.56.2 Constructor & Destructor Documentation

6.56.2.1 edu.ou.asgbook.projections.LambertConformal2SP.Lambert-Conformal2SP (Ellipsoid *ellipsoid*, LatLon *falseOriginLl*, double *lat_1*, double *lat_2*, Coord *falseOriginLam*)

```
                                                              {
        this.ellipsoid = ellipsoid;
        this.false_origin_ll = falseOriginLl;
        this.lat_1 = lat_1;
        this.lat_2 = lat_2;
        this.false_origin_lam = falseOriginLam;

        this.e = Math.sqrt(ellipsoid.eccsq);
        double phi1 = Math.toRadians(this.lat_1);
        double phi2 = Math.toRadians(this.lat_2);
        double t1 = compute_t(e,phi1);
        double t2 = compute_t(e,phi2);
        double m1 = compute_m(e,phi1);
        double m2 = compute_m(e,phi2);

        this.n = (Math.log(m1) - Math.log(m2))/(Math.log(t1) - Math.log(t2));
        this.F = m1 / (n*Math.pow(t1,n));

        double phiF = Math.toRadians(false_origin_ll.getLat());
        double tF = compute_t(e, phiF);
        this.rF = ellipsoid.eqr * F * Math.pow(tF, n);
    }
```

6.56.3 Member Function Documentation

6.56.3.1 Coord edu.ou.asgbook.projections.LambertConformal2SP.getLambert (LatLon *in*)

```
                                        {
        double phi = Math.toRadians(in.getLat());
        double t = compute_t(e,phi);
        double r = ellipsoid.eqr * F * Math.pow(t,n);
        double lambda = Math.toRadians(in.getLon());
        double lambdaF = Math.toRadians(false_origin_ll.getLon());
        double theta = n * (lambda - lambdaF);

        double easting = false_origin_lam.easting + r * Math.sin(theta);
        double northing = false_origin_lam.northing + rF - r * Math.cos(theta);
        return new Coord(northing, easting);
    }
```

6.56.3.2 LatLon edu.ou.asgbook.projections.LambertConformal2SP.getLatLon (Coord *lam*)

```
                                         {
```

```
      double eastdiff = (lam.easting - false_origin_lam.easting);
      double northdiff = (lam.northing - false_origin_lam.northing);
      double rFnorthdiff = rF - northdiff;
      double r = Math.sqrt( eastdiff*eastdiff + rFnorthdiff*rFnorthdiff );
      if ( n < 0 ) r = -r;
      double t = Math.pow( r / (ellipsoid.eqr*F) , 1/n );
      double theta = Math.atan( eastdiff/rFnorthdiff );

      double lon = Math.toDegrees(theta/n) + false_origin_ll.getLon();

      // iterate to find phi
      double phi = Math.PI/2 - 2 * Math.atan(t);
      double old_phi;
      int iter=0;
      do{
        old_phi = phi;
        ++iter;
        phi = Math.PI/2 - 2 * Math.atan( t*Math.pow( (1-e*Math.sin(phi))/(1+e*Math.sin(phi)), e/2 ) );
      } while ( Math.abs(phi-old_phi) > 0.00001 && iter < 5 );

      double lat = Math.toDegrees(phi);
      return new LatLon( lat, lon );
    }
```

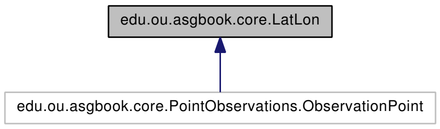
6.56.3.3 static void edu.ou.asgbook.projections.LambertConformal2SP.main (String[ ] *args*) [static]

```
                                   {
        LambertConformal2SP conv = new LambertConformal2SP(Ellipsoid.WGS84(), new LatLon(51,-127), 43.
        LatLon ll = new LatLon(36,-96);
        LambertConformal2SP.Coord lam = conv.getLambert(ll);
        LatLon ll2 = conv.getLatLon(lam);
        System.out.println(ll + "->" + lam + "->" + ll2);

        ll = new LatLon(51,-96);
        lam = conv.getLambert(ll);
        ll2 = conv.getLatLon(lam);
        System.out.println(ll + "->" + lam + "->" + ll2);

        ll = new LatLon(21,-96);
        lam = conv.getLambert(ll);
        ll2 = conv.getLatLon(lam);
        System.out.println(ll + "->" + lam + "->" + ll2);

        ll = new LatLon(36,-127);
        lam = conv.getLambert(ll);
        ll2 = conv.getLatLon(lam);
        System.out.println(ll + "->" + lam + "->" + ll2);

        ll = new LatLon(36,-65);
        lam = conv.getLambert(ll);
        ll2 = conv.getLatLon(lam);
        System.out.println(ll + "->" + lam + "->" + ll2);
    }
```

6.57 edu.ou.asgbook.core.LatLon Class Reference

A point on the earth's surface typically in WGS84.

Inheritance diagram for edu.ou.asgbook.core.LatLon:


edu.ou.asgbook.core.LatLon
edu.ou.asgbook.core.PointObservations.ObservationPoint

Public Member Functions

- double getLat ()
- double getLon ()
- LatLon (double lat, double lon)
- double distanceInKms (LatLon other)
- Override String toString ()

Static Public Member Functions

- static void main (String[ ] args)

6.57.1 Detailed Description

A point on the earth's surface typically in WGS84.

Author:

Valliappa.Lakshmanan

6.57.2 Constructor & Destructor Documentation

6.57.2.1 edu.ou.asgbook.core.LatLon.LatLon (double *lat*, double *lon*)

```
                                            {
        super();
        this.lat = lat;
        this.lon = lon;
    }
```

6.57.3 Member Function Documentation

6.57.3.1 double edu.ou.asgbook.core.LatLon.distanceInKms (LatLon *other*)

```
                                           {
        double lat1 = Math.toRadians(this.lat);
        double lat2 = Math.toRadians(other.lat);
        double lon1 = Math.toRadians(this.lon);
        double lon2 = Math.toRadians(other.lon);

        // double R = 6371; // spherical earth radius
        double lat0 = (lat2+lat1)/2;
        double a = 6378.137; // WGS-84
        double f = 1.0/298.257223563;
        double esq = f*(2-f);
        double R=a * (1-esq)/Math.pow(sq(1-esq*(Math.sin(lat0))),1.5);

        double dlon = lon2 - lon1;
        double dlat = lat2 - lat1;
        double term = sq(Math.sin(dlat/2)) + Math.cos(lat1) * Math.cos(lat2) * sq(Math.sin(dlon/2));
        return (2 * R * Math.asin(Math.min(1,Math.sqrt(term))));
    }
```

6.57.3.2 double edu.ou.asgbook.core.LatLon.getLat ()

```
                          {
        return lat;
    }
```

6.57.3.3 double edu.ou.asgbook.core.LatLon.getLon ()

```
                          {
        return lon;
    }
```

6.57.3.4 static void edu.ou.asgbook.core.LatLon.main (String[ ] *args*) [static]

```
                                        {
        LatLon pt1 = new LatLon(35,-97);
        LatLon pt2 = new LatLon(35.01, -97);
        LatLon pt3 = new LatLon(35, -97.01);
        System.out.println("sph: 0.01 in lat = " + pt1.distanceInKms(pt2) + " kms at " + pt1);
        System.out.println("sph: 0.01 in lon = " + pt1.distanceInKms(pt3) + " kms at " + pt1);
    }
```

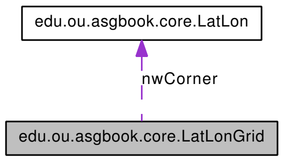
6.57.3.5 Override String edu.ou.asgbook.core.LatLon.toString ()

```
                              {
        return new StringBuilder().append("[").append(lat).append(",")
                .append(lon).append("]").toString();
    }
```

6.58 edu.ou.asgbook.core.LatLonGrid Class Reference

A geospatial grid of data in equilat equilon coordinates typically in WGS84 ellipsoid.

Collaboration diagram for edu.ou.asgbook.core.LatLonGrid:



Public Member Functions

- LatLonGrid (int[ ][ ] data, int missing, LatLon nwCorner, double latres, double lonres)
- LatLonGrid crop (int startRow, int startCol, int numLat, int numLon)

    Crop this grid.

- LatLonGrid (int nrows, int ncols, int missing, LatLon nwCorner, double latres, double lonres)

    Initialize a grid of data at zero.

- int[ ][ ] getData ()
- int getMissing ()
- LatLon getNwCorner ()

    Note that this is the true corner, not the center of the first grid point.

- double getLatRes ()
- double getLonRes ()
- LatLon getLocation (int row, int col)
- LatLon getLocation (Pixel p)
- LatLon getLocation (double row, double col)
- LatLon getSeCorner ()

    This is the true corner, not the middle of the last grid point.

- int getNumLon ()
- int getNumLat ()
- int getValue (int row, int col)
- void setValue (int row, int col, int value)
- void setMissing (int i)
- final int getRow (LatLon location)

The returned row may be outside this grid's dimensions.

- final Pixel getPixel (LatLon location)
- int getCol (LatLon location)

  The returned col may be outside this grid's dimensions.

- final int getValue (LatLon location)
- final boolean isValid (int row, int col)

  Are the pixel coordinates in bounds?

- void fill (int newval)
- void replace (int oldval, int newval)
- int getValue (Pixel pixel)
- Pixel[ ] asPixels ()
- int[ ][ ] longitudewrap (int Ny)
- LatLonGrid remapTo (LatLonGrid other)

Static Public Member Functions

- static LatLonGrid copyOf (final LatLonGrid original)

  Make a deep copy.

- static LatLonGrid add (LatLonGrid a, LatLonGrid b)

6.58.1 Detailed Description

A geospatial grid of data in equilat equilon coordinates typically in WGS84 ellipsoid.

Author:

Valliappa.Lakshmanan

6.58.2 Constructor & Destructor Documentation

6.58.2.1 edu.ou.asgbook.core.LatLonGrid.LatLonGrid (int *data*[ ][ ], int *missing*, LatLon *nwCorner*, double *latres*, double *lonres*)

Parameters:

data Holds on to provided data (does not clone the data)

missing Missing data value, typically -9999 or similar

nwCorner the true corner, not the center of the first grid point

latres A positive number

lonres A positive number

```
                                        {
        super();
        this.data = data;
        this.missing = missing;
        this.nwCorner = nwCorner;
        this.latRes = latres;
        this.lonRes = lonres;
    }
```

6.58.2.2 edu.ou.asgbook.core.LatLonGrid.LatLonGrid (int *nrows*, int *ncols*, int *missing*, LatLon *nwCorner*, double *latres*, double *lonres*)

Initialize a grid of data at zero.

Parameters:

nrows

ncols

missing Missing data value, typically -9999 or similar

nwCorner the true corner, not the center of the first grid point

latres A positive number

lonres A positive number

```
                                        {
        this( new int[nrows][ncols], missing, nwCorner, latres, lonres );
    }
```

6.58.3 Member Function Documentation

6.58.3.1 static LatLonGrid edu.ou.asgbook.core.LatLonGrid.add (LatLonGrid *a*, LatLonGrid *b*) [static]

```
                                                            {
        int nrows = a.getNumLat();
        int ncols = a.getNumLon();
        if (b.getNumLat() != nrows || b.getNumLon() != ncols){
            throw new IllegalArgumentException("Grids are of different dimensions: first grid is " + r
        }
        LatLonGrid result = LatLonGrid.copyOf(a);
        for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
            if (result.data[i][j] != result.missing){
                int bval = b.data[i][j];
                if (bval != b.missing){
                    result.data[i][j] += bval;
                } else {
                    result.data[i][j] = result.missing;
                }
            }
        }
        return result;
    }
```

6.58.3.2 Pixel [ ] edu.ou.asgbook.core.LatLonGrid.asPixels ()

```
                              {
        List<Pixel> pixels = new ArrayList<Pixel>();
        for (int i=0; i < data.length; ++i) for (int j=0; j < data[i].length; ++j){
            if (data[i][j] != missing){
                pixels.add(new Pixel(i,j,data[i][j]));
            }
        }
        return pixels.toArray(new Pixel[0]);
    }
```

6.58.3.3 static LatLonGrid edu.ou.asgbook.core.LatLonGrid.copyOf (final LatLonGrid *original*) [static]

Make a deep copy.

```
                                                          {
        int[][] copy = new int[original.getNumLat()][original.getNumLon()];
        for (int i=0; i < original.getNumLat(); ++i){
            for (int j=0; j < original.getNumLon(); ++j){
                copy[i][j] = original.data[i][j];
            }
        }
        return new LatLonGrid(copy, original.missing, original.nwCorner, original.latRes, o
    }
```

6.58.3.4 LatLonGrid edu.ou.asgbook.core.LatLonGrid.crop (int *startRow*, int *startCol*, int *numLat*, int *numLon*)

Crop this grid.

Does not check dimensions

```
                                                                          {
        int[][] copy = new int[numLat][numLon];
        for (int i=0; i < numLat; ++i){
            for (int j=0; j < numLon; ++j){
                copy[i][j] = data[i+startRow][j+startCol];
            }
        }
        LatLon origin = this.getLocation(startRow, startCol);
        LatLon nwCorner = new LatLon( origin.getLat() + latRes/2 , origin.getLon() - lonRes/2 );
        return new LatLonGrid(copy, missing, nwCorner, latRes, lonRes);
    }
```

6.58.3.5 void edu.ou.asgbook.core.LatLonGrid.fill (int *newval*)

```
                                 {
        final int nrows = data.length;
        final int ncols = data[0].length;
        for (int i=0; i < nrows; ++i){
            for (int j=0; j < ncols; ++j){
                data[i][j] = newval;
            }
        }
    }
```

6.58.3.6 int edu.ou.asgbook.core.LatLonGrid.getCol (LatLon *location*)

The returned col may be outside this grid's dimensions.

```
                                        {
        int col = (int) ( (location.getLon() - nwCorner.getLon())/lonRes );
        return col;
    }
```

6.58.3.7 int [ ][ ] edu.ou.asgbook.core.LatLonGrid.getData ()

```
                             {
        return data;
    }
```

6.58.3.8 double edu.ou.asgbook.core.LatLonGrid.getLatRes ()

```
                              {
        return latRes;
    }
```

6.58.3.9 LatLon edu.ou.asgbook.core.LatLonGrid.getLocation (double *row*, double *col*)

```
                                             {
        // latitude decreases, longitude increases
        return new LatLon( nwCorner.getLat() - (row+0.5)*latRes,
                nwCorner.getLon() + (col+0.5)*lonRes );
    }
```

6.58.3.10 LatLon edu.ou.asgbook.core.LatLonGrid.getLocation (Pixel *p*)

```
                              {
        return getLocation(p.getRow(), p.getCol());
    }
```

6.58.3.11 LatLon edu.ou.asgbook.core.LatLonGrid.getLocation (int *row*, int *col*)

```
                                          {
        // latitude decreases, longitude increases
        return new LatLon( nwCorner.getLat() - (row+0.5)*latRes,
                nwCorner.getLon() + (col+0.5)*lonRes );
    }
```

6.58.3.12 double edu.ou.asgbook.core.LatLonGrid.getLonRes ()

```
                          {
        return lonRes;
    }
```

6.58.3.13 int edu.ou.asgbook.core.LatLonGrid.getMissing ()

```
                          {
        return missing;
    }
```

6.58.3.14 int edu.ou.asgbook.core.LatLonGrid.getNumLat ()

```
                          {
        return data.length;
    }
```

6.58.3.15 int edu.ou.asgbook.core.LatLonGrid.getNumLon ()

```
                         {
        return data[0].length;
    }
```

6.58.3.16 LatLon edu.ou.asgbook.core.LatLonGrid.getNwCorner ()

Note that this is the true corner, not the center of the first grid point.

Returns:

```
                              {
        return nwCorner;
    }
```

6.58.3.17 final Pixel edu.ou.asgbook.core.LatLonGrid.getPixel (LatLon *location*)

```
                                               {
        int row = getRow(location);
        int col = getCol(location);
        return new Pixel(row, col, data[row][col]);
    }
```

6.58.3.18 final int edu.ou.asgbook.core.LatLonGrid.getRow (LatLon *location*)

The returned row may be outside this grid's dimensions.

```
                                           {
        int row = (int) ( (nwCorner.getLat() - location.getLat())/latRes );
        return row;
    }
```

6.58.3.19 LatLon edu.ou.asgbook.core.LatLonGrid.getSeCorner ()

This is the true corner, not the middle of the last grid point.

```
                              {
        // latitude decreases, longitude increases
        return new LatLon( nwCorner.getLat() - getNumLat()*latRes,
                nwCorner.getLon() + getNumLon()*lonRes );
    }
```

6.58.3.20 int edu.ou.asgbook.core.LatLonGrid.getValue (Pixel *pixel*)

```
                                        {
            return getValue(pixel.getX(), pixel.getY());
        }
```

6.58.3.21 final int edu.ou.asgbook.core.LatLonGrid.getValue (LatLon *location*)

```
                                                  {
            int row = getRow(location);
            int col = getCol(location);
            if ( isValid(row, col) ){
                return data[row][col];
            }
            return missing;
        }
```

6.58.3.22 int edu.ou.asgbook.core.LatLonGrid.getValue (int *row*, int *col*)

```
                                            {
            return data[row][col];
        }
```

6.58.3.23 final boolean edu.ou.asgbook.core.LatLonGrid.isValid (int *row*, int *col*)

Are the pixel coordinates in bounds?

```
                                                      {
            return row >= 0 && row < data.length && col >= 0 && col < data[row].length;
        }
```

6.58.3.24 int [ ][ ] edu.ou.asgbook.core.LatLonGrid.longitudewrap (int *Ny*)

```
                                             {
               int nrows = data.length;
               int ncols = data[0].length;
               int hy = Ny/2;
               int outcols = ncols + 2*hy;
               int[][] result = new int[nrows][outcols];
               for (int i=0; i < nrows; ++i) for (int j=0; j < outcols; ++j){
                 int incol = j - hy;
                 if (incol < 0) incol += ncols; // wrap
                 else if (incol >= ncols) incol -= ncols;
                 result[i][j] = data[i][incol];
```

```
        }
        return result;
    }
```

6.58.3.25 LatLonGrid edu.ou.asgbook.core.LatLonGrid.remapTo (LatLonGrid *other*)

```
                                            {
        LatLonGrid result = LatLonGrid.copyOf(other);
        result.setMissing(this.getMissing());
        for (int i=0; i < other.getNumLat(); ++i){
            int row = getRow( other.getLocation(i,0) );
            for (int j=0; j < other.getNumLon(); ++j){
                int col = getCol( other.getLocation(i,j) );
                if (this.isValid(row,col)){
                    result.setValue(i,j, data[row][col]);
                } else {
                    result.setValue(i,j, result.missing);
                }
            }
        }
        return result;
    }
```

6.58.3.26 void edu.ou.asgbook.core.LatLonGrid.replace (int *oldval*, int *newval*)

```
                                           {
        final int nrows = data.length;
        final int ncols = data[0].length;
        for (int i=0; i < nrows; ++i){
            for (int j=0; j < ncols; ++j){
                if (data[i][j] == oldval){
                    data[i][j] = newval;
                }
            }
        }
    }
```

6.58.3.27 void edu.ou.asgbook.core.LatLonGrid.setMissing (int *i*)

```
                                {
        missing = i;
    }
```

6.58.3.28 void edu.ou.asgbook.core.LatLonGrid.setValue (int *row*, int *col*, int *value*)

```
                                                     {
```

```
        data[row][col] = value;
    }
```

6.59 edu.ou.asgbook.core.LevelSet Class Reference

A representation of a spatial grid as a set of levels.

6.59.1 Detailed Description

A representation of a spatial grid as a set of levels.

Author:

valliappa.lakshmanan

6.60 edu.ou.asgbook.rasterization.Line Class Reference

A line that connects two points on the earth's surface.

Public Member Functions

- Line (double lat0, double lon0, double lat1, double lon1)
- Line (LatLon p0, LatLon p1)
- double getLat0 ()
- double getLon0 ()
- double getLat1 ()
- double getLon1 ()
- List< Pixel > getPositionIn (LatLonGrid grid)
- Double getXIntercept (double y)

  Find the intersection point.

- Double getYIntercept (double x)

  Find the intersection point.

Static Public Member Functions

- static void main (String args[ ]) throws Exception

6.60.1 Detailed Description

A line that connects two points on the earth's surface.

Author:

valliappa.lakshmanan

6.60.2 Constructor & Destructor Documentation

6.60.2.1 edu.ou.asgbook.rasterization.Line.Line (double *lat0*, double *lon0*, double *lat1*, double *lon1*)

```
                                                            {
           this.lat0 = lat0;
           this.lon0 = lon0;
           this.lat1 = lat1;
```

```
        this.lon1 = lon1;
    }
```

6.60.2.2 edu.ou.asgbook.rasterization.Line.Line (LatLon *p0*, LatLon *p1*)

```
                                    {
        this.lat0 = p0.getLat();
        this.lon0 = p0.getLon();
        this.lat1 = p1.getLat();
        this.lon1 = p1.getLon();
    }
```

6.60.3 Member Function Documentation

6.60.3.1 double edu.ou.asgbook.rasterization.Line.getLat0 ()

```
                         {
        return lat0;
    }
```

6.60.3.2 double edu.ou.asgbook.rasterization.Line.getLat1 ()

```
                         {
        return lat1;
    }
```

6.60.3.3 double edu.ou.asgbook.rasterization.Line.getLon0 ()

```
                         {
        return lon0;
    }
```

6.60.3.4 double edu.ou.asgbook.rasterization.Line.getLon1 ()

```
                         {
        return lon1;
    }
```

6.60.3.5 List<Pixel> edu.ou.asgbook.rasterization.Line.getPositionIn (LatLonGrid *grid*)

```
                                                  {
        List<Pixel> result = new ArrayList<Pixel>();
        Pixel p0 = grid.getPixel( new LatLon(lat0, lon0) );
        Pixel p1 = grid.getPixel( new LatLon(lat1, lon1) );
        System.out.println("Line from " + p0 + " to " + p1);
        int rowlen = Math.abs(p0.getRow() - p1.getRow());
        int collen = Math.abs(p0.getCol() - p1.getCol());
        // avoid divide by zero in slope calculations below
        if ( rowlen == 0 && collen == 0){
            result.add(p0);
            return result;
        }
        if ( rowlen > collen ){
            // increment in row
            int startrow = Math.min(p0.getRow(), p1.getRow());
            int endrow = Math.max(p0.getRow(), p1.getRow());
            double slope = (p1.getCol() - p0.getCol())/((double)(p1.getRow()-p0.getRow()));
            for (int row=startrow; row <= endrow; ++row){
                int col = (int) Math.round(slope*(row-p0.getRow())+p0.getCol());
                if (grid.isValid(row, col)){
                    result.add( new Pixel(row, col, grid.getValue(row, col)) );
                }
            }
        } else {
            int startcol = Math.min(p0.getCol(), p1.getCol());
            int endcol = Math.max(p0.getCol(), p1.getCol());
            double slope = (p1.getRow()-p0.getRow())/((double)(p1.getCol()-p0.getCol()));
            for (int col=startcol; col <= endcol; ++col){
                int row = (int) Math.round(slope*(col-p0.getCol())+p0.getRow());
                if (grid.isValid(row, col)){
                    result.add( new Pixel(row, col, grid.getValue(row, col)) );
                }
            }
        }
        return result;
    }
```

6.60.3.6 Double edu.ou.asgbook.rasterization.Line.getXIntercept (double *y*)

Find the intersection point.

Returns null if not in range.

```
                                          {
        if (!isBetween(lon0, y, lon1)) {
            return null;
        }
        // if y0=y1, then inrange would be false
        double x;
        if (lon0 != lon1) {
            x = lat0 + (y - lon0) * (lat1 - lat0) / (lon1 - lon0);
        } else {
            x = (lat1 + lat0) / 2;
        }
        return x;
    }
```

6.60.3.7 Double edu.ou.asgbook.rasterization.Line.getYIntercept (double *x*)

Find the intersection point.

Returns null if not in range.

```
                                          {
        if (!isBetween(lat0, x, lat1)) {
            return null;
        }
        double y;
        if (lat0 != lat1) {
            y = lon0 + (x - lat0) * (lon1 - lon0) / (lat1 - lat0);
        } else {
            y = (lon1 + lon0) / 2;
        }
        return y;
    }
```

6.60.3.8 static void edu.ou.asgbook.rasterization.Line.main (String *args*[ ]) throws Exception [static]

```
                                                            {
        LatLonGrid grid = new LatLonGrid(100,100,0,new LatLon(100,-90),0.01,0.01);
        List<Pixel> ver = new Line(99.3,-89.3,99.7,-89.4).getPositionIn(grid);
        List<Pixel> hor = new Line(99.3,-89.3,99.4,-89.7).getPositionIn(grid);
        for (Pixel p : ver){
            grid.setValue(p.getRow(), p.getCol(), 10);
        }
        for (Pixel p : hor){
            grid.setValue(p.getRow(), p.getCol(), 20);
        }

        File out = OutputDirectory.getDefault("raster");
        KmlWriter.write(grid, out, "drawlines", PngWriter.createCoolToWarmColormap());
    }
```

6.61 edu.ou.asgbook.linearity.LinearityVerifier Class Reference

Given a 2D array of points, reports error measures of assuming linearity.

Static Public Member Functions

- static ScalarStatistic verify (int[ ][ ] data, DataSelector selector, DataTransform transform, int neighSize)

  Returns the Mean Square Error statistic.

- static void main (String[ ] args) throws Exception

Classes

- interface **DataSelector**
- class **InRange**
- class **NotMissing**

6.61.1 Detailed Description

Given a 2D array of points, reports error measures of assuming linearity.

By passing in different transformations, it is possible to compare potential ways of transforming the data.

Author:

valliappa.lakshmanan

6.61.2 Member Function Documentation

6.61.2.1 static void edu.ou.asgbook.linearity.LinearityVerifier.main (String[ ] *args*) throws Exception [static]

```
                                                             {
        // read input
        LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new Li
        for (int i=0; i < popdensity.getNumLat(); ++i){
            for (int j=0; j < popdensity.getNumLon(); ++j){
                 if (popdensity.getValue(i,j) < 1){
                     popdensity.setValue(i, j, popdensity.getMissing());
                 }
             }
        }

        // DataSelector selector = new NotMissing(popdensity.getMissing());
        DecimalFormat df = new DecimalFormat("0.0");
        // int maxval = new NHighest(1).findHighestValued(popdensity)[0].getValue();
        int[] neighSize = new int[]{ 1, 3, 5, 11, 21, 31, 41 };
        int[] thresh1 = new int[]{      1,   1,   50,   500,   5000, 50000 };
        int[] thresh2 = new int[]{ 500000, 50, 500, 5000, 50000, 500000 };
        final String sep = " & ";
        System.out.println("D & data range & N & RMSE (raw) & RMSE (log) \\\\ \\hline");
        for (int D : neighSize){
            for (int i=0; i < thresh1.length; ++i){
                int minval = thresh1[i];
                int maxval = thresh2[i];
                DataSelector selector = new InRange(minval, maxval, popdensity.getMissing());
                // check linearity two ways
                ScalarStatistic logstat = verify(popdensity.getData(), selector, new LogScaling(10), D
                ScalarStatistic rawstat = verify(popdensity.getData(), selector, new LinearScaling(100
                System.out.println(D + sep +
                        minval + "-" + maxval + sep +
                        rawstat.getNumSamples() + sep +
                        df.format(Math.sqrt(rawstat.getMean())) + sep +
                        df.format(Math.sqrt(logstat.getMean())) + " \\\\");
            }
            System.out.println("\\hline");
        }
    }
```

6.61.2.2 static ScalarStatistic edu.ou.asgbook.linearity.LinearityVerifier.verify (int *data*[ ][ ], DataSelector *selector*, DataTransform *transform*, int *neighSize*) [static]

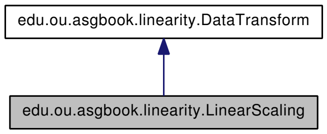
Returns the Mean Square Error statistic.

```

        // setup
        ScalarStatistic errorstat = new ScalarStatistic();
        int nrows = data.length;
        if ( nrows == 0 ){
            return errorstat;
        }
        int ncols = data[0].length;
        if ( ncols == 0 ){
            return errorstat;
        }

        // find the error in every triad interpolating along rows
        for (int col=0; col < ncols; ++col){
            for (int row=neighSize; row < nrows-neighSize; ++row){
                if (selector.shouldSelect(data[row][col], data[row-neighSize][col], data[row+neighSize]
                    int actualValue = data[row][col];
                    double trans0 = transform.transform(data[row-neighSize][col]);
                    double trans1 = transform.transform(data[row+neighSize][col]);
                    double trans_interp = (trans0 + trans1)/2;
                    double interpValue = transform.inverse(trans_interp);
                    double error = (interpValue - actualValue);
                    errorstat.update(error*error);
                }
            }
        }

        // repeat for columns
        for (int row=0; row < nrows; ++row){
            for (int col=neighSize; col < ncols-neighSize; ++col){
                if (selector.shouldSelect(data[row][col], data[row][col-neighSize], data[row
                    int actualValue = data[row][col];
                    double trans0 = transform.transform(data[row][col-neighSize]);
                    double trans1 = transform.transform(data[row][col+neighSize]);
                    double trans_interp = (trans0 + trans1)/2;
                    double interpValue = transform.inverse(trans_interp);
                    double error = (interpValue - actualValue);
                    errorstat.update(error*error);
                }
            }
        }

        return errorstat;
    }
```

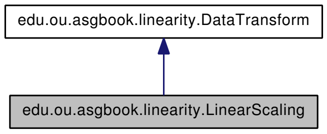
6.62 edu.ou.asgbook.linearity.LinearScaling Class Reference

Scales pixel values as Ax.

Inheritance diagram for edu.ou.asgbook.linearity.LinearScaling:


edu.ou.asgbook.linearity.DataTransform
edu.ou.asgbook.linearity.LinearScaling

Collaboration diagram for edu.ou.asgbook.linearity.LinearScaling:


edu.ou.asgbook.linearity.DataTransform
edu.ou.asgbook.linearity.LinearScaling

Public Member Functions

- LinearScaling (double multiplier)

    Multiply input values by this amount.

- Override double transform (double value)
- Override double inverse (double value)

6.62.1 Detailed Description

Scales pixel values as Ax.

This is useful since the LatLonGrid stores integers.

Author:

valliappa.lakshmanan

6.62.2 Constructor & Destructor Documentation

6.62.2.1 edu.ou.asgbook.linearity.LinearScaling.LinearScaling (double *multiplier*)

Multiply input values by this amount.

```
                                    {
        this.scale = multiplier;
    }
```

6.62.3 Member Function Documentation

6.62.3.1 Override double edu.ou.asgbook.linearity.LinearScaling.inverse (double *value*) [virtual]

Implements edu.ou.asgbook.linearity.DataTransform.

```
                                 {
        return (value / scale);
    }
```

6.62.3.2 Override double edu.ou.asgbook.linearity.LinearScaling.transform (double *value*) [virtual]

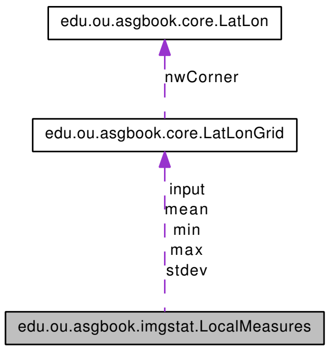
Implements edu.ou.asgbook.linearity.DataTransform.

```
                                  {
        return (scale * value);
    }
```

6.63 edu.ou.asgbook.imgstat.LocalMeasures Class Reference

Statistics computed in the neighborhood of a pixel.

Collaboration diagram for edu.ou.asgbook.imgstat.LocalMeasures:


edu.ou.asgbook.core.LatLon
nwCorner
edu.ou.asgbook.core.LatLonGrid
input
mean
min
max
stdev
edu.ou.asgbook.imgstat.LocalMeasures

Public Member Functions

- LatLonGrid getMean ()
- LatLonGrid getStdDeviation ()
- LatLonGrid getMin ()
- LatLonGrid getMax ()
- LocalMeasures (LatLonGrid input, int Nx, int Ny)

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.63.1 Detailed Description

Statistics computed in the neighborhood of a pixel.

Author:

valliappa.lakshmanan

6.63.2 Constructor & Destructor Documentation

6.63.2.1 edu.ou.asgbook.imgstat.LocalMeasures.LocalMeasures (LatLonGrid *input*, int *Nx*, int *Ny*)

```
                                                          {
           this.hx = Nx/2;
           this.hy = Ny/2;
           this.input = input;
           this.mean = LatLonGrid.copyOf(input);
           this.stdev  = LatLonGrid.copyOf(input);
           this.min  = LatLonGrid.copyOf(input);
           this.max  = LatLonGrid.copyOf(input);
           compute();
       }
```

6.63.3 Member Function Documentation

6.63.3.1 LatLonGrid edu.ou.asgbook.imgstat.LocalMeasures.getMax ()

```
                               {
           return max;
       }
```

6.63.3.2 LatLonGrid edu.ou.asgbook.imgstat.LocalMeasures.getMean ()

```
                                {
           return mean;
       }
```

6.63.3.3 LatLonGrid edu.ou.asgbook.imgstat.LocalMeasures.getMin ()

```
                               {
           return min;
       }
```

6.63.3.4 LatLonGrid edu.ou.asgbook.imgstat.LocalMeasures.getStdDeviation ()

```
                                        {
           return stdev;
       }
```

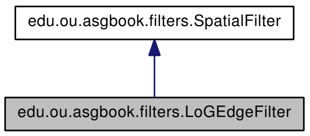
6.63.3.5 static void edu.ou.asgbook.imgstat.LocalMeasures.main (String[ ] *args*) throws Exception [static]

```
                                                             {
        // log-scaled population density
        LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new GlobalPopulati
        popdensity = popdensity.crop(900, 2500, 200, 200);
        File out = OutputDirectory.getDefault("localstat");

        KmlWriter.write(popdensity, out, "popdensity", PngWriter.createCoolToWarmColormap());
        for (int neigh = 5; neigh < 12; neigh += 6){ // 5, 11
            LocalMeasures stat = new LocalMeasures(popdensity, neigh, neigh);
            KmlWriter.write(stat.getMean(), out, "mean_" + neigh, PngWriter.createCoolToWarmColormap())
            KmlWriter.write(stat.getStdDeviation(), out, "stdev_" + neigh, PngWriter.createCoolToWarmCo
            KmlWriter.write(stat.getMin(), out, "min_" + neigh, PngWriter.createCoolToWarmColormap());
            KmlWriter.write(stat.getMax(), out, "max_" + neigh, PngWriter.createCoolToWarmColormap());
        }
    }
```

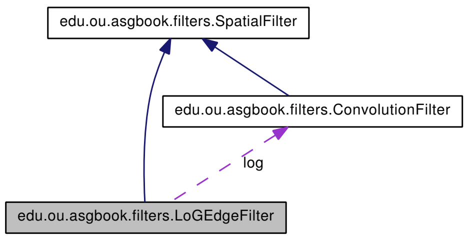
6.64 edu.ou.asgbook.filters.LoGEdgeFilter Class Reference

Laplacian of a Gaussian edge filter.

Inheritance diagram for edu.ou.asgbook.filters.LoGEdgeFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.LoGEdgeFilter

Collaboration diagram for edu.ou.asgbook.filters.LoGEdgeFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.ConvolutionFilter
log
edu.ou.asgbook.filters.LoGEdgeFilter

Public Member Functions

- LoGEdgeFilter (int halfsize, int edgethresh)
- Override LatLonGrid filter (LatLonGrid input)
- LatLonGrid edgeFilter (final LatLonGrid input)
- LatLonGrid edgeFilter (final LatLonGrid input, File out)

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.64.1 Detailed Description

Laplacian of a Gaussian edge filter.

Author:

valliappa.lakshmanan

6.64.2 Constructor & Destructor Documentation

6.64.2.1 edu.ou.asgbook.filters.LoGEdgeFilter.LoGEdgeFilter (int *halfsize*, int *edgethresh*)

```
                                                      {
        double sigma = halfsize/3.0;
        double[][] coeffs = new double[2*halfsize+1][2*halfsize+1];
        double tot = 0;
        for (int x=-halfsize; x <= halfsize; ++x){
            for (int y=-halfsize; y <= halfsize; ++y){
                double term1 = (x*x + y*y - sigma*sigma)/Math.pow(sigma,4);
                double term2 = Math.exp(-(x*x + y*y)/(2*sigma*sigma));
                double coeff = term1 * term2;
                coeffs[x+halfsize][y+halfsize] = coeff;
                tot += coeff;
            }
        }
        // ensure that coeffs add up to zero
        coeffs[halfsize][halfsize] -= tot;
        this.log = new ConvolutionFilter(coeffs);
        this.thresh = edgethresh;
    }
```

6.64.3 Member Function Documentation

6.64.3.1 LatLonGrid edu.ou.asgbook.filters.LoGEdgeFilter.edgeFilter (final LatLonGrid *input*, File *out*)

```
                                                                  {
        // Laplacian of a Gaussian
        LatLonGrid dgg = log.convolve(input);
        KmlWriter.debugWrite(dgg, out, "laplacianofgaussian");
        int bound = log.getFilterNumRows();
        // find zero crossings in 3x3 neighborhood
        int nrows = dgg.getNumLat();
        int ncols = dgg.getNumLon();
        LatLonGrid result = new LatLonGrid(nrows,ncols,-1,dgg.getNwCorner(),dgg.getLatRes(),dgg.getLonR
        for (int i=bound; i < (nrows-bound); ++i) for (int j=bound; j < (ncols-bound); ++j){
            int mag = 0;
            // ver
            mag = checkZeroCrossing(dgg.getValue(i-1,j), dgg.getValue(i+1,j), mag);
            // hor
            mag = checkZeroCrossing(dgg.getValue(i,j-1), dgg.getValue(i,j+1), mag);
            // diag1
            mag = checkZeroCrossing(dgg.getValue(i-1,j-1), dgg.getValue(i+1,j+1), mag);
            // diag2
            mag = checkZeroCrossing(dgg.getValue(i+1,j-1), dgg.getValue(i-1,j+1), mag);
            // mag is over 2 bins
            mag /= 2;
            if ( mag > thresh ){
                result.setValue(i,j, mag);
            }
        }
        return result;
    }
```

6.64.3.2 LatLonGrid edu.ou.asgbook.filters.LoGEdgeFilter.edgeFilter (final LatLonGrid *input*)

```
                                                        {
        return edgeFilter(input, null);
    }
```

6.64.3.3 Override LatLonGrid edu.ou.asgbook.filters.LoGEdgeFilter.filter (LatLonGrid *input*)

```
                                               {
        return edgeFilter(input);
    }
```

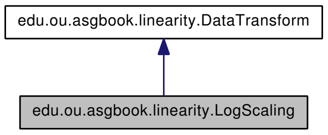
6.64.3.4 static void edu.ou.asgbook.filters.LoGEdgeFilter.main (String[ ] *args*) throws Exception [static]

```
                                                            {
        // create output directory
        File out = OutputDirectory.getDefault("logedge");

        // read input
        DataTransform t = new GlobalPopulation.LogScaling();
        LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, t).crop
        KmlWriter.write(popdensity, out, "orig", PngWriter.createCoolToWarmColormap());

        LoGEdgeFilter filter = new LoGEdgeFilter(5,400);
        LatLonGrid edges = filter.edgeFilter(popdensity, out);
        KmlWriter.write(edges, out, "logedge", PngWriter.createCoolToWarmColormap());
    }
```

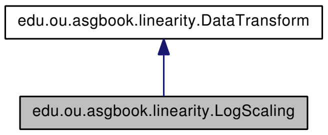
6.65 edu.ou.asgbook.linearity.LogScaling Class Reference

Transforms pixel values as log(x).

Inheritance diagram for edu.ou.asgbook.linearity.LogScaling:


edu.ou.asgbook.linearity.DataTransform
edu.ou.asgbook.linearity.LogScaling

Collaboration diagram for edu.ou.asgbook.linearity.LogScaling:


edu.ou.asgbook.linearity.DataTransform
edu.ou.asgbook.linearity.LogScaling

Public Member Functions

- LogScaling (double multiplier)

  Multiply log(input) values by this amount i.e.

- Override double transform (double value)
- Override double inverse (double value)

6.65.1 Detailed Description

Transforms pixel values as log(x).

Author:

valliappa.lakshmanan

6.65.2 Constructor & Destructor Documentation

6.65.2.1 edu.ou.asgbook.linearity.LogScaling.LogScaling (double *multiplier*)

Multiply log(input) values by this amount i.e.

it is multiplier∗log(value)

```
                                         {
        this.scale = multiplier;
    }
```

6.65.3 Member Function Documentation

6.65.3.1 Override double edu.ou.asgbook.linearity.LogScaling.inverse (double *value*) [virtual]

Implements edu.ou.asgbook.linearity.DataTransform.

```
                                         {
        if ( value == 0 ){
            return 1;
        } else {
            return Math.pow(10, value/scale);
        }
    }
```

6.65.3.2 Override double edu.ou.asgbook.linearity.LogScaling.transform (double *value*) [virtual]

Implements edu.ou.asgbook.linearity.DataTransform.

```
                                          {
        if ( value > 1 ){
            return (scale*Math.log10(value));
        } else {
            return 0;
        }
    }
```

6.66 edu.ou.asgbook.dataset.MadisTemperature Class Reference

Reads the ASCII temperature data available at http://madis-data.noaa.gov/public/sfcdumpguest.html.

Static Public Member Functions

- static PointObservations read (File file) throws IOException
- static void main (String[ ] args) throws Exception

Static Public Attributes

- static final File TN_Oct2010 = new File("data/madishydro/tn_oct2010_-temp.txt")

Package Functions

- SuppressWarnings ("unused") public static PointObservations read(Reader r) throws IOException

6.66.1 Detailed Description

Reads the ASCII temperature data available at http://madis-data.noaa.gov/public/sfcdumpguest.html.

Author:

Valliappa.Lakshmanan

6.66.2 Member Function Documentation

6.66.2.1 static void edu.ou.asgbook.dataset.MadisTemperature.main (String[ ] *args*) throws Exception [static]

```
                                                                  {
        PointObservations data = MadisTemperature.read(MadisTemperature.TN_Oct2010);
        for (int i=0; i < data.getPoints().length; ++i){
            System.out.println(data.getPoints()[i]);
        }
    }
```

6.66.2.2 static PointObservations edu.ou.asgbook.dataset.MadisTemperature.read (File *file*) throws IOException [static]

```
                                                                        {
        Reader f = null;
        if (file.getAbsolutePath().endsWith(".gz")) {
            f = new InputStreamReader(new GZIPInputStream(new FileInputStream(
                    file)));
        } else {
            f = new FileReader(file);
        }
        return read(f);
    }
```

6.66.2.3 edu.ou.asgbook.dataset.MadisTemperature.SuppressWarnings ("unused") throws IOException [package]

```
                                                                       {
        Scanner s = new Scanner(r);
        List<PointObservations.ObservationPoint> result = new ArrayList<PointObservations.Ob

        final int FACTOR = 10;
        final int MISSING = -99999 * FACTOR;
        s.nextLine(); // header
        while (s.hasNext()){
            String station = s.next();
            String date = s.next();
            String time = s.next();
            int precip = (int) Math.round( s.nextDouble() * FACTOR );
            double lat = s.nextDouble();
            double lon = s.nextDouble();
            result.add(new PointObservations.ObservationPoint(lat,lon,precip));
        }

        PointObservations.ObservationPoint[] pts = result.toArray(new PointObservations.Obse
        return new PointObservations(pts, MISSING);
    }
```

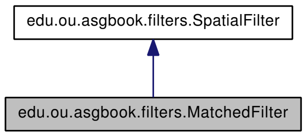
6.66.3 Member Data Documentation

6.66.3.1 final File edu.ou.asgbook.dataset.MadisTemperature.TN_Oct2010 = new File("data/madishydro/tn_oct2010_temp.txt") [static]

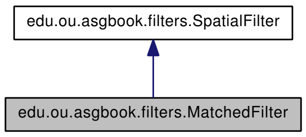
6.67 edu.ou.asgbook.filters.MatchedFilter Class Reference

Convolve an image by a window that is akin to the features we want to extract.

Inheritance diagram for edu.ou.asgbook.filters.MatchedFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.MatchedFilter

Collaboration diagram for edu.ou.asgbook.filters.MatchedFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.MatchedFilter

Public Member Functions

- MatchedFilter (double[ ][ ] coeffs)
- LatLonGrid match (final LatLonGrid input)

  returns a grid with values in the range 0-100

- Override LatLonGrid filter (LatLonGrid input)

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.67.1 Detailed Description

Convolve an image by a window that is akin to the features we want to extract.

Author:

Valliappa.Lakshmanan

6.67.2 Constructor & Destructor Documentation

6.67.2.1 edu.ou.asgbook.filters.MatchedFilter.MatchedFilter (double *coeffs*[ ][ ])

```
                                        {
        this.coeffs = coeffs;
        if ( coeffs.length % 2 == 0 || coeffs[0].length % 2 == 0 ){
            throw new IllegalArgumentException("Dimensions of coefficients array needs to be
        }
        // normalize
        double sum = 0;
        for (int i=0; i < coeffs.length; ++i){
            for (int j=0; j < coeffs[i].length; ++j){
                sum += coeffs[i][j];
            }
        }
        System.out.println("Normalizing coefficients by " + sum);
        DecimalFormat df = new DecimalFormat("0.000");
        for (int i=0; i < coeffs.length; ++i){
            for (int j=0; j < coeffs[i].length; ++j){
                coeffs[i][j] /= sum;
                System.out.print(df.format(coeffs[i][j]) + "&"); // for LaTeX
            }
            System.out.println("\\\\");
        }
    }
```

6.67.3 Member Function Documentation

6.67.3.1 Override LatLonGrid edu.ou.asgbook.filters.MatchedFilter.filter (LatLonGrid *input*)

```
                                              {
        return match(input);
    }
```

6.67.3.2 static void edu.ou.asgbook.filters.MatchedFilter.main (String[ ] *args*) throws Exception [static]

```
                                                            {
        // create output directory
        File out = OutputDirectory.getDefault("matched");

        // read input
        LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new Glo
        KmlWriter.write(popdensity, out, "orig", PngWriter.createCoolToWarmColormap());

        for (int hx=1; hx <= 5; hx += 2){
            int hy = hx*2;
            double[][] coeffs = new double[2*hx+1][2*hy+1];
            for (int i=0; i < coeffs.length; ++i){
                for (int j=0; j < coeffs[i].length; ++j){
                    int t = i + j;
                    if ( t < coeffs.length ) coeffs[i][j] = 1;
                }
            }
             MatchedFilter filter = new MatchedFilter(coeffs);
             LatLonGrid sm = filter.match(popdensity);
             KmlWriter.write(sm, out, "northwest"+hx, PngWriter.createCoolToWarmColormap());
         }

         for (int hx=5; hx < 15; hx += 3){
             // int hx = 8;
             int hy = hx;
             double[][] coeffs = new double[2*hx+1][2*hy+1];
             for (int i=0; i < coeffs.length; ++i){
                 for (int j=0; j < coeffs[i].length; ++j){
                     int dx = i - hx;
                     int dy = j - hy;
                     if ( Math.abs(dx) < hx/2 && Math.abs(dy) < hy/2 )
                         coeffs[i][j] = 1;
                 }
             }
             MatchedFilter filter = new MatchedFilter(coeffs);
             LatLonGrid sm = filter.match(popdensity);
             KmlWriter.write(sm, out, "isolated"+hx, PngWriter.createCoolToWarmColormap());
         }
     }
```

6.67.3.3 LatLonGrid edu.ou.asgbook.filters.MatchedFilter.match (final LatLonGrid *input*)

returns a grid with values in the range 0-100

```
                                                    {
        LatLonGrid output = LatLonGrid.copyOf(input);
        output.setMissing(-1);
        output.fill(output.getMissing());
        int[][] outData = output.getData();
        int[][] inData = input.getData();
        final int hx = coeffs.length / 2;
        final int hy = coeffs[0].length / 2;
        final int nx = output.getNumLat();
        final int ny = output.getNumLon();
        for (int i=hx; i < (nx-hx); ++i){
            for (int j=hy; j < (ny-hy); ++j){
                double tot = 0;
                int totval = 0; // normalize values in window
                for (int m=-hx; m <= hx; ++m){
                    for (int n=-hy; n <= hy; ++n){
                        double coeff = coeffs[m+hx][n+hy];
                        int inval = inData[i+m][j+n];
                        if (inval != input.getMissing()){
                            tot += inval*coeff;
                            totval += inval;
                        }
                    }
                }
                if (totval != 0){
                    outData[i][j] = (int) Math.round(10000 * tot / totval);
                }
            }
        }
        return output;
    }
```

6.68 edu.ou.asgbook.filters.MaxValueFilter Class Reference

Finds the highest value pixel in the image.

Public Member Functions

- Result findHighestValued (LatLonGrid input)

Static Public Member Functions

- static void main (String[ ] args) throws Exception

Classes

- class **Result**

6.68.1 Detailed Description

Finds the highest value pixel in the image.

Author:

Valliappa.Lakshmanan

6.68.2 Member Function Documentation

6.68.2.1 Result edu.ou.asgbook.filters.MaxValueFilter.findHighestValued (LatLonGrid *input*)

```
                                                        {
        int[][] data = input.getData();
        int x = -1;
        int y = -1;
        int maxval = input.getMissing();
        for (int i=0; i < input.getNumLat(); ++i){
            for (int j=0; j < input.getNumLon(); ++j){
                if ( data[i][j] != input.getMissing() ){
                    if ( maxval == input.getMissing() ||
                         maxval < data[i][j] ){
                        x = i; // new maximum
                        y = j;
                        maxval = data[x][y];
                    }
                }
            }
        }
        if ( x >=0 && y >= 0 ){
            LatLon loc = input.getLocation(x, y);
            return new Result(data[x][y], loc);
        }
        return null;
    }
```

6.68.2.2 static void edu.ou.asgbook.filters.MaxValueFilter.main (String[ ] *args*) throws Exception [static]

```
                                                            {
        // read input
        LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA);
        popdensity.setMissing(0); // will get to process less data this way

        // find 10 highest
        MaxValueFilter filter = new MaxValueFilter();
        Result result = filter.findHighestValued(popdensity);
        System.out.println("Maximum is " + result.value + " at " + result.location);
    }
```

6.69 edu.ou.asgbook.thinning.MedialAxisSkeletonization Class Reference

The MAT method of skeletonizing a grid.

Static Public Member Functions

- static LatLonGrid findSkeleton (LatLonGrid input, int thresh, File out) throws Exception
- static void main (String[ ] args) throws Exception

6.69.1 Detailed Description

The MAT method of skeletonizing a grid.

Author:

v.lakshmanan

6.69.2 Member Function Documentation

6.69.2.1 static LatLonGrid edu.ou.asgbook.thinning.MedialAxisSkeletonization.findSkeleton (LatLonGrid *input*, int *thresh*, File *out*) throws Exception `[static]`

```
                                                                                                    {
        // threshold and invert
        LatLonGrid binaryImage = new SimpleThresholder(thresh).threshold(input);
        if (out != null){
            KmlWriter.write(binaryImage, out, "thresh", PngWriter.createCoolToWarmColormap());
        }
        binaryImage = new Inverter(1).invert(binaryImage);

        // compute distance to pts > 0 i.e. boundary pixels
        LatLonGrid edt = new EuclideanDTSaito().getDistanceTransform(binaryImage, 0);
        if (out != null){
            KmlWriter.write(edt, out, "edt", PngWriter.createCoolToWarmColormap());
        }

        // retain local maximum in 4-neighborhood
        LatLonGrid result = new LatLonGrid(edt.getNumLat(),edt.getNumLon(),edt.getMissing(),edt.getNwCo
        for (int i=1; i < edt.getNumLat()-1; ++i){
            for (int j=1; j < edt.getNumLon()-1; ++j){
                int edtval = edt.getValue(i, j);
                if ( edtval != 0 &&
                     edt.getValue(i-1,j) <= edtval &&
                     edt.getValue(i,j-1) <= edtval &&
                     edt.getValue(i+1,j) <= edtval &&
                     edt.getValue(i,j+1) <= edtval ){
                    result.setValue(i,j, 1);
                } else {
                    result.setValue(i,j, 0);
                }
            }
        }
        return result;
    }
```

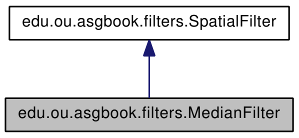
6.69.2.2 static void edu.ou.asgbook.thinning.MedialAxisSkeletonization.main (String[ ] *args*) throws Exception [static]

```
                                                     {
     File out = OutputDirectory.getDefault("matskeleton");
     LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new Gl
     KmlWriter.write(popdensity, out, "orig", PngWriter.createCoolToWarmColormap());

     popdensity = new DilateErodeFilter(2,3).filter(popdensity);
     // popdensity = new ErosionFilter(3).filter(popdensity);
     popdensity = new ErodeDilateFilter(2,3).filter(popdensity);
     KmlWriter.write(popdensity, out, "filledin", PngWriter.createCoolToWarmColormap());

     LatLonGrid result = findSkeleton(popdensity, 300, out);
     result.setMissing(0); // to make the 1s pop out
     KmlWriter.write(result, out, "mat", PngWriter.createCoolToWarmColormap());
 }
```

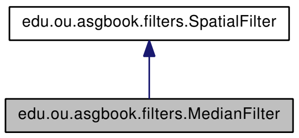
6.70 edu.ou.asgbook.filters.MedianFilter Class Reference

A smoothing operation that involves replacing a pixel by the local median.

Inheritance diagram for edu.ou.asgbook.filters.MedianFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.MedianFilter

Collaboration diagram for edu.ou.asgbook.filters.MedianFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.MedianFilter

Public Member Functions

- MedianFilter (int halfSize)
- Override LatLonGrid filter (LatLonGrid input)
- LatLonGrid smooth (final LatLonGrid input)

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.70.1 Detailed Description

A smoothing operation that involves replacing a pixel by the local median.

Author:

Valliappa.Lakshmanan

6.70.2 Constructor & Destructor Documentation

6.70.2.1 edu.ou.asgbook.filters.MedianFilter.MedianFilter (int *halfSize*)

```
                                      {
        this.halfSize = halfSize;
    }
```

6.70.3 Member Function Documentation

6.70.3.1 Override LatLonGrid edu.ou.asgbook.filters.MedianFilter.filter (LatLonGrid *input*)

```
                                               {
        return smooth(input);
    }
```

6.70.3.2 static void edu.ou.asgbook.filters.MedianFilter.main (String[ ] *args*) throws Exception [static]

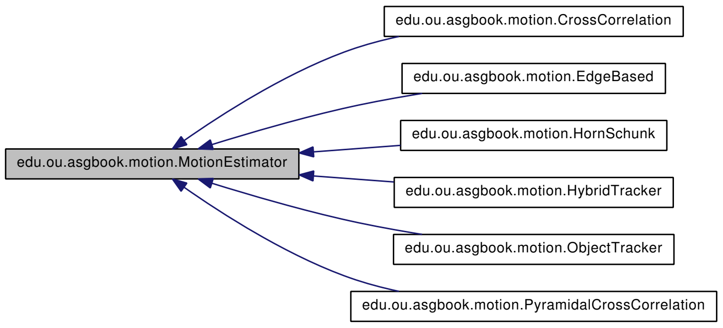
```
                                                              {
        // create output directory
        File out = OutputDirectory.getDefault("median");

        // read input
        LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new Glo
        KmlWriter.write(popdensity, out, "orig", PngWriter.createCoolToWarmColormap());

        // dilate
        LatLonGrid dilate1 = new MedianFilter(1).smooth(popdensity);
        KmlWriter.write(dilate1, out, "median_3", PngWriter.createCoolToWarmColormap());
        LatLonGrid dilate3 = new MedianFilter(3).smooth(popdensity);
        KmlWriter.write(dilate3, out, "median_7", PngWriter.createCoolToWarmColormap());
        LatLonGrid dilate5 = new MedianFilter(5).smooth(popdensity);
        KmlWriter.write(dilate5, out, "median_11", PngWriter.createCoolToWarmColormap());
    }
```

6.70.3.3 LatLonGrid edu.ou.asgbook.filters.MedianFilter.smooth (final LatLonGrid *input*)

```
                                                    {
        LatLonGrid output = LatLonGrid.copyOf(input);
        output.fill(output.getMissing());
        int[][] outData = output.getData();
        int[][] inData = input.getData();
        int hx = halfSize;
        int hy = halfSize;
        int nx = inData.length;
        int ny = inData[0].length;
        int[] arr = new int[(2*hx+1)*(2*hy+1)];
        for (int i=hx; i < (nx-hx); ++i){
            for (int j=hy; j < (ny-hy); ++j){
                int nelements = 0;
                for (int m=-hx; m <= hx; ++m){
                    for (int n=-hy; n <= hy; ++n){
                        int inval = inData[i+m][j+n];
                        if (inval != input.getMissing()){
                            arr[nelements] = inval;
                            ++nelements;
                        }
                    }
                }
                if (nelements > 0){
                    outData[i][j] = QuickSelect.kth_element(arr, nelements, nelements/2);
                }
            }
        }
        return output;
    }
```

6.71 edu.ou.asgbook.motion.MotionEstimator Interface Reference

Inheritance diagram for edu.ou.asgbook.motion.MotionEstimator:


edu.ou.asgbook.motion.CrossCorrelation
edu.ou.asgbook.motion.EdgeBased
edu.ou.asgbook.motion.HornSchunk
edu.ou.asgbook.motion.MotionEstimator
edu.ou.asgbook.motion.HybridTracker
edu.ou.asgbook.motion.ObjectTracker
edu.ou.asgbook.motion.PyramidalCrossCorrelation

Public Member Functions

- Pair< LatLonGrid, LatLonGrid > compute (LatLonGrid data0, LatLonGrid data1, File outdir)

  returns motion in the two directions.

6.71.1 Member Function Documentation

6.71.1.1 Pair<LatLonGrid, LatLonGrid> edu.ou.asgbook.motion.MotionEstimator.compute (LatLonGrid *data0*, LatLonGrid *data1*, File *outdir*)

returns motion in the two directions.

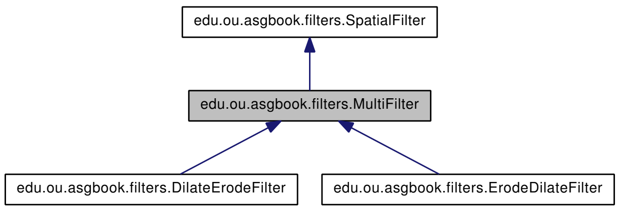
The first one is north to south and the second one is east to west. The data is aligned to second time frame. The output dir is used for intermediate products and may be null.

Implemented in edu.ou.asgbook.motion.CrossCorrelation, edu.ou.asgbook.motion.EdgeBased, edu.ou.asgbook.motion.HornSchunk, edu.ou.asgbook.motion.HybridTracker, edu.ou.asgbook.motion.ObjectTracker, and edu.ou.asgbook.motion.PyramidalCrossCorrelation.

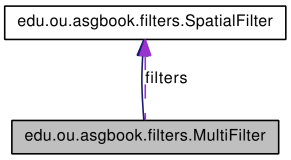
6.72 edu.ou.asgbook.filters.MultiFilter Class Reference

Carries out multiple operations.

Inheritance diagram for edu.ou.asgbook.filters.MultiFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.MultiFilter
edu.ou.asgbook.filters.DilateErodeFilter
edu.ou.asgbook.filters.ErodeDilateFilter

Collaboration diagram for edu.ou.asgbook.filters.MultiFilter:


edu.ou.asgbook.filters.SpatialFilter
filters
edu.ou.asgbook.filters.MultiFilter

Public Member Functions

- MultiFilter (SpatialFilter[ ] filters, int numTimes)
- Override LatLonGrid filter (LatLonGrid input)

6.72.1 Detailed Description

Carries out multiple operations.

Author:

Valliappa.Lakshmanan

6.72.2 Constructor & Destructor Documentation

6.72.2.1 edu.ou.asgbook.filters.MultiFilter.MultiFilter (SpatialFilter[ ] *filters*, int *numTimes*)

```
{
        super();
        this.filters = filters;
        this.numTimes = numTimes;
    }
```

6.72.3 Member Function Documentation

6.72.3.1 Override LatLonGrid edu.ou.asgbook.filters.MultiFilter.filter (LatLonGrid *input*)

```
                                            {
        LatLonGrid output = LatLonGrid.copyOf(input);
        for (int i=0; i < numTimes; ++i){
            for (SpatialFilter filter : filters){
                output = filter.filter(output);
            }
        }
        return output;
    }
```

6.73 edu.ou.asgbook.segmentation.MultiscaleKMeans-Segmenter Class Reference

Quantizes image into K levels, then does multiscale segmentation Does not implement the pruning techniques discussed in the paper.

Public Member Functions

- MultiscaleKMeansSegmenter (int thresh1, int thresh2, int K)

  Specify contouring levels.

- LatLonGrid quantize (LatLonGrid data, File out)

  Contours grid into levels 1,2,3.

- List< LabelResult > label (LatLonGrid data, File out)

  Returns K scales of output.

Static Public Member Functions

- static void main (String[ ] args) throws Exception

Classes

- class **Cluster**

6.73.1 Detailed Description

Quantizes image into K levels, then does multiscale segmentation Does not implement the pruning techniques discussed in the paper.

See: V. Lakshmanan, R. Rabin, and V. DeBrunner, "Multiscale storm identification and forecast," J. Atm. Res., vol. 67, pp. 367-380, July 2003

Author:

v.lakshmanan

6.73.2 Constructor & Destructor Documentation

6.73.2.1 edu.ou.asgbook.segmentation.MultiscaleKMeans-Segmenter.MultiscaleKMeansSegmenter (int *thresh1*, int *thresh2*, int *K*)

Specify contouring levels.

```
                                                                             {
        this.THRESH1 = thresh1;
        this.THRESH2 = thresh2;
        this.K = K;
        this.INCR = (double)(THRESH2 - THRESH1) / K;
    }
```

6.73.3 Member Function Documentation

6.73.3.1 List<LabelResult> edu.ou.asgbook.segmentation.Multiscale-KMeansSegmenter.label (LatLonGrid *data*, File *out*)

Returns K scales of output.

```
                                                                  {
        List<LabelResult> result = new ArrayList<LabelResult>();
        LatLonGrid levels = quantize(data, out);
        for (int thresh=1; thresh <= K; ++thresh){
            ThresholdSegmenter seg = new ThresholdSegmenter(thresh);
            LabelResult label = seg.label(levels);
            if (out != null){
                try {
                    KmlWriter.write(label.label, out, "label_" + thresh, PngWriter.createCo
                } catch (Exception e) {
                    System.err.println(e);
                }
            }
            result.add(label);
        }
        return result;
    }
```

6.73.3.2 static void edu.ou.asgbook.segmentation.Multiscale-KMeansSegmenter.main (String[ ] *args*) throws Exception [static]

```
                                                                 {
        File out = OutputDirectory.getDefault("multiscalekmeans");

        // data
        LatLonGrid conus = SurfaceAlbedo.read(SurfaceAlbedo.CONUS, 100).crop(100, 100, 500, 200);
        KmlWriter.write(conus, out, "orig", PngWriter.createCoolToWarmColormap());

        new MultiscaleKMeansSegmenter(20,25,5).label(conus, out);
    }
```

6.73.3.3 LatLonGrid edu.ou.asgbook.segmentation.MultiscaleKMeansSegmenter.quantize (LatLonGrid *data*, File *out*)

Contours grid into levels 1,2,3.

..K

```
                                                         {
        final int nrows = data.getNumLat();
        final int ncols = data.getNumLon();

        // initialize based on simple quantization
        LatLonGrid seed = LatLonGrid.copyOf(data);
        for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
            int levelno = 0;
            if (data.getValue(i, j) != data.getMissing()){
                levelno = (int) Math.round((data.getValue(i,j) - THRESH1)/INCR);
                if ( levelno < 0 ) levelno = 0;
                else if ( levelno > K ) levelno = K;
            }
            seed.setValue(i,j, levelno);
        }
        if (out != null){
            try {
                KmlWriter.write(seed, out, "levels_0", PngWriter.createCoolToWarmColormap());
            } catch (Exception e) {
                System.err.println(e);
            }
        }

        // Start K-means
        int iter = 1;
        int n_changed = 0;
        do {
            // compute means: could get away with simply using center of data range ...
            Cluster[] clusters = findClusters(data, seed, K);
            // move pixels
            LatLonGrid next = LatLonGrid.copyOf(seed);
            n_changed = 0;
            for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
                if ( data.getValue(i,j) != data.getMissing() ){
                    int closest = findClosestCluster(data.getValue(i,j),i,j,seed, clusters);
                    if (closest != seed.getValue(i,j)){
                        // change the label to closest
                        next.setValue(i, j, closest);
                        ++n_changed;
                    }
                }
            }
            System.out.println("Changing " + n_changed + " at " + iter + " th iteration");
            // for next step
            seed = next;

            if (out != null){
                try {
                    KmlWriter.write(seed, out, "levels_" + iter, PngWriter.createCoolToWarr
                } catch (Exception e) {
                    System.err.println(e);
                }
            }

            ++iter;
        } while (iter < MAX_ITER && n_changed > 0);
        return seed;
    }
```

6.74 edu.ou.asgbook.filters.NHighest Class Reference

Finds the N highest valued-pixels in image.

Public Member Functions

- NHighest (int n)
- Pixel[ ] findHighestValued (LatLonGrid input)

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.74.1 Detailed Description

Finds the N highest valued-pixels in image.

Author:

Valliappa.Lakshmanan

6.74.2 Constructor & Destructor Documentation

6.74.2.1 edu.ou.asgbook.filters.NHighest.NHighest (int *n*)

```
                            {
        this.n = n;
    }
```

6.74.3 Member Function Documentation

6.74.3.1 Pixel [ ] edu.ou.asgbook.filters.NHighest.findHighestValued (LatLonGrid *input*)

```
                                                     {
        // create array of pixels
        int[][] data = input.getData();
        final int initialCapacity = (input.getNumLat() * input.getNumLon())  / 10;
        List<Pixel> a = new ArrayList<Pixel>(initialCapacity);
        for (int i=0; i < input.getNumLat(); ++i){
            for (int j=0; j < input.getNumLon(); ++j){
                if ( data[i][j] != input.getMissing() ){
                    a.add(new Pixel(i,j,data[i][j]));
                }
            }
        }
        System.out.println("Finding the " + n + " highest values out of " + a.size() + " pix

        // selection sort this array to find n highest
        Pixel[] result = new Pixel[n];
        Pixel.CompareValue comparator = new Pixel.CompareValue();
        for (int i=0; i < n; ++i){
            int p = i;
            for (int j=i; j < a.size(); ++j){
                if ( comparator.compare(a.get(j), a.get(p)) > 0 ){
                    p = j;
                }
            }
            result[i] = a.get(p);
            // swap a[i] and a[p]
            Pixel temp = a.get(i);
            a.set(i, a.get(p));
            a.set(p, temp);
        }
        return result;
    }
```

6.74.3.2 static void edu.ou.asgbook.filters.NHighest.main (String[ ] *args*) throws Exception `[static]`

```
                                                            {
        // read input
        LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA);
        popdensity.setMissing(0); // will get to process less data this way

        // find 10 highest
        NHighest filter = new NHighest(10);
        Pixel[] result = filter.findHighestValued(popdensity);
        for (int i=0; i < result.length; ++i){
            System.out.println(i + " " + result[i] + " loc=" + popdensity.getLocation(result
        }

        // plot the result on a map
        popdensity.fill(popdensity.getMissing());
        for (int i=0; i < result.length; ++i){
            popdensity.setValue(result[i].getX(), result[i].getY(), 1);
        }
        File out = OutputDirectory.getDefault("nhighest");
        KmlWriter.write(popdensity, out, "highest10", PngWriter.createHotColormap());

        // plot as KML points
        List<LatLon> points = new ArrayList<LatLon>();
        List<String> names = new ArrayList<String>();
        for (int i=0; i < result.length; ++i){
            Pixel p = result[i];
            points.add( popdensity.getLocation(p.getRow(), p.getCol()));
            names.add("Pixel#"+ (i+1) );
        }
        KmlWriter.write(points, names, out, "top10pixels");
```

```
    }
```

6.75 edu.ou.asgbook.filters.NHighestLevelSetImpl Class Reference

Finds the N highest valued-pixels in image using a levelset implementation.

Public Member Functions

- NHighestLevelSetImpl (int nth)
- Pixel[ ] findHighestValued (LatLonGrid input)

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.75.1 Detailed Description

Finds the N highest valued-pixels in image using a levelset implementation.

Author:

Valliappa.Lakshmanan

6.75.2 Constructor & Destructor Documentation

6.75.2.1 edu.ou.asgbook.filters.NHighestLevelSetImpl.NHighestLevelSetImpl (int *nth*)

```
                                    {
        this.nth = nth;
    }
```

6.75.3 Member Function Documentation

6.75.3.1 Pixel [ ] edu.ou.asgbook.filters.NHighestLevelSetImpl.findHighestValued (LatLonGrid *input*)

```
                                                 {
        // create level set
        LevelSet levelset = LevelSet.newInstance(input);

        // find the top n pixels
        Map.Entry<Integer, List<Pixel>>[] levels = levelset.getLevels();
        List<Pixel> result = new ArrayList<Pixel>();
        int curr = levels.length;
        while (result.size() < nth && curr > 0){
            --curr; // next
            result.addAll(levels[curr].getValue()); // all pixels at this level
        }

        return result.toArray(new Pixel[0]);
    }
```

6.75.3.2 static void edu.ou.asgbook.filters.NHighestLevelSetImpl.main (String[ ] *args*) throws Exception [static]

```
                                                                {
        // read input
        LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA);
        popdensity.setMissing(0); // will get to process less data this way

        // find 10 highest
        NHighestLevelSetImpl filter = new NHighestLevelSetImpl(10);
        Pixel[] result = filter.findHighestValued(popdensity);
        for (int i=0; i < result.length; ++i){
            System.out.println(i + " " + result[i] + " loc=" + popdensity.getLocation(result[i].getX(),
        }

        // plot the result on a map
        popdensity.fill(popdensity.getMissing());
        for (int i=0; i < result.length; ++i){
            popdensity.setValue(result[i].getX(), result[i].getY(), 1);
        }
        File out = OutputDirectory.getDefault("levelset");
        KmlWriter.write(popdensity, out, "highest10", PngWriter.createHotColormap());

        // plot as KML points
        List<LatLon> points = new ArrayList<LatLon>();
        List<String> names = new ArrayList<String>();
        for (int i=0; i < result.length; ++i){
            Pixel p = result[i];
            points.add( popdensity.getLocation(p.getRow(), p.getCol()));
            names.add("Pixel#"+ (i+1) );
        }
        KmlWriter.write(points, names, out, "top10pixels");
    }
```

6.76 edu.ou.asgbook.dataset.NightimeLights Class Reference

Reads night-time lights data in ESRI grid format.

Static Public Member Functions

- static LatLonGrid read (File file) throws IOException
- static void main (String[ ] args) throws Exception

Static Public Attributes

- static File WORLD = new File("data/nighttime/nighttimelights.txt.gz")

6.76.1 Detailed Description

Reads night-time lights data in ESRI grid format.

Author:

Valliappa.Lakshmanan

6.76.2 Member Function Documentation

6.76.2.1 static void edu.ou.asgbook.dataset.NightimeLights.main (String[ ] *args*) throws Exception [static]

```
                                                           {
           // create output directory
           File out = OutputDirectory.getDefault("nighttime");

           // read input
           LatLonGrid lights = NightimeLights.read(NightimeLights.WORLD);

           // write out as image, for viewing
           KmlWriter.write(lights, out, "lights", PngWriter.createCoolToWarmColormap());
       }
```

6.76.2.2 static LatLonGrid edu.ou.asgbook.dataset.NightimeLights.read (File *file*) throws IOException [static]

```
                                                               {
           return EsriGrid.read(file, new LinearScaling(100.0/63)); // 0-100
       }
```

6.76.3 Member Data Documentation

6.76.3.1 File edu.ou.asgbook.dataset.NightimeLights.WORLD = new File("data/nighttime/nighttimelights.txt.gz") `[static]`

6.77 edu.ou.asgbook.oban.ObjectiveAnalysisUtils Class Reference

Utility functions for objective analysis.

Static Public Member Functions

- static LatLonGrid createBoundingGrid (PointObservations data, double latres, double lonres)
- static double computeMeanDistance (PointObservations data)

6.77.1 Detailed Description

Utility functions for objective analysis.

Author:

Valliappa.Lakshmanan

6.77.2 Member Function Documentation

6.77.2.1 static double edu.ou.asgbook.oban.ObjectiveAnalysisUtils.computeMeanDistance (PointObservations *data*) [static]

```
                                                                 {
    PointObservations.ObservationPoint[] points = data.getPoints();
    if ( points.length < 1 ){
        throw new IllegalArgumentException("Number of points has be greater than one");
    }

    double totdist = 0;
    for (int i=0; i < points.length; ++i){
        double mindistsq = Double.MAX_VALUE;
        for (int j=0; j < points.length; ++j){
            if ( j != i ){
                double latdist = points[i].getLat() - points[j].getLat();
                double londist = points[i].getLon() - points[j].getLon();
                double distsq = (latdist*latdist + londist*londist);
                if ( distsq < mindistsq ){
                    mindistsq = distsq;
                }
            }
        }
        totdist += Math.sqrt(mindistsq);
    }

```

```
        return totdist / points.length;
    }
```

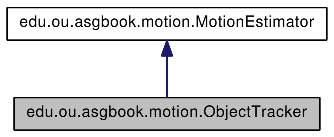
6.77.2.2 static LatLonGrid edu.ou.asgbook.oban.ObjectiveAnalysisUtils.createBoundingGrid (PointObservations *data*, double *latres*, double *lonres*) [static]

```
                                                                                                      {
        PointObservations.ObservationPoint[] points = data.getPoints();
        if ( points.length == 0 ){
            throw new IllegalArgumentException("Number of points has be greater than zero");
        }

        // find bounding box
        double minlat = 90;
        double maxlat = -90;
        double minlon = 180;
        double maxlon = -180;
        for (int i=0; i < points.length; ++i){
            if (points[i].getLat() > maxlat){
                maxlat = points[i].getLat();
            }
            if (points[i].getLat() < minlat){
                minlat = points[i].getLat();
            }
            if (points[i].getLon() > maxlon){
                maxlon = points[i].getLon();
            }
            if (points[i].getLon() < minlon){
                minlon = points[i].getLon();
            }
        }

        // go a little bit off to the side and roundoff so that grid bounds are multiples of res
        minlat = round(minlat - latres, latres);
        maxlat = round(maxlat + latres, latres);
        minlon = round(minlon - lonres, lonres);
        maxlon = round(maxlon + lonres, lonres);

        int nrows = (int) Math.round((maxlat - minlat)/latres);
        int ncols = (int) Math.round((maxlon - minlon)/lonres);

        System.out.println(points.length + " points will fit inside a " + nrows + "x" + ncols + " grid"
        return new LatLonGrid(nrows, ncols, data.getMissing(), new LatLon(maxlat,minlon), latres, lonre
    }
```

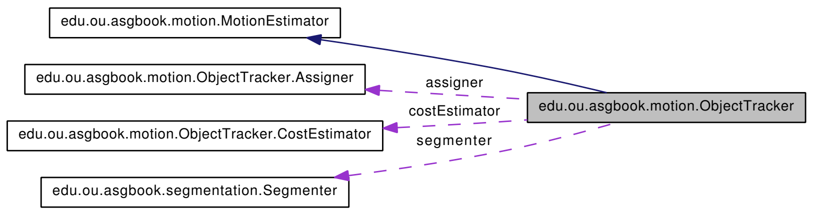
6.78 edu.ou.asgbook.motion.ObjectTracker Class Reference

Estimates motion based on assigning objects in one frame to objects in the previous frame.

Inheritance diagram for edu.ou.asgbook.motion.ObjectTracker:


edu.ou.asgbook.motion.MotionEstimator
edu.ou.asgbook.motion.ObjectTracker

Collaboration diagram for edu.ou.asgbook.motion.ObjectTracker:


edu.ou.asgbook.motion.MotionEstimator
edu.ou.asgbook.motion.ObjectTracker.Assigner
assigner
costEstimator
edu.ou.asgbook.motion.ObjectTracker
edu.ou.asgbook.motion.ObjectTracker.CostEstimator
segmenter
edu.ou.asgbook.segmentation.Segmenter

Public Member Functions

- ObjectTracker (int hysThresh1, int hysThresh2, int minsize)
- ObjectTracker (Segmenter seg, CostEstimator cost, Assigner a)
- Override Pair< LatLonGrid, LatLonGrid > compute (LatLonGrid data0, LatLonGrid data1, File outdir)

  returns motion in the two directions.

Static Public Member Functions

- static void main (String[ ] args) throws Exception

Classes

- interface Assigner
- class **CentroidDistance**

- interface CostEstimator
- class **GreedyAssigment**
- class **SimpleSegmenter**

6.78.1 Detailed Description

Estimates motion based on assigning objects in one frame to objects in the previous frame.

Author:

v.lakshmanan

6.78.2 Constructor & Destructor Documentation

6.78.2.1 edu.ou.asgbook.motion.ObjectTracker.ObjectTracker (int *hysThresh1*, int *hysThresh2*, int *minsize*)

```
                                                                     {
        this(new SimpleSegmenter(hysThresh1, hysThresh2, minsize),
                new CentroidDistance(), new GreedyAssigment());
    }
```

6.78.2.2 edu.ou.asgbook.motion.ObjectTracker.ObjectTracker (Segmenter *seg*, CostEstimator *cost*, Assigner *a*)

```
                                                                       {
        segmenter = seg;
        costEstimator = cost;
        assigner = a;
    }
```

6.78.3 Member Function Documentation

6.78.3.1 Override Pair<LatLonGrid, LatLonGrid> edu.ou.asgbook.motion.ObjectTracker.compute (LatLonGrid *data0*, LatLonGrid *data1*, File *outdir*)

returns motion in the two directions.

The first one is north to south and the second one is east to west. The data is aligned to second time frame. The output dir is used for intermediate products and may be null.

Implements edu.ou.asgbook.motion.MotionEstimator.

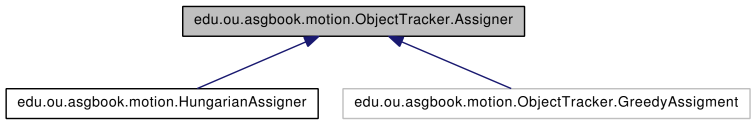
```

        LabelResult objects0 = segmenter.label(data0);
        LabelResult objects1 = segmenter.label(data1);

        if (outdir != null){
            try {
                KmlWriter.write(objects0.label, outdir, "objects0", PngWriter.createRandom
                KmlWriter.write(objects1.label, outdir, "objects1", PngWriter.createRandom
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

        // match the objects across frames
        RegionProperty[] regions0 = RegionProperty.compute(objects0, data0);
        RegionProperty[] regions1 = RegionProperty.compute(objects1, data1);
        int[][] cost = computeCost(regions0, regions1);
        int[] assigned = getAssignments(cost, outdir);

        // find motion for each region
        int[] regu = new int[assigned.length];
        int[] regv = new int[assigned.length];
        for (int i=1; i < assigned.length; ++i){
            int oldregno = assigned[i];
            if ( oldregno > 0 ){
                double cx = regions1[i].getCx();
                double cy = regions1[i].getCy();
                double oldcx = regions0[oldregno].getCx();
                double oldcy = regions0[oldregno].getCy();
                regu[i] = (int) Math.round( (cx - oldcx)*MOT_SCALE );
                regv[i] = (int) Math.round( (cy - oldcy)*MOT_SCALE );
                // System.out.println("Object at " + cx + "," + cy + " moving at " + regu[
            }
        }

        // apply the motion estimate based on assignment to all pixels
        LatLonGrid u = new LatLonGrid(data0.getNumLat(), data0.getNumLon(), 0, data0.getNw
        LatLonGrid v = LatLonGrid.copyOf(u);
        for (int i=0; i < u.getNumLat(); ++i) for (int j=0; j < u.getNumLon(); ++j){
            int regno = objects1.label.getValue(i,j);
            if ( regno > 0 ){
                u.setValue(i,j, regu[regno]);
                v.setValue(i,j, regv[regno]);
            }
        }

        return new Pair<LatLonGrid,LatLonGrid>(u,v);
    }
```

6.78.3.2 static void edu.ou.asgbook.motion.ObjectTracker.main (String[ ] *args*) throws Exception [static]

```
                                                        {
        // create output directory
        File out = OutputDirectory.getDefault("objecttracker");

        // read
        File f = new File("data/seviri");
        Pair<LatLonGrid,Date>[] grids = SeviriInfraredTemperature.readAll(f);

        // do alg
        MotionEstimator alg = new ObjectTracker(100, 110, 1000);
        MedianFilter smoother = new MedianFilter(10);
        LatLonGrid grid0 = smoother.filter(grids[0].first);
        LatLonGrid grid1 = smoother.filter(grids[1].first);
        Pair<LatLonGrid,LatLonGrid> motion = alg.compute(grid0, grid1, out);

        // write
        SaturateFilter filter = new SaturateFilter(-150, 150);
        LatLonGrid u = filter.filter(motion.first);
        LatLonGrid v = filter.filter(motion.second);
        KmlWriter.write(u, out, "closest_u", PngWriter.createCoolToWarmColormap());
        KmlWriter.write(v, out, "closest_v", PngWriter.createCoolToWarmColormap());
    }
```

6.79 edu.ou.asgbook.motion.ObjectTracker.Assigner Interface Reference

Inheritance diagram for edu.ou.asgbook.motion.ObjectTracker.Assigner:


edu.ou.asgbook.motion.ObjectTracker.Assigner
edu.ou.asgbook.motion.HungarianAssigner
edu.ou.asgbook.motion.ObjectTracker.GreedyAssigment

Public Member Functions

- int[ ] getAssignments (int[ ][ ] cost, int maxcost)

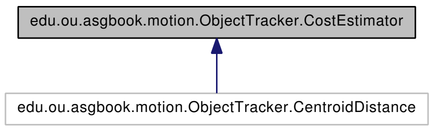
6.79.1 Member Function Documentation

6.79.1.1 int [ ] edu.ou.asgbook.motion.ObjectTracker.Assigner.getAssignments (int *cost*[ ][ ], int *maxcost*)

Implemented in edu.ou.asgbook.motion.HungarianAssigner.

6.80 edu.ou.asgbook.motion.ObjectTracker.CostEstimator Interface Reference

Inheritance diagram for edu.ou.asgbook.motion.ObjectTracker.CostEstimator:


edu.ou.asgbook.motion.ObjectTracker.CostEstimator
edu.ou.asgbook.motion.ObjectTracker.CentroidDistance

Public Member Functions

- int computeCost (RegionProperty a, RegionProperty b)
- int getMaxCost ()

6.80.1 Member Function Documentation

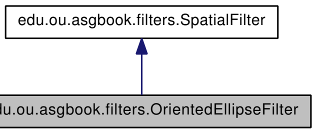
6.80.1.1 int edu.ou.asgbook.motion.ObjectTracker.CostEstimator.computeCost (RegionProperty *a*, RegionProperty *b*)

6.80.1.2 int edu.ou.asgbook.motion.ObjectTracker.CostEstimator.getMaxCost ()

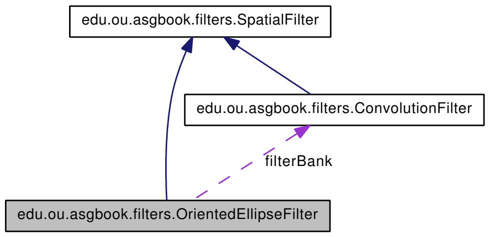
6.81 edu.ou.asgbook.filters.OrientedEllipseFilter Class Reference

A non-isotropic smoothing filter.

Inheritance diagram for edu.ou.asgbook.filters.OrientedEllipseFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.OrientedEllipseFilter

Collaboration diagram for edu.ou.asgbook.filters.OrientedEllipseFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.ConvolutionFilter
filterBank
edu.ou.asgbook.filters.OrientedEllipseFilter

Public Member Functions

- OrientedEllipseFilter (int numFilters, int a, int b)
- LatLonGrid smooth (LatLonGrid input, File out)

  Finds the maximum response of all the oriented filters.

- Override LatLonGrid filter (LatLonGrid input)

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.81.1 Detailed Description

A non-isotropic smoothing filter.

Author:

valliappa.lakshmanan

6.81.2 Constructor & Destructor Documentation

6.81.2.1 edu.ou.asgbook.filters.OrientedEllipseFilter.OrientedEllipseFilter (int *numFilters*, int *a*, int *b*)

```
                                                              {
        if (a == b){
            throw new IllegalArgumentException("For an ellipse, a != b");
        }
        filterBank = new ConvolutionFilter[numFilters];
        int size = Math.max(a,b) * 2 + 1;
        for (int f=0; f < numFilters; ++f){
            double[][] coeffs = new double[size][size];
            double theta = (f * Math.PI) / numFilters; // 0 to 180
            for (int i=0; i < size; ++i) for (int j=0; j < size; ++j){
                double x = i - a;
                double y = j - b;
                double term1 = x*Math.cos(theta) - y*Math.sin(theta);
                term1 = (term1*term1) / (a*a);
                double term2 = x*Math.sin(theta) + y*Math.cos(theta);
                term2 = (term2*term2) / (b*b);
                if ( (term1+term2) <= 1 ){
                    coeffs[i][j] = 1;
                } // else zero
            }
            filterBank[f] = new ConvolutionFilter(coeffs);
        }
    }
```

6.81.3 Member Function Documentation

6.81.3.1 Override LatLonGrid edu.ou.asgbook.filters.OrientedEllipse-Filter.filter (LatLonGrid *input*)

```
                                               {
        return smooth(input, null);
    }
```

6.81.3.2 static void edu.ou.asgbook.filters.OrientedEllipseFilter.main (String[ ] *args*) throws Exception [static]

```
                                                            {
        // create output directory
        File out = OutputDirectory.getDefault("oriented");

        // read input
        LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new GlobalPopulati
        KmlWriter.write(popdensity, out, "orig", PngWriter.createCoolToWarmColormap());

        OrientedEllipseFilter filter = new OrientedEllipseFilter(8, 1, 5);
        LatLonGrid sm = filter.smooth(popdensity, out);
        KmlWriter.write(sm, out, "ellipse", PngWriter.createCoolToWarmColormap());
    }
```

6.81.3.3 LatLonGrid edu.ou.asgbook.filters.OrientedEllipseFilter.smooth (LatLonGrid *input*, File *out*)

Finds the maximum response of all the oriented filters.

```
                                                        {
        LatLonGrid result = filterBank[0].smooth(input);
        KmlWriter.debugWrite(result, out, "ellipse0");
        for (int f=1; f < filterBank.length; ++f){
            LatLonGrid fth = filterBank[f].smooth(input);
            KmlWriter.debugWrite(fth, out, "ellipse"+f);
            int nrows = fth.getNumLat();
            int ncols = fth.getNumLon();
            for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
                int maxval = result.getValue(i,j);
                int fthval = fth.getValue(i,j);
                if (maxval == input.getMissing() ||
                    (fthval != input.getMissing() && fthval > maxval) ){
                    maxval = fthval;
                }
                result.setValue(i,j, maxval);
            }
        }
        return result;
    }
```

6.82 edu.ou.asgbook.histogram.OtsuThreshold-Selector Class Reference

Uses Otsu (1979) to select optimal threshold.

Public Member Functions

- OtsuThresholdSelector (Histogram hist)

  If x is returned, then values < x are one category and values >= x are another.

- int getOptimalThreshold ()

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.82.1 Detailed Description

Uses Otsu (1979) to select optimal threshold.

Author:

v.lakshmanan

6.82.2 Constructor & Destructor Documentation

6.82.2.1 edu.ou.asgbook.histogram.OtsuThresholdSelector.OtsuThreshold-Selector (Histogram *hist*)

If x is returned, then values < x are one category and values >= x are another.

```
                                                  {
           // compute p_i
           float[] prob = hist.calcProb();

           // mu_T
           float mu_T = 0;
           for (int i=0; i < hist.getHist().length; ++i){
               mu_T += (i+1) * prob[i];
           }

           // find k*
           var = new float[hist.getHist().length];
           int best_k = -1;
```

```
        float maxvar = 0;
        float w_k = 0;
        float mu_k = 0;
        for (int k=0; k < hist.getHist().length; ++k){
            w_k += prob[k];
            mu_k += (k+1) * prob[k];
            float denom = w_k * (1-w_k);
            float num = mu_T*w_k - mu_k;
            if ( denom > 0 ){
                var[k] = (num * num) / denom;
                // System.out.println(k + " " + var[k]);
                if ( var[k] > maxvar ){
                    maxvar = var[k];
                    best_k = k;
                }
            }
        }

        // return min value of (k+1)th bin
        optimalThreshold = (hist.getMin() + (best_k+1)* hist.getIncr());
    }
```

6.82.3 Member Function Documentation

6.82.3.1 int edu.ou.asgbook.histogram.OtsuThresholdSelector.getOptimal-Threshold ()

```
                                {
        return optimalThreshold;
    }
```

6.82.3.2 static void edu.ou.asgbook.histogram.OtsuThresholdSelector.main (String[ ] *args*) throws Exception [static]

```
                                                            {
        // create output directory
        File outdir = OutputDirectory.getDefault("otsu");

        // read input
        LatLonGrid conus = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA);

        // find threshold
        final int MIN = 0;
        final int MAX = 100;
        final int incr = 1;
        Histogram hist = new Histogram(MIN, incr, (MAX-MIN)/incr );
        hist.update(conus);
        // System.out.println(hist);
        OtsuThresholdSelector thresholder = new OtsuThresholdSelector(hist);
        int thresh = thresholder.optimalThreshold;
        System.out.println("Optimal threshold=" + thresh);

        // plot histogram and variance
        PrintWriter writer = new PrintWriter(new FileWriter(outdir + "/var.txt"));
        for (int i=0; i < hist.getHist().length; ++i){
            int val = (int) (0.5 + hist.getMin() + (i+0.5)*hist.getIncr());
            writer.println(val + " " + hist.getHist()[i] + " " + thresholder.var[i]);
        }
        writer.close();

        // threshold
        SimpleThresholder filter = new SimpleThresholder(thresh);
        LatLonGrid binaryImage = filter.threshold(conus);

        KmlWriter.write(binaryImage, outdir, "highpop", PngWriter.createCoolToWarmColormap());
    }
```

6.83 edu.ou.asgbook.io.OutputDirectory Class Reference

Change this to change the output directory that is used by all the main().

Static Public Member Functions

- static File temporary (String prefix) throws IOException
- static File relative (String prefix) throws IOException
- static File getDefault (String prefix) throws IOException
  Change this to change the output directory that is used by all the main().

6.83.1 Detailed Description

Change this to change the output directory that is used by all the main().

Author:

Valliappa.Lakshmanan

6.83.2 Member Function Documentation

6.83.2.1 static File edu.ou.asgbook.io.OutputDirectory.getDefault (String *prefix*) throws IOException [static]

Change this to change the output directory that is used by all the main().

```
                                                                    {
        // return temporary(prefix);
        return relative(prefix);
    }
```

6.83.2.2 static File edu.ou.asgbook.io.OutputDirectory.relative (String *prefix*) throws IOException [static]

```
                            {
        // current directory
        File out = new File("output/" + prefix + "_files");
        out.delete();
        out.mkdirs();
        System.out.println("Output will be in " + out);
        return out;
    }
```

6.83.2.3 static File edu.ou.asgbook.io.OutputDirectory.temporary (String *prefix*) throws IOException [static]

```
                              {

         File out = File.createTempFile(prefix, "_files");
         out.delete();
         out.mkdirs();
         System.out.println("Output will be in " + out);
         return out;

     }
```

6.84 edu.ou.asgbook.core.Pair< X, Y > Class Reference

An utility class so that methods can return two objects.

Public Member Functions

- Pair (X a, Y b)

Public Attributes

- final X first
- final Y second

6.84.1 Detailed Description

An utility class so that methods can return two objects.

Author:

v.lakshmanan

6.84.2 Constructor & Destructor Documentation

6.84.2.1 edu.ou.asgbook.core.Pair< X, Y >.Pair (X *a*, Y *b*)

```
                            {
            first = a;
            second = b;
    }
```

6.84.3 Member Data Documentation

6.84.3.1 final X edu.ou.asgbook.core.Pair< X, Y >.first

6.84.3.2 final Y edu.ou.asgbook.core.Pair< X, Y >.second

6.85 edu.ou.asgbook.motion.PhaseCorrelation Class Reference

Estimate motion based on FFT.

Public Member Functions

- PhaseCorrelation (int maxu, int maxv)
- Pair< Integer, Integer > compute (LatLonGrid data0, LatLonGrid data1)

Static Public Member Functions

- static Pixel computeCentroid (LatLonGrid a)
- static void test (File out) throws Exception
- static void main (String[ ] args) throws Exception

Package Attributes

- final int MAXU
- final int MAXV

6.85.1 Detailed Description

Estimate motion based on FFT.

Author:

v.lakshmanan

6.85.2 Constructor & Destructor Documentation

6.85.2.1 edu.ou.asgbook.motion.PhaseCorrelation.PhaseCorrelation (int *maxu*, int *maxv*)

```
                                                {
        this.MAXU = maxu;
        this.MAXV = maxv;
    }
```

6.85.3 Member Function Documentation

6.85.3.1 Pair<Integer,Integer> edu.ou.asgbook.motion.PhaseCorrelation.compute (LatLonGrid *data0*, LatLonGrid *data1*)

```
                                                                                  {
        int motNS = 0, motEW = 0;
        // a
        Complex[][] in1 = FFT2D.fft(FFT2D.zeropad(data0));

        // zero-out an area of thickness MAXU/MAXV around the boundary to avoid boundary iss
        LatLonGrid centerb = LatLonGrid.copyOf(data1);
        int minx = MAXU;
        int miny = MAXV;
        int maxx = centerb.getNumLat() - minx;
        int maxy = centerb.getNumLon() - miny;
        for (int i=0; i < data1.getNumLat(); ++i){
            for (int j=0; j < data1.getNumLon(); ++j){
                if (i < minx || j < miny || i > maxx || j > maxy){
                    centerb.setValue(i, j, 0);
                }
            }
        }
        Complex[][] in2 = FFT2D.fft(FFT2D.zeropad(centerb));

        // find phase shift at this point
        for (int i=0; i < in1.length; ++i) for (int j=0; j < in1[0].length; ++j){
            in1[i][j] = in1[i][j].multiply(in2[i][j].conjugate());
            in1[i][j] = in1[i][j].multiply( 1.0 / in1[i][j].norm() );
        }
        // take ifft
        Complex[][] result = FFT2D.ifft(in1);

        // find location at which the cross-power specturm is maximum
        double bestValue = Integer.MIN_VALUE;
        int startx = 0; // result.length/2 - MAXU;
        int starty = 0; // result[0].length/2 - MAXV;
        int endx = result.length; // /2 + MAXU;
        int endy = result[0].length; // /2 + MAXV;
        for (int i=startx; i < endx; ++i) for (int j=starty; j < endy; ++j){
            if ( result[i][j].normsq() > bestValue ){
                bestValue = result[i][j].real;
                motNS = -i;
                motEW = -j;
            }
        }

        // we don't want a 345-degree phase shift; we want it to be 15-degrees
        if ( Math.abs(motNS) > result.length/2 ){
            if (motNS < 0) motNS += result.length;
            else motNS -= result.length;
        }
        if ( Math.abs(motEW) > result[0].length/2 ){
            if (motEW < 0) motEW += result[0].length;
            else motEW -= result[0].length;
        }

        return new Pair<Integer,Integer>(motNS, motEW);
    }
```

6.85.3.2 static Pixel edu.ou.asgbook.motion.PhaseCorrelation.computeCentroid (LatLonGrid *a*) [static]

```
                                                      {
        double sumx = 0;
        double sumy = 0;
        double sumwt = 0;
        int N = 0;
        for (int i=0; i < a.getNumLat(); ++i) for (int j=0; j < a.getNumLon(); ++j){
            double wt = a.getValue(i,j);
            sumx += i * wt;
            sumy += j * wt;
            sumwt += wt;
             ++N;
        }
        return new Pixel((int)Math.round(sumx/sumwt), (int)Math.round(sumy/sumwt), (int)Math.round(sum
    }
```

6.85.3.3 static void edu.ou.asgbook.motion.PhaseCorrelation.main (String[ ] *args*) throws Exception [static]

```
                                                              {
        // create output directory
        File out = OutputDirectory.getDefault("phasecorr");
        test(out);

        // read
        File f = new File("data/seviri");
        Pair<LatLonGrid,Date>[] grids = SeviriInfraredTemperature.readAll(f);

        // do alg
        Pair<Integer,Integer> uv = new PhaseCorrelation(5, 5).compute(grids[0].first, grids[1].first);
        System.out.println("u=" + uv.first + " v=" + uv.second);
    }
```

6.85.3.4 static void edu.ou.asgbook.motion.PhaseCorrelation.test (File *out*) throws Exception [static]

```
                                                          {
        // because the alignment doesn't really check lat-lon extents,
        // cropping from offset corners will look like translation ...
        LatLonGrid conus = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA);
        LatLonGrid[] grids = new LatLonGrid[2];
        grids[0] = conus.crop(900, 2500, 256, 256);
```

```
            int motx = 5; int moty = 9;
            grids[1] = conus.crop(900-motx, 2500-moty, 256, 256);

            // do alg
            Pair<Integer,Integer> motion = new PhaseCorrelation(30,30).compute(grids[0], grids
            System.out.println("Motion N/S ="  + motion.first + " true N/S=" + motx);
            System.out.println("Motion E/W ="  + motion.second + " true E/W=" + moty);

            System.out.println("Centroid of first = " + computeCentroid(grids[0]));
            System.out.println("Centroid of second = " + computeCentroid(grids[1]));

            KmlWriter.write(grids[0], out, "popdensityA", PngWriter.createCoolToWarmColormap()
            KmlWriter.write(grids[1], out, "popdensityB", PngWriter.createCoolToWarmColormap()

            // based on edges alone
            SobelEdgeFilter edgeFilter = new SobelEdgeFilter();
            LatLonGrid edge1 = edgeFilter.edgeFilter(grids[0]);
            LatLonGrid edge2 = edgeFilter.edgeFilter(grids[1]);
            motion = new PhaseCorrelation(30,30).compute(edge1, edge2);
            System.out.println("Motion N/S ="  + motion.first + " true N/S=" + motx);
            System.out.println("Motion E/W ="  + motion.second + " true E/W=" + moty);
        }
```

6.85.4 Member Data Documentation

6.85.4.1 final int edu.ou.asgbook.motion.PhaseCorrelation.MAXU

[package]

6.85.4.2 final int edu.ou.asgbook.motion.PhaseCorrelation.MAXV

[package]

6.86 edu.ou.asgbook.core.Pixel Class Reference

A grid point in a spatial grid consists of a location and value.

Public Member Functions

- int getX ()
- int getY ()
- int getRow ()
- int getCol ()
- int getValue ()
- Pixel (int x, int y, int value)
- Override boolean equals (Object o)
- int getDistanceSquared (Pixel other)
- int getDistanceSquared (int otherx, int othery)
- Override int compareTo (Pixel other)

    Compares both location and value.

- Override String toString ()

Static Public Member Functions

- static void main (String[ ] args)

Classes

- class **CompareLocation**
- class **CompareValue**

6.86.1 Detailed Description

A grid point in a spatial grid consists of a location and value.

Author:

Valliappa.Lakshmanan

6.86.2 Constructor & Destructor Documentation

6.86.2.1 edu.ou.asgbook.core.Pixel.Pixel (int *x*, int *y*, int *value*)

```
                                        {
        super();
        this.x = x;
        this.y = y;
        this.value = value;
    }
```

6.86.3 Member Function Documentation

6.86.3.1 Override int edu.ou.asgbook.core.Pixel.compareTo (Pixel *other*)

Compares both location and value.

To compare only based on location or based on value

See also:

CompareValue
CompareLocation

```
                                    {
        if ( other == null ){
            return 1;
        }
        if ( other.value == value ){
            if ( other.x == x ){
                return (y - other.y);
            } else {
                return (x - other.x);
            }
        } else {
            return value - other.value;
        }
    }
```

6.86.3.2 Override boolean edu.ou.asgbook.core.Pixel.equals (Object *o*)

```
                                   {
        if (o == this){
            return true;
        }
        if (o == null || !o.getClass().equals(this.getClass())){
            return false;
        }
        Pixel other = (Pixel) o;
        return (other.x == x && other.y == y && other.value == value);
    }
```

6.86.3.3 int edu.ou.asgbook.core.Pixel.getCol ()

```
                        {
        return y;
    }
```

6.86.3.4 int edu.ou.asgbook.core.Pixel.getDistanceSquared (int *otherx*, int *othery*)

```
                                                        {
        int distx = this.getX() - otherx;
        int disty = this.getY() - othery;
        return (distx*distx) + (disty*disty);
    }
```

6.86.3.5 int edu.ou.asgbook.core.Pixel.getDistanceSquared (Pixel *other*)

```
                                             {
        return getDistanceSquared(other.x, other.y);
    }
```

6.86.3.6 int edu.ou.asgbook.core.Pixel.getRow ()

```
                        {
        return x;
    }
```

6.86.3.7 int edu.ou.asgbook.core.Pixel.getValue ()

```
                          {
        return value;
    }
```

6.86.3.8 int edu.ou.asgbook.core.Pixel.getX ()

```
                      {
        return x;
    }
```

6.86.3.9 int edu.ou.asgbook.core.Pixel.getY ()

```
                    {
        return y;
    }
```

6.86.3.10 static void edu.ou.asgbook.core.Pixel.main (String[ ] *args*) [static]

```
                                        {
        Pixel a = new Pixel(3,4,3);
        System.out.println("should be +ve: " + new CompareValue().compare(a,new Pixel(-1,-1
        System.out.println("should be -ve: " + new CompareValue().compare(a,new Pixel(-1,-1
        System.out.println("should be  0: " + new CompareValue().compare(a,new Pixel(-1,-1,

        System.out.println("should be +ve: " + new CompareLocation().compare(a,new Pixel(2,
        System.out.println("should be -ve: " + new CompareLocation().compare(a,new Pixel(5,
        System.out.println("should be  0: " + new CompareLocation().compare(a,new Pixel(3,4

        System.out.println("should be +ve: " + a.compareTo(new Pixel(2,3,1)));
        System.out.println("should be -ve: " + a.compareTo(new Pixel(3,4,11)));
        System.out.println("should be  0: " + a.compareTo(new Pixel(3,4,3)));

        System.out.println("Dist = " + a.getDistanceSquared(new Pixel(4,5,6)));
    }
```

6.86.3.11 Override String edu.ou.asgbook.core.Pixel.toString ()

```
                          {
        return new StringBuilder().append("[").append(x).append(",").append(y).append(" ")
    }
```

6.87 edu.ou.asgbook.io.PngWriter Class Reference

Writes a spatial grid out as PNG file.

Static Public Member Functions

- static void writeAutoScaled (LatLonGrid grid, File outputFile, ColorModel colormap) throws Exception
- static IndexColorModel createGrayScaleColormap ()

  black-to-white colormap

- static IndexColorModel createHotColormap ()

  a colormap that goes from blue to red through purple

- static IndexColorModel createCoolToWarmColormap ()

  a colormap that goes from green to red through white.

- static IndexColorModel createRandomColormap ()

  Randomized colormap, useful for displaying object labels, for example where the datavalues themselves do not mean anything beyond being a means to distinguish between objects.

- static void writeScaled (LatLonGrid grid, File outputFile, int min, int max, ColorModel colormap) throws Exception
- static void main (String[ ] args) throws Exception

Static Package Attributes

- static final byte DEFAULT_TRANSPARENCY = (byte) 200

6.87.1 Detailed Description

Writes a spatial grid out as PNG file.

Author:

Valliappa.Lakshmanan

6.87.2 Member Function Documentation

6.87.2.1 static IndexColorModel edu.ou.asgbook.io.PngWriter.createCoolToWarmColormap () [static]

a colormap that goes from green to red through white.

See Candidate2 in http://www.paraview.org/ParaView3/index.php/Default_Color_Map Adapted from the work of Cindy Brewer for use in ParaView

```
                                                            {
        byte[] red = new byte[256];
        byte[] green = new byte[red.length];
        byte[] blue = new byte[red.length];
        byte[] alpha = new byte[red.length];

        interpolate(red, green, blue, 0,  25, 0.0196078, 0.188235, 0.380392, 0.129412, 0.4,
        interpolate(red, green, blue, 25, 51, 0.129412, 0.4, 0.67451, 0.262745, 0.576471, 
        interpolate(red, green, blue, 51, 76, 0.262745, 0.576471, 0.764706, 0.572549, 0.77
        interpolate(red, green, blue, 76, 102, 0.572549, 0.772549, 0.870588, 0.819608, 0.8
        interpolate(red, green, blue, 102, 127, 0.819608, 0.898039, 0.941176, 0.968627, 0.
        interpolate(red, green, blue, 127, 153, 0.968627, 0.968627, 0.968627, 0.992157, 0.
        interpolate(red, green, blue, 153, 178, 0.992157, 0.858824, 0.780392, 0.956863, 0.
        interpolate(red, green, blue, 178, 204, 0.956863, 0.647059, 0.509804, 0.839216, 0.
        interpolate(red, green, blue, 204, 229, 0.839216, 0.376471, 0.301961, 0.698039, 0.
        interpolate(red, green, blue, 229, 256, 0.698039, 0.0941176, 0.168627, 0.403922, 0,

        alpha[0] = 0;
        for (int i=1; i < alpha.length; ++i){
            alpha[i] = DEFAULT_TRANSPARENCY;
        }

        IndexColorModel colormap = new IndexColorModel(16, red.length, red, green, blue, a
        return colormap;
    }
```

6.87.2.2 static IndexColorModel edu.ou.asgbook.io.PngWriter.createGrayScaleColormap () [static]

black-to-white colormap

```
                                                       {
        byte[] red = new byte[256];
        byte[] alpha = new byte[red.length];
        for (int i=0; i < red.length; ++i){
            red[i] = (byte) i;
        }
        alpha[0] = 0;
        for (int i=1; i < alpha.length; ++i){
            alpha[i] = DEFAULT_TRANSPARENCY;
        }
        IndexColorModel colormap = new IndexColorModel(16, red.length, red, red, red, alpha)
        return colormap;
    }
```

6.87.2.3 static IndexColorModel edu.ou.asgbook.io.PngWriter.createHotColormap () [static]

a colormap that goes from blue to red through purple

```
                                                    {
        byte[] red = new byte[256];
        byte[] green = new byte[red.length];
        byte[] blue = new byte[red.length];
        byte[] alpha = new byte[red.length];
        for (int i=0; i < red.length; ++i){
            red[i] = (byte) i;
            blue[i] = (byte)( 255 - red[i]);
            green[i] = (byte)( (red[i] + blue[i])/2 );
        }
        alpha[0] = 0;
        for (int i=1; i < alpha.length; ++i){
            alpha[i] = DEFAULT_TRANSPARENCY;
        }
        IndexColorModel colormap = new IndexColorModel(16, red.length, red, green, blue, alpha);
        return colormap;
    }
```

6.87.2.4 static IndexColorModel edu.ou.asgbook.io.PngWriter.createRandomColormap () [static]

Randomized colormap, useful for displaying object labels, for example where the datavalues themselves do not mean anything beyond being a means to distinguish between objects.

```
                                                    {
        byte[] red = new byte[256];
        byte[] green = new byte[red.length];
        byte[] blue = new byte[red.length];
        byte[] alpha = new byte[red.length];

        Random rnd = new Random();

        // random colors for the three channels
        for (int i=0; i < red.length; ++i){
            red[i] = (byte) rnd.nextInt(255);
            green[i] = (byte) rnd.nextInt(255);
            blue[i] = (byte) rnd.nextInt(255);
        }

        // 0 is transparent; everything else is opaque
        alpha[0] = 0;
        for (int i=1; i < alpha.length; ++i){
            alpha[i] = DEFAULT_TRANSPARENCY;
        }

        IndexColorModel colormap = new IndexColorModel(16, red.length, red, green, blue, al
        return colormap;
    }
```

6.87.2.5 static void edu.ou.asgbook.io.PngWriter.main (String[ ] *args*) throws Exception [static]

```
                                                                    {
        LatLonGrid grid = new LatLonGrid(100, 200, -1, new LatLon(35,-97), 0.1, 0.1);
        for (int i=0; i < grid.getNumLat(); ++i){
            for (int j=0; j < grid.getNumLon(); ++j){
                grid.getData()[i][j] = i + j;
                if ( i%10 == 0 || j%20 == 0){
                    grid.getData()[i][j] = grid.getMissing();
                }
            }
        }

        File outdir = OutputDirectory.getDefault("pngwriter");
        File out = new File(outdir.getAbsoluteFile() + "/autoscaled_cooltowarm.png");
        PngWriter.writeAutoScaled(grid, out, PngWriter.createCoolToWarmColormap());

        out = new File(outdir.getAbsoluteFile() + "/autoscaled_hot.png");
        PngWriter.writeAutoScaled(grid, out, PngWriter.createHotColormap());

        out = new File(outdir.getAbsoluteFile() + "/autoscaled_gray.png");
        PngWriter.writeAutoScaled(grid, out, PngWriter.createGrayScaleColormap());

        out = new File(outdir.getAbsoluteFile() + "/scaled_0_10.png");
        PngWriter.writeScaled(grid, out, 10, 100, PngWriter.createCoolToWarmColormap());
    }
```

6.87.2.6 static void edu.ou.asgbook.io.PngWriter.writeAutoScaled (LatLonGrid *grid*, File *outputFile*, ColorModel *colormap*) throws Exception [static]

```

        // find min, max in data
        int[][] data = grid.getData();
        int min = Integer.MAX_VALUE;
        int max = Integer.MIN_VALUE;
        int numvalid = 0;
        for (int i=0; i < data.length; ++i){
            for (int j=0; j < data[0].length; ++j){
                if ( data[i][j] != grid.getMissing() ){
                    ++numvalid;
                    if ( data[i][j] < min ){
                        min = data[i][j];
                    }
                    if ( data[i][j] > max ){
                        max = data[i][j];
                    }
                }
            }
        }
        System.out.println("Autoscaling " + numvalid + " valid pixels between " + min + " and " + max);
        writeScaled(grid, outputFile, min, max, colormap);
    }
```

6.87.2.7 static void edu.ou.asgbook.io.PngWriter.writeScaled (LatLonGrid *grid*, File *outputFile*, int *min*, int *max*, ColorModel *colormap*) throws Exception `[static]`

```

        // scale the data and lookup the color
        int[][] data = grid.getData();
        double scale = 255.0 / (max - min + 1); // first is for 'missing'
        BufferedImage result = new BufferedImage(grid.getNumLon(), grid.getNumLat(), BufferedImage.TYP
        for (int i=0; i < data.length; ++i){
            for (int j=0; j < data[0].length; ++j){
                int scaled = 0;
                if ( data[i][j] == grid.getMissing() ){
                    scaled = 0;
                } else if ( data[i][j] < min ){
                    // System.out.println(data[i][j] + " " + scaled);
                    scaled = 0;
                } else if ( data[i][j] >= max ){
                    scaled = 255;
                } else {
                    scaled = (int) ( (data[i][j]-min) * scale + 1.5);
                }

                result.setRGB(j, i, colormap.getRGB(scaled));
            }
        }

        // write it out
        ImageIO.write(result, "png", outputFile);
        System.out.println("Wrote " + outputFile + " by scaling data between " + min + " and " + max);
    }
```

6.87.3 Member Data Documentation

6.87.3.1 final byte edu.ou.asgbook.io.PngWriter.DEFAULT_TRANSPARENCY = (byte) 200 `[static, package]`

6.88 edu.ou.asgbook.core.PointObservations Class Reference

A set of observation points.

Public Member Functions

- PointObservations (ObservationPoint[ ] points, int missing)
- int getMaxValue ()
- ObservationPoint[ ] getPoints ()
- int getMissing ()

Classes

- class **ObservationPoint**

  An observation point is a value at a given location.

6.88.1 Detailed Description

A set of observation points.

Author:

valliappa.lakshmanan

6.88.2 Constructor & Destructor Documentation

6.88.2.1 edu.ou.asgbook.core.PointObservations.PointObservations (ObservationPoint[ ] *points*, int *missing*)

```
                                                                      {
        this.points = points;
        this.missing = missing;
    }
```

6.88.3 Member Function Documentation

6.88.3.1 int edu.ou.asgbook.core.PointObservations.getMaxValue ()

```
                            {
        int result = missing;
        for (int i=0; i < points.length; ++i){
            if ( result == missing || points[i].getValue() > result ){
                result = points[i].getValue();
            }
        }
        return result;
    }
```

6.88.3.2 int edu.ou.asgbook.core.PointObservations.getMissing ()

```
                            {
        return missing;
    }
```

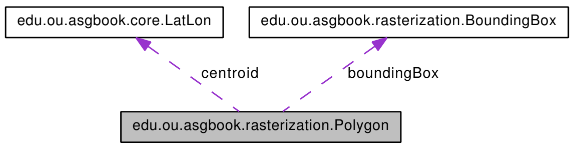
6.88.3.3 ObservationPoint [ ] edu.ou.asgbook.core.PointObservations.getPoints ()

```
                                          {
        return points;
    }
```

6.89 edu.ou.asgbook.rasterization.Polygon Class Reference

A polygon consisting of straight edges along the earth's surface.

Collaboration diagram for edu.ou.asgbook.rasterization.Polygon:


edu.ou.asgbook.core.LatLon
edu.ou.asgbook.rasterization.BoundingBox
centroid
boundingBox
edu.ou.asgbook.rasterization.Polygon

Public Member Functions

- BoundingBox getBoundingBox ()
- Polygon (LatLon[ ] vertices)
- List< Line > getEdges ()
- LatLon getCentroid ()
- double getArea ()

  The area is in degrees^2, so not very useful unless you can convert to km^2.

- boolean contains (double x, double y)

  Workhorse method: finds out if this point is within this polygon.

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.89.1 Detailed Description

A polygon consisting of straight edges along the earth's surface.

Author:

valliappa.lakshmanan

6.89.2 Constructor & Destructor Documentation

6.89.2.1 edu.ou.asgbook.rasterization.Polygon.Polygon (LatLon[ ] *vertices*)

```
                                      {
        if (vertices.length < 3){
            throw new IllegalArgumentException("Need atleast 3 vertices for polygon.");
        }
        for (int i=0; i < vertices.length-1; ++i){
            Line line = new Line( vertices[i].getLat(), vertices[i].getLon(),
                    vertices[i+1].getLat(), vertices[i+1].getLon() );
            edges.add(line);
        }

        // connect start and end point
        LatLon last = vertices[vertices.length-1];
        LatLon first = vertices[0];
        edges.add(new Line(last.getLat(),last.getLon(),first.getLat(),first.getLon()));
        computeAreaAndCentroid();
        boundingBox = new BoundingBox(vertices);
    }
```

6.89.3 Member Function Documentation

6.89.3.1 boolean edu.ou.asgbook.rasterization.Polygon.contains (double *x*, double *y*)

Workhorse method: finds out if this point is within this polygon.

```
                                                {
        // as an optimization, check the bounding box first
        if (!boundingBox.contains(x,y)){
            return false;
        }

        int num_xcrossing = 0;
        int num_ycrossing = 0;
        for (int i = 0; i < edges.size(); ++i) {
            Double x_intercept = edges.get(i).getXIntercept(y);
            Double y_intercept = edges.get(i).getYIntercept(x);
            if (y_intercept != null) {
                if (y_intercept >= y) {
                    ++num_ycrossing;
                }
            }
            if (x_intercept != null) {
                if (x_intercept >= x) {
                    ++num_xcrossing;
                }
            }
        }
        // odd number of crossings means inside
        return ((num_xcrossing % 2 == 1) && (num_ycrossing % 2 == 1));
```

```
    }
```

6.89.3.2 double edu.ou.asgbook.rasterization.Polygon.getArea ()

The area is in degrees^2, so not very useful unless you can convert to km^2.

```
                            {
        return area;
    }
```

6.89.3.3 BoundingBox edu.ou.asgbook.rasterization.Polygon.getBoundingBox ()

```
                                        {
        return boundingBox;
    }
```

6.89.3.4 LatLon edu.ou.asgbook.rasterization.Polygon.getCentroid ()

```
                                {
        return centroid;
    }
```

6.89.3.5 List<Line> edu.ou.asgbook.rasterization.Polygon.getEdges ()

```
                                 {
        return edges;
    }
```

6.89.3.6 static void edu.ou.asgbook.rasterization.Polygon.main (String[ ] *args*) throws Exception [static]

```
                                                             {
        File out = OutputDirectory.getDefault("rasterpolygon");

        // made up
        LatLonGrid grid = new LatLonGrid(500, 500, 0, new LatLon(20, -10), 0.01, 0.01);
        LatLon[] vertices = new LatLon[]{
                new LatLon(19,-7),
                new LatLon(17.5,-6),
                new LatLon(16.5,-6.8),
                new LatLon(17.2,-8.5),
                new LatLon(16,-9.5),
                new LatLon(17,-9)
        };
        Polygon poly = new Polygon(vertices);

        // draw edges
        final int EDGE = 10;
        for (Line line : poly.getEdges()){
            for (Pixel p : line.getPositionIn(grid)){
                grid.setValue(p.getRow(), p.getCol(), EDGE);
            }
        }
        KmlWriter.write(grid, out, "edges", PngWriter.createCoolToWarmColormap());


        // fill points inside
        final int POLY = 5;
        int npix = 0;
        for (int i=0; i < grid.getNumLat(); ++i){
            for (int j=0; j < grid.getNumLon(); ++j){
                LatLon loc = grid.getLocation(i, j);
                if ( poly.contains(loc.getLat(), loc.getLon())){
                    grid.setValue(i,j, POLY);
                    ++npix;
                }
            }
        }
        KmlWriter.write(grid, out, "polygon", PngWriter.createCoolToWarmColormap());

        System.out.println("Area of polygon: " + poly.getArea() + " num-pixels colored=" + npix);
        System.out.println("Centroid of polygon: " + poly.getCentroid());
    }
```

6.90 edu.ou.asgbook.datamining.PrimaryCities Class Reference

Identifies the primary cities in each country.

Static Public Member Functions

- static LabelResult findPrimaryCities (LatLonGrid population, LatLonGrid countries, File out)
- static void main (String[ ] args) throws Exception

6.90.1 Detailed Description

Identifies the primary cities in each country.

Author:

valliappa.lakshmanan

6.90.2 Member Function Documentation

6.90.2.1 static LabelResult edu.ou.asgbook.datamining.PrimaryCities.findPrimaryCities (LatLonGrid *population*, LatLonGrid *countries*, File *out*) [static]

```

        // find cities from population data using watershed
        write(out, population, "pop", PngWriter.createCoolToWarmColormap());
        EnhancedWatershedSegmenter seg = new EnhancedWatershedSegmenter(10, 1, 600, 10, 5);
        LabelResult label = seg.label(population);
        RegionProperty[] popProps = RegionProperty.compute(label, population);
        write(out, label.label, "allcities", PngWriter.createRandomColormap());

        // initialize primary-cities
        int ncountries = 1 + new MaxValueFilter().findHighestValued(countries).value;
        int[] primaryCity = new int[ncountries]; // one for each country
        for (int i=0; i < ncountries; ++i){
            primaryCity[i] = -1; // none
        }

        // go through the cities and assign them to their appropriate country
        for (int i=1; i < popProps.length; ++i){
            LatLon centroid = population.getLocation(popProps[i].getCx(), popProps[i].getCy
            int country = countries.getValue(centroid);
            if (country >= 0){
                if (primaryCity[country] < 0){
                    primaryCity[country] = i; // first city in country
                } else {
                    // the primary city is the one with the greater avg population
                    int previous = primaryCity[country];
                    if (popProps[i].getCval() > popProps[previous].getCval()){
                        primaryCity[country] = i;
                    }
                }
            }
        }

        // keep only those cities that are primary
        boolean[] keep = new boolean[popProps.length];
        for (int i=0; i < ncountries; ++i){
            if (primaryCity[i] >= 0){
                int regno = primaryCity[i];
                keep[regno] = true;
            }
        }
        return RegionProperty.prune(label, keep);
    }
```

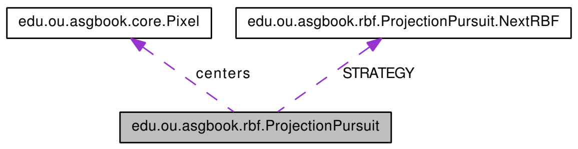
6.90.2.2 static void edu.ou.asgbook.datamining.PrimaryCities.main (String[ ] *args*) throws Exception [static]

```
                                                             {
        // create output directory
        File out = OutputDirectory.getDefault("primary");

        // read input (crop to cover Spain)
        LatLonGrid pop    = GlobalPopulation.read(GlobalPopulation.WORLD).crop(980, 4080, 220, 350);
        LatLonGrid countries = CountryPolygons.readGrid(CountryPolygons.WORLD_GRID);
        LabelResult primary = PrimaryCities.findPrimaryCities(pop, countries, out);

        KmlWriter.write(primary.label, out, "primarycities", PngWriter.createRandomColormap());
    }
```

6.91 edu.ou.asgbook.rbf.ProjectionPursuit Class Reference

Approximates a spatial grid by a RBF when nothing is known beyond the number of Gaussians desired.

Collaboration diagram for edu.ou.asgbook.rbf.ProjectionPursuit:


edu.ou.asgbook.core.Pixel
edu.ou.asgbook.rbf.ProjectionPursuit.NextRBF
centers
STRATEGY
edu.ou.asgbook.rbf.ProjectionPursuit

Public Member Functions

- ProjectionPursuit (LatLonGrid orig, int max_tot_abs_error, int max_number_-rbfs, File outDir)
- Pixel[ ] getCenters ()
- double[ ] getSigmax ()
- double[ ] getSigmay ()
- String toString ()

Static Public Member Functions

- static void runOnSimulatedInput () throws Exception
- static void runOnPopDensity (boolean crop) throws Exception
- static void main (String[ ] args) throws Exception

Static Public Attributes

- static final NextRBF STRATEGY = new LocalMax()

Classes

- class **LocalMax**
- interface NextRBF
- class **SpatialMean**

6.91.1 Detailed Description

Approximates a spatial grid by a RBF when nothing is known beyond the number of Gaussians desired.

Author:

v.lakshmanan

6.91.2 Constructor & Destructor Documentation

6.91.2.1 edu.ou.asgbook.rbf.ProjectionPursuit.ProjectionPursuit (LatLonGrid *orig*, int *max_tot_abs_error*, int *max_number_rbfs*, File *outDir*)

```
                                                                                                           {
        this.MAX_TOT_ABS_ERROR = max_tot_abs_error;
        this.MAX_NUMBER_RBFS = max_number_rbfs;
        if ( MAX_NUMBER_RBFS < 10 ){
            outputInterval = 1;
        } else if ( MAX_NUMBER_RBFS < 50){
            outputInterval = 5;
        } else {
            outputInterval = 10;
        }
        this.outDir = outDir;
        fit(orig, STRATEGY);
    }
```

6.91.3 Member Function Documentation

6.91.3.1 Pixel [ ] edu.ou.asgbook.rbf.ProjectionPursuit.getCenters ()

```
                                {
        return centers;
    }
```

6.91.3.2 double [ ] edu.ou.asgbook.rbf.ProjectionPursuit.getSigmax ()

```
                                {
        return sigmax;
    }
```

6.91.3.3 double [ ] edu.ou.asgbook.rbf.ProjectionPursuit.getSigmay ()

```
                                {
        return sigmay;
    }
```

6.91.3.4 static void edu.ou.asgbook.rbf.ProjectionPursuit.main (String[ ] *args*) throws Exception [static]

```
                                                           {
        // runOnSimulatedInput();
        runOnPopDensity(false); // run with -Xmx1024m otherwise, you'll get out-of-memory e
    }
```

6.91.3.5 static void edu.ou.asgbook.rbf.ProjectionPursuit.runOnPopDensity (boolean *crop*) throws Exception [static]

```
                                                                       {
        LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new Gl
        if (crop){
            popdensity = popdensity.crop(900, 2500, 200, 200);
        } else {
            LatLon nwCorner = new LatLon(60, -130);
            LatLon seCorner = new LatLon(12, -52);
            popdensity = popdensity.crop(popdensity.getRow(nwCorner),
                    popdensity.getCol(nwCorner),
                    popdensity.getRow(seCorner) - popdensity.getRow(nwCorner),
                    popdensity.getCol(seCorner) - popdensity.getCol(nwCorner));
        }

        File out = OutputDirectory.getDefault("rbfpopdensity");

        ProjectionPursuit fit = new ProjectionPursuit(popdensity, 1000, 9, out);
        List<LatLon> locs = new ArrayList<LatLon>();
        List<String> names = new ArrayList<String>();
        for (int i=0; i < fit.getCenters().length; ++i){
            LatLon loc = popdensity.getLocation( fit.getCenters()[i].getX(), fit.getCenters
            String name = ("RBF#" + i + " ampl=" +  fit.getCenters()[i].getValue() + " sigm
            System.out.println(" loc: " + loc + name);
            if (fit.getCenters()[i].getValue() > 0){
                locs.add(loc);
                names.add(name);
            }
        }
        KmlWriter.write(locs, names, out, "rbfcities");
    }
```

6.91.3.6 static void edu.ou.asgbook.rbf.ProjectionPursuit.runOnSimulated-Input () throws Exception [static]

```
                                                            {
        int nrows = 100;
        int ncols = 100;
        Pixel[] centers = new Pixel[]{ new Pixel(nrows/4,ncols/3,20), new Pixel(nrows/3,nco
        double[] sigmax = new double[] { nrows/12, ncols/8 };
        double[] sigmay = new double[] { nrows/8, ncols/12 };
        LatLonGrid m = DataSimulator.simulateData(centers, sigmax, sigmay, nrows, ncols);
        System.out.println("Created data of size " + m.getNumLat() + "x" + m.getNumLon());
        for (int i=0; i < centers.length; ++i){
            System.out.println("true RBF#" + i + " center: " + centers[i] + " sigmax=" + sigmax[i] + "
        }

        File out = OutputDirectory.getDefault("rbf");

        ProjectionPursuit fit = new ProjectionPursuit(m, 100, 4, out);
        System.out.println(fit);
    }
```

6.91.3.7 String edu.ou.asgbook.rbf.ProjectionPursuit.toString ()

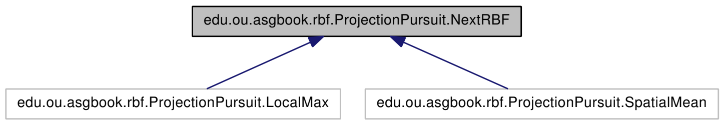
```
                            {
        StringBuilder sb = new StringBuilder();
        for (int i=0; i < centers.length; ++i){
            sb.append("RBF#" + i + " center: " + centers[i] + " sigmax=" + sigmax[i] + " sigmay=" + sig
        }
        return sb.toString();
    }
```

6.91.4 Member Data Documentation

6.91.4.1 final NextRBF edu.ou.asgbook.rbf.ProjectionPursuit.STRATEGY = new LocalMax() [static]

6.92 edu.ou.asgbook.rbf.ProjectionPursuit.NextRBF Interface Reference

Inheritance diagram for edu.ou.asgbook.rbf.ProjectionPursuit.NextRBF:


edu.ou.asgbook.rbf.ProjectionPursuit.NextRBF
edu.ou.asgbook.rbf.ProjectionPursuit.LocalMax
edu.ou.asgbook.rbf.ProjectionPursuit.SpatialMean

Public Member Functions

- double[ ] getNewCenterAndSigmas (LatLonGrid error)

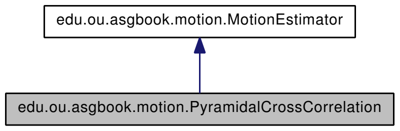
6.92.1 Member Function Documentation

6.92.1.1 double [ ] edu.ou.asgbook.rbf.ProjectionPursuit.NextRBF.getNewCenterAndSigmas (LatLonGrid *error*)

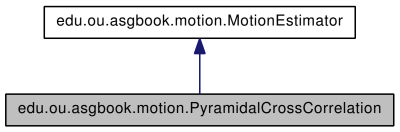
6.93 edu.ou.asgbook.motion.PyramidalCross-Correlation Class Reference

Cross-correlation at muliple resolutions.

Inheritance diagram for edu.ou.asgbook.motion.PyramidalCrossCorrelation:


edu.ou.asgbook.motion.MotionEstimator
edu.ou.asgbook.motion.PyramidalCrossCorrelation

Collaboration diagram for edu.ou.asgbook.motion.PyramidalCrossCorrelation:


edu.ou.asgbook.motion.MotionEstimator
edu.ou.asgbook.motion.PyramidalCrossCorrelation

Public Member Functions

- PyramidalCrossCorrelation (int maxmotion_x, int maxmotion_y)

  Pass in the maximum movement in the two directions.

- Override Pair< LatLonGrid, LatLonGrid > compute (LatLonGrid data0, LatLonGrid data1, File outdir)

  returns motion in the two directions.

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.93.1 Detailed Description

Cross-correlation at muliple resolutions.

Author:

v.lakshmanan

6.93.2 Constructor & Destructor Documentation

6.93.2.1 edu.ou.asgbook.motion.PyramidalCrossCorrelation.PyramidalCrossCorrelation (int *maxmotion_x*, int *maxmotion_y*)

Pass in the maximum movement in the two directions.

```
                                                                              {
        MAX_U = maxmotion_x;
        MAX_V = maxmotion_y;
    }
```

6.93.3 Member Function Documentation

6.93.3.1 Override Pair<LatLonGrid, LatLonGrid> edu.ou.asgbook.motion.PyramidalCrossCorrelation.compute (LatLonGrid *data0*, LatLonGrid *data1*, File *outdir*)

returns motion in the two directions.

The first one is north to south and the second one is east to west. The data is aligned to second time frame. The output dir is used for intermediate products and may be null.

Implements edu.ou.asgbook.motion.MotionEstimator.

```

        int numres = (int) Math.round( Math.log( Math.max(MAX_U,MAX_V))/Math.log(2) );
        System.out.println("Will do computation at " + numres + " resolutions");

        // add missing data to the borders to get them to be divisible by factor
        int factor = (int) ( Math.pow(2, numres) + 0.01 );
        data0 = pad( data0, nextMultiple(data0.getNumLat(),factor), nextMultiple(data0.getNu
        data1 = pad( data1, nextMultiple(data0.getNumLat(),factor), nextMultiple(data0.getNu

        LatLonGrid aligned_data0 = data0;
        LatLonGrid u = null;
        LatLonGrid v = null;
        for (int res = numres; res >= 0; --res){
            // shift data0 using u,v
            if (u != null){
                aligned_data0 = align(data0, u, v);
                if (outdir != null){
                    try {
                        KmlWriter.write(aligned_data0, outdir, "pxaligned_" + res, PngWriter
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                }
            }

            // downsample grids to this resolution
            int smsize = (int) ( Math.pow(2, res) + 0.01 ); // at res=4, this is 2^4 or 16
            LatLonGrid grid0 = decreaseSize(aligned_data0, smsize);
            LatLonGrid grid1 = decreaseSize(data1, smsize);
            if (outdir != null){
                try {
                    KmlWriter.write(grid0, outdir, "pxdata0_" + res, PngWriter.createCoolToWarmColormap
                    KmlWriter.write(grid1, outdir, "pxdata1_" + res, PngWriter.createCoolToWarmColormap
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }

            // find u,v at this resolution
            int est_size = numres-res; // 0 at coarsest resolution, +1 with each resolution
            int search_radius = 1;
            CrossCorrelation xcorr = new CrossCorrelation(est_size, est_size, search_radius, search_rac
            Pair<LatLonGrid,LatLonGrid> motion = xcorr.compute(grid0, grid1, outdir);
            LatLonGrid thisu = increaseSize(motion.first, smsize);
            LatLonGrid thisv = increaseSize(motion.second, smsize);
            thisu = multiply(thisu, smsize); // movement of 1 pixel at res=4 is equal to movement of 16
            thisv = multiply(thisv, smsize);
            // update the total u,v
            u = (u != null)? LatLonGrid.add(thisu, u) : thisu;
            v = (v != null)? LatLonGrid.add(thisv, v) : thisv;
            if (outdir != null){
                try {
                    KmlWriter.write(u, outdir, "pxu_" + res, PngWriter.createCoolToWarmColormap());
                    KmlWriter.write(u, outdir, "pxv_" + res, PngWriter.createCoolToWarmColormap());
                } catch (Exception e) {
                    e.printStackTrace();
                }
             }
        }
        return new Pair<LatLonGrid,LatLonGrid>(u,v);
    }
```

6.93.3.2 static void edu.ou.asgbook.motion.PyramidalCrossCorrelation.main (String[ ] *args*) throws Exception [static]

```
                                                                   {
        // create output directory
        File out = OutputDirectory.getDefault("pyramidxcorr");

        // read
        File f = new File("data/seviri");
        Pair<LatLonGrid,Date>[] grids = SeviriInfraredTemperature.readAll(f);

        // do alg
        PyramidalCrossCorrelation alg = new PyramidalCrossCorrelation(20,20);
        Pair<LatLonGrid,LatLonGrid> motion = alg.compute(grids[0].first, grids[1].first, out);

        // write
        KmlWriter.write(motion.first, out, "pxfinal_u", PngWriter.createCoolToWarmColormap());
        KmlWriter.write(motion.second, out, "pxfinal_v", PngWriter.createCoolToWarmColormap());
    }
```

6.94 edu.ou.asgbook.imgstat.Quantizer Class Reference

Develops a quantization scheme using histogram equalization.

Public Member Functions

- Quantizer (Histogram hist, int K)

  Pass in a high-resolution histogram i.e.

- int getBinNumber (int val)
- int getCenterValue (int bin_no)
- LatLonGrid band (LatLonGrid data)

  replaces each pixel by the center of its bin

- Override String toString ()

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.94.1 Detailed Description

Develops a quantization scheme using histogram equalization.

Author:

valliappa.lakshmanan

6.94.2 Constructor & Destructor Documentation

6.94.2.1 edu.ou.asgbook.imgstat.Quantizer.Quantizer (Histogram *hist*, int *K*)

Pass in a high-resolution histogram i.e.

with incr=1, for example

Parameters:

hist

K number of levels

```
                                        {
        this.min = hist.getMin();
        int incr = hist.getIncr();
        int[] freq = hist.getHist();
        int N = 0; // number of samples
        for (int i=0; i < freq.length; ++i){
            N += freq[i];
        }
        double N_per_level = N/(double)K;

        // populate
        upperBound = new int[K];
        int level_no=0;
        int at_this_level = 0;
        for (int bin_no=0; bin_no < freq.length; ++bin_no){
            if (at_this_level < N_per_level){
                at_this_level += freq[bin_no]; // on to next
            } else {
                upperBound[level_no] = min + (bin_no * incr);
                // next level
                ++level_no;
                at_this_level = freq[bin_no];
            }
        }
        for (; level_no < K; ++level_no){
            upperBound[level_no] = min + freq.length * incr;
        }
        System.out.println(this);
    }
```

6.94.3 Member Function Documentation

6.94.3.1 LatLonGrid edu.ou.asgbook.imgstat.Quantizer.band (LatLonGrid *data*)

replaces each pixel by the center of its bin

```
                                        {
        LatLonGrid result = LatLonGrid.copyOf(data);
        int nrows = result.getNumLat();
        int ncols = result.getNumLon();
        for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
            int bin_no = getBinNumber(data.getValue(i,j));
            int cval = (bin_no < 0)? data.getMissing() : getCenterValue(bin_no);
            result.setValue(i,j, cval);
        }
        return result;
    }
```

6.94.3.2 int edu.ou.asgbook.imgstat.Quantizer.getBinNumber (int *val*)

```
                                 {
```

```
        for (int i=0; i < upperBound.length; ++i){
            if (val < upperBound[i]){
                return i;
            }
        }
        return -1;
    }
```

6.94.3.3 int edu.ou.asgbook.imgstat.Quantizer.getCenterValue (int *bin_no*)

```
                                    {
        int lb = (bin_no > 0)? upperBound[bin_no-1] : this.min;
        int ub = upperBound[bin_no];
        return (ub+lb)/2;
    }
```

6.94.3.4 static void edu.ou.asgbook.imgstat.Quantizer.main (String[ ] *args*) throws Exception [static]

```
                                                            {
        // create output directory
        File outdir = OutputDirectory.getDefault("quantizer");

        // read input
        LatLonGrid conus = SurfaceAlbedo.read(SurfaceAlbedo.CONUS, 100);

        // find histogram in three different ways
        Histogram full = HistogramBinSelection.createHighestResolution(conus);
        for (int k=4; k < 32; k *= 2){ // 4, 8, 16
            Quantizer quant = new Quantizer(full, k);
            LatLonGrid banded = quant.band(conus);
            KmlWriter.write(banded, outdir, "quant_" + k, PngWriter.createCoolToWarmColorma

            int incr = (int) Math.round( full.getIncr() * full.getHist().length / (double)
            Histogram eq = new Histogram(full.getMin(),incr, k);
            banded = HistogramBinSelection.band(conus, eq);
            KmlWriter.write(banded, outdir, "hist_" + k, PngWriter.createCoolToWarmColormap
        }
    }
```

6.94.3.5 Override String edu.ou.asgbook.imgstat.Quantizer.toString ()

```
                            {
        StringBuilder sb = new StringBuilder();
        sb.append("Quantizer levels: ");
        for (int i=0; i < upperBound.length; ++i){
            sb.append(upperBound[i]);
            sb.append(" ");
        }
        return sb.toString();
    }
```

6.95 edu.ou.asgbook.filters.QuickSelect Class Reference

From Numerical Recipes, a fast way to find the kth smallest item in a list Useful to implement rank filters.

Static Public Member Functions

- static int kth_element (int[ ] arr, int k)
- static int kth_element (int[ ] arr, int n, int k)

  Finds the kth smallest item in the list.

6.95.1 Detailed Description

From Numerical Recipes, a fast way to find the kth smallest item in a list Useful to implement rank filters.

Author:

v.lakshmanan

6.95.2 Member Function Documentation

6.95.2.1 static int edu.ou.asgbook.filters.QuickSelect.kth_element (int[ ] *arr*, int *n*, int *k*) `[static]`

Finds the kth smallest item in the list.

Parameters:

arr list

n number of elements in list, in case the last elements of list are unfilled

k finds kth smallest

Returns:

```
                                                                    {
        if (k > n){
            throw new IllegalArgumentException("k should be less than n!");
        }
        int i, ir, j, low, mid;
        int a;

        low = 0;
        ir = n - 1;
        for (;;) {
            if (ir <= low + 1) {
                if (ir == low + 1 && arr[ir] < arr[low]) {
                    SWAP(arr, low, ir);
                }
                return arr[k];
            } else {
                mid = (low + ir) >> 1;
                SWAP(arr, mid, low + 1);
                if (arr[low] > arr[ir]) {
                    SWAP(arr, low, ir);
                }
                if (arr[low + 1] > arr[ir]) {
                    SWAP(arr, low + 1, ir);
                }
                if (arr[low] > arr[low + 1]) {
                    SWAP(arr, low, low + 1);
                }
                i = low + 1;
                j = ir;
                a = arr[low + 1];
                for (;;) {
                    do
                        i++;
                    while (arr[i] < a);
                    do
                        j--;
                    while (arr[j] > a);
                    if (j < i)
                        break;
                    SWAP(arr, i, j);
                }
                arr[low + 1] = arr[j];
                arr[j] = a;
                if (j >= k)
                    ir = j - 1;
                if (j <= k)
                    low = i;
            }
        }
    }
```

6.95.2.2 static int edu.ou.asgbook.filters.QuickSelect.kth_element (int[ ] *arr*, int *k*) [static]

```
                                                    {
        return kth_element(arr, arr.length, k);
    }
```

6.96 edu.ou.asgbook.rbf.RadialBasisFunction Class Reference

Finds best fit of a spatial grid to a sum of Gaussians when the centers and sigmas of the Gaussians are known.

Static Public Member Functions

- static double[ ] fit (LatLonGrid data, Pixel[ ] center, double[ ] sigmax, double[ ] sigmay)

  Provide the center locations and this function will fill in the optimal amplitude.

- static void main (String[ ] args)

6.96.1 Detailed Description

Finds best fit of a spatial grid to a sum of Gaussians when the centers and sigmas of the Gaussians are known.

Author:

v.lakshmanan

6.96.2 Member Function Documentation

6.96.2.1 static double [ ] edu.ou.asgbook.rbf.RadialBasisFunction.fit (LatLonGrid *data*, Pixel[ ] *center*, double[ ] *sigmax*, double[ ] *sigmay*) [static]

Provide the center locations and this function will fill in the optimal amplitude.

```

        // inv( transpose(H) * H) * transpose(H) * data
        int p = data.getNumLat() * data.getNumLon();
        int m = center.length;
        Matrix H = new Matrix(p, m);
        Matrix ycap = new Matrix(p, 1);
        for (int i=0; i < p; ++i){
            int x = i / data.getNumLon();
            int y = i % data.getNumLon();
            for (int j=0; j < m; ++j){
                double xdist = x - center[j].getX();
                double ydist = y - center[j].getY();
                double xnorm = (xdist*xdist) / (sigmax[j] * sigmax[j]);
                double ynorm = (ydist*ydist) / (sigmay[j] * sigmay[j]);
                double wt = Math.exp(-(xnorm + ynorm));
                H.set(i,j, wt);
            }
            ycap.set(i, 0, data.getValue(x, y));
        }
        // H.print(H.getColumnDimension(), H.getRowDimension());

        Matrix HT = H.transpose();
        Matrix HTH = HT.times(H);
        Matrix HTHinv = HTH.inverse();
        Matrix HTHinvHT = HTHinv.times(HT);

        return HTHinvHT.times(ycap).transpose().getArray()[0];
    }
```

6.96.2.2 static void edu.ou.asgbook.rbf.RadialBasisFunction.main (String[ ] *args*) [static]

```
                                          {
        int nrows = 100;
        int ncols = 100;
        Pixel[] centers = new Pixel[]{ new Pixel(nrows/4,ncols/3,20), new Pixel(nrows/3,ncols/2,10) };
        double[] sigmax = new double[] { nrows/12, ncols/8 };
        double[] sigmay = new double[] { nrows/8, ncols/12 };
        LatLonGrid m = DataSimulator.simulateData(centers, sigmax, sigmay, nrows, ncols);

        System.out.println("Created data of size " + m.getNumLat() + "x" + m.getNumLon());
        double[] weights = fit( m, centers, sigmax, sigmay );
        for (int i=0; i < weights.length; ++i){
            System.out.println("Actual: " + centers[i].getValue() + " RBF: " + +weights[i]);
        }
    }
```

6.97 edu.ou.asgbook.segmentation.RegionGrowing Class Reference

Common object-identification utility.

Static Public Member Functions

- static void growRegion (int x, int y, LatLonGrid data, int thresh, LatLonGrid label, int currLabel)

6.97.1 Detailed Description

Common object-identification utility.

Author:

v.lakshmanan

6.97.2 Member Function Documentation

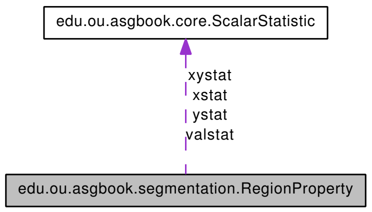
6.97.2.1 static void edu.ou.asgbook.segmentation.RegionGrowing.growRegion (int *x*, int *y*, LatLonGrid *data*, int *thresh*, LatLonGrid *label*, int *currLabel*) [static]

```

        final int junk = 0; // data value not needed for region growing
        final int UNSET = 0;
        List<Pixel> stack = new ArrayList<Pixel>();
        stack.add(new Pixel(x,y,junk));
        while (stack.size() > 0){
            Pixel p = stack.remove(stack.size()-1);
            label.setValue(p.getX(), p.getY(), currLabel);
            for (int i=p.getX()-1; i <= p.getX()+1; ++i){
                for (int j=p.getY()-1; j <= p.getY()+1; ++j){
                    if (data.isValid(i, j) && data.getValue(i,j) > thresh && label.getValue
                        stack.add(new Pixel(i,j,junk));
                    }
                }
            }
        }
    }
```

6.98 edu.ou.asgbook.segmentation.RegionProperty Class Reference

Properties of a region such as geometric (centroid, area, etc) and physical (based on other grid values).

Collaboration diagram for edu.ou.asgbook.segmentation.RegionProperty:


edu.ou.asgbook.core.ScalarStatistic
xystat
xstat
ystat
valstat
edu.ou.asgbook.segmentation.RegionProperty

Public Member Functions

- double getCx ()
- double getCy ()
- double getCval ()
- int getSize ()
- Ellipse getEllipseFit ()

Static Public Member Functions

- static RegionProperty[ ] compute (LabelResult label, LatLonGrid data)
- static Pixel[ ][ ] getPixelsInRegions (LatLonGrid data1, LabelResult objects1)

  All the pixels for each region.

- static LabelResult prune (LabelResult input, boolean[ ] keep)

  Regions for which keep=false are removed.

- static LabelResult pruneBySize (LabelResult input, LatLonGrid grid, int sizethresh)

  Regions smaller than sizethresh are removed.

- static void main (String[ ] args) throws Exception

Classes

- class **Ellipse**

6.98.1 Detailed Description

Properties of a region such as geometric (centroid, area, etc) and physical (based on other grid values).

Author:

v.lakshmanan

6.98.2 Member Function Documentation

6.98.2.1 static RegionProperty [ ] edu.ou.asgbook.segmentation.RegionProperty.compute (LabelResult *label*, LatLonGrid *data*) [static]

```
                                                                                    {
        RegionProperty[] props = new RegionProperty[label.maxlabel+1];
        for (int i=1; i < props.length; ++i){
            props[i] = new RegionProperty();
        }
        int nrows = label.label.getNumLat();
        int ncols = label.label.getNumLon();
        for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
            if (label.label.getValue(i,j) > 0){
                props[ label.label.getValue(i,j) ].update(i, j, data.getValue(i,j));
            }
        }
        return props;
    }
```

6.98.2.2 double edu.ou.asgbook.segmentation.RegionProperty.getCval ()

```
                          {
        return valstat.getMean();
    }
```

6.98.2.3 double edu.ou.asgbook.segmentation.RegionProperty.getCx ()

```
                        {
        return xstat.getMean();
    }
```

6.98.2.4 double edu.ou.asgbook.segmentation.RegionProperty.getCy ()

```
                        {
        return ystat.getMean();
    }
```

6.98.2.5 Ellipse edu.ou.asgbook.segmentation.RegionProperty.getEllipseFit ()

```
                                    {
        final double cx = xstat.getMean();
        final double cy = ystat.getMean();
        final double s11 = xstat.getVariance();
        final double s22 = ystat.getVariance();
        final double s12 = xystat.getMean() - cx*cy;
        double tmp = (s11 - s22) * (s11 - s22) + 4 * s12 * s12;
        if (tmp >= 0.00001) {
            tmp = Math.sqrt(tmp);
        } else {
            tmp = 0;
        }
        double eigen1 = (s11 + s22 + tmp) / 2;
        double eigen2 = (s11 + s22 - tmp) / 2;

         double v1 = s12
                 / Math.sqrt((eigen1 - s11) * (eigen1 - s11) + s12 * s12);
         double v2 = (eigen1 - s11)
                 / Math.sqrt((eigen1 - s11) * (eigen1 - s11) + s12 * s12);

         double a = 2 * Math.sqrt(eigen1);
         double b = 2 * Math.sqrt(eigen2);
         double phi = Math.toDegrees(Math.atan2(v2, v1));
         return new Ellipse(cx, cy, a, b, phi);

    }
```

6.98.2.6 static Pixel [ ][ ] edu.ou.asgbook.segmentation.RegionProperty.getPixelsInRegions (LatLonGrid *data1*, LabelResult *objects1*) [static]

All the pixels for each region.

The array is organized as pixels[regno][pixelno]

Parameters:

data1

objects1

Returns:

```
                                                                                                          {
        @SuppressWarnings("unchecked")
        List<Pixel>[] regions = new List[objects1.maxlabel+1];
        for (int reg=1; reg < regions.length; ++reg){
            regions[reg] = new ArrayList<Pixel>();
        }
        for (int i=0; i < data1.getNumLat(); ++i) for (int j=0; j < data1.getNumLon(); ++j
            int reg = objects1.label.getValue(i,j);
            if (reg > 0){
                regions[reg].add(new Pixel(i,j,data1.getValue(i,j)));
            }
        }
        Pixel[][] result = new Pixel[regions.length][];
        for (int i=1; i < result.length; ++i){
            result[i] = regions[i].toArray(new Pixel[0]);
        }
        return result;
    }
```

6.98.2.7 int edu.ou.asgbook.segmentation.RegionProperty.getSize ()

```
                        {
        return xstat.getNumSamples();
    }
```

6.98.2.8 static void edu.ou.asgbook.segmentation.RegionProperty.main (String[ ] *args*) throws Exception [static]

```
                                                                {
        File out = OutputDirectory.getDefault("regionproperty");

        // data
        LatLonGrid grid = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new GlobalPo
        KmlWriter.write(grid, out, "orig", PngWriter.createCoolToWarmColormap());

        // global thresh
        int thresh = 20;
        ThresholdSegmenter seg = new ThresholdSegmenter(thresh);
        LabelResult labelResult = seg.label(grid);
        labelResult.label.setMissing(-1); // to get background color
        KmlWriter.write(labelResult.label, out, "allcities_"+thresh, PngWriter.createRandom

        // prune cities less than 5 pixels in size
        for (int sizethresh = 2; sizethresh <= 5; ++sizethresh){
            LabelResult pruned = pruneBySize(labelResult, grid, sizethresh);
            pruned.label.setMissing(-1); // to get background color
            KmlWriter.write(pruned.label, out, "sizepruned_"+sizethresh, PngWriter.createRa
            // get geocode
            RegionProperty[] prop = RegionProperty.compute(pruned, grid);
            for (int i=1; i < prop.length; ++i){
                LatLon loc = grid.getLocation(prop[i].getCx(), prop[i].getCy());
                UsaZipcode.Entry entry = UsaZipcode.getInstance().getEntryClosestTo(loc);
                System.out.println(entry + " " + prop[i].getEllipseFit());
            }

            // there are more efficient ways to paint an ellipse, but this will do
            Ellipse[] ellipses = new Ellipse[prop.length];
            for (int i=1; i < prop.length; ++i){
                ellipses[i] = prop[i].getEllipseFit();
            }
            LatLonGrid ellipse = LatLonGrid.copyOf(pruned.label);
            ellipse.fill(0);
            int nrows = ellipse.getNumLat();
            int ncols = ellipse.getNumLon();
            for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
                for (int k=1; k < prop.length; ++k){
                    if (ellipses[k].contains(i, j)){
                        ellipse.setValue(i, j, k); // paint pixel
                    }
                }
            }
            KmlWriter.write(ellipse, out, "ellipse_"+sizethresh, PngWriter.createRandomColormap());
        }
    }
```

6.98.2.9 static LabelResult edu.ou.asgbook.segmentation.RegionProperty.prune (LabelResult *input*, boolean[ ] *keep*) [static]

Regions for which keep=false are removed.

```
                                                                           {
        // find mapping
        int[] newRegionNo = new int[keep.length]; // init to zero
        int numRegions = 0;
        for (int i=1; i < keep.length; ++i){
            if ( keep[i] ){
                ++numRegions;
                newRegionNo[i] = numRegions;
            }
        }

        // replace old label by new label
        LatLonGrid newLabel = LatLonGrid.copyOf(input.label);
        int nrows = newLabel.getNumLat();
        int ncols = newLabel.getNumLon();
        for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
            int oldno = input.label.getValue(i,j);
            newLabel.setValue(i,j, newRegionNo[oldno]);
        }
        return new LabelResult(newLabel, numRegions);
    }
```

6.98.2.10 static LabelResult edu.ou.asgbook.segmentation.RegionProperty.pruneBySize (LabelResult *input*, LatLonGrid *grid*, int *sizethresh*) [static]

Regions smaller than sizethresh are removed.

```

        RegionProperty[] prop = RegionProperty.compute(input, grid);

        // find mapping
        int[] newRegionNo = new int[prop.length]; // init to zero
        int numRegions = 0;
        for (int i=1; i < prop.length; ++i){
            if ( prop[i].getSize() >= sizethresh ){
                ++numRegions;
                newRegionNo[i] = numRegions;
            }
        }

        // replace old label by new label
        LatLonGrid newLabel = LatLonGrid.copyOf(input.label);
        int nrows = newLabel.getNumLat();
        int ncols = newLabel.getNumLon();
        for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
            int oldno = input.label.getValue(i,j);
            newLabel.setValue(i,j, newRegionNo[oldno]);
        }
        return new LabelResult(newLabel, numRegions);
    }
```

6.99 edu.ou.asgbook.projections.Remapper Class Reference

Utilities to remap one map projection to another.

Static Public Member Functions

- static int nearestNeighbor (double rowno, double colno, int[ ][ ] input, int missing)
- static int bilinearInterpolation (double rowno, double colno, int[ ][ ] input, int missing)

6.99.1 Detailed Description

Utilities to remap one map projection to another.

Author:

valliappa.lakshmanan

6.99.2 Member Function Documentation

6.99.2.1 static int edu.ou.asgbook.projections.Remapper.bilinearInterpolation (double *rowno*, double *colno*, int *input*[ ][ ], int *missing*) [static]

```
                                                                                                   {
        final int row0 = (int) Math.floor( rowno );
        final int col0 = (int) Math.floor( colno );
        final int row1 = (int) Math.ceil( rowno );
        final int col1 = (int) Math.ceil( colno );
        final int nrows = input.length;
        final int ncols = (nrows > 0)? input[0].length : 0;

        int npts = 0;
        double totwt = 0;
        double totval = 0;
        for (int row = row0; row <= row1; ++row){
            for (int col = col0; col <= col1; ++col){
                if ( row >= 0 && col >= 0 && row < nrows && col < ncols && input[row][col] != missing )
                    double rowwt = 1 - Math.abs(rowno-row);
                    double colwt = 1 - Math.abs(colno-col);
                    double wt = rowwt * colwt;
                    npts++;
                    totwt += wt;
                    totval += wt * input[row][col];
                }
            }
        }

        // weighted average
        if (npts == 0){
            return missing;
        } else {
            return (int) Math.round(totval / totwt);
        }
    }
```

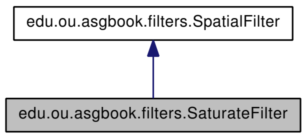
6.99.2.2 static int edu.ou.asgbook.projections.Remapper.nearestNeighbor (double *rowno*, double *colno*, int *input*[ ][ ], int *missing*) [static]

```

        final int row = (int) Math.round( rowno );
        final int col = (int) Math.round( colno );
        final int nrows = input.length;
        final int ncols = (nrows > 0)? input[0].length : 0;
        if ( row >= 0 && col >= 0 && row < nrows && col < ncols ){
            return input[row][col];
        } else {
            return missing;
        }
    }
```

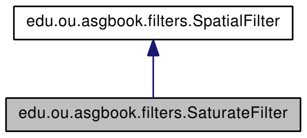
6.100 edu.ou.asgbook.filters.SaturateFilter Class Reference

Sets all values < MIN to MIN and all values > MAX to MAX.

Inheritance diagram for edu.ou.asgbook.filters.SaturateFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.SaturateFilter

Collaboration diagram for edu.ou.asgbook.filters.SaturateFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.SaturateFilter

Public Member Functions

- SaturateFilter (int min, int max)
- Override LatLonGrid filter (LatLonGrid input)
- LatLonGrid saturate (final LatLonGrid input)

6.100.1 Detailed Description

Sets all values < MIN to MIN and all values > MAX to MAX.

Author:

Valliappa.Lakshmanan

6.100.2 Constructor & Destructor Documentation

6.100.2.1 edu.ou.asgbook.filters.SaturateFilter.SaturateFilter (int *min*, int *max*)

```
                                         {
        this.min = min;
        this.max = max;
    }
```

6.100.3 Member Function Documentation

6.100.3.1 Override LatLonGrid edu.ou.asgbook.filters.SaturateFilter.filter (LatLonGrid *input*)

```
                                                    {
        return saturate(input);
    }
```

6.100.3.2 LatLonGrid edu.ou.asgbook.filters.SaturateFilter.saturate (final LatLonGrid *input*)

```
                                                           {
        LatLonGrid output = LatLonGrid.copyOf(input);
        int[][] outData = output.getData();
        int[][] inData = input.getData();
        for (int i=0; i < output.getNumLat(); ++i){
            for (int j=0; j < output.getNumLon(); ++j){
                int inval = inData[i][j];
                if ( inval < min || inval == input.getMissing() ){
                    outData[i][j] = min;
                } else if ( inval > max ){
                    outData[i][j] = max;
                }
            }
        }
        return output;
    }
```

6.101 edu.ou.asgbook.core.ScalarStatistic Class Reference

A utility class to compute mean, variance of a streaming set of inputs.

Public Member Functions

- void update (double x)
- void update (double x, int relwt)
- void update (ScalarStatistic other)
- double getMean ()
- double getMin ()
- double getMax ()
- double getVariance ()
- double getStdDeviation ()
- Override String toString ()
- int getNumSamples ()

6.101.1 Detailed Description

A utility class to compute mean, variance of a streaming set of inputs.

Author:

v.lakshmanan

6.101.2 Member Function Documentation

6.101.2.1 double edu.ou.asgbook.core.ScalarStatistic.getMax ()

```
                             {
        return max;
    }
```

6.101.2.2 double edu.ou.asgbook.core.ScalarStatistic.getMean ()

```
                              {
        compute();
        return mean;
    }
```

6.101.2.3 double edu.ou.asgbook.core.ScalarStatistic.getMin ()

```
                           {
        return min;
    }
```

6.101.2.4 int edu.ou.asgbook.core.ScalarStatistic.getNumSamples ()

```
                                {
        return N;
    }
```

6.101.2.5 double edu.ou.asgbook.core.ScalarStatistic.getStdDeviation ()

```
                                    {
        compute();
        return stddev;
    }
```

6.101.2.6 double edu.ou.asgbook.core.ScalarStatistic.getVariance ()

```
                                {
        compute();
        return var;
    }
```

6.101.2.7 Override String edu.ou.asgbook.core.ScalarStatistic.toString ()

```
                            {
        return "value = " + getMean() + "+/-" + getStdDeviation() + " based on " + N + " sa
    }
```

6.101.2.8 void edu.ou.asgbook.core.ScalarStatistic.update (ScalarStatistic *other*)

```
                                              {
        sumx += other.sumx;
        sumx2 += other.sumx2;
        if ( N == 0 ){
            min = other.min;
            max = other.max;
        } else if (other.N != 0){
            min = Math.min(min, other.min);
            max = Math.max(max, other.max);
        }
        N += other.N;
    }
```

6.101.2.9 void edu.ou.asgbook.core.ScalarStatistic.update (double *x*, int *relwt*)

```
                                        {
        sumx += (x*relwt);
        sumx2 += (x*x*relwt);
        if ( N == 0 ){
            min = max = x;
        } else {
            min = Math.min(min, x);
            max = Math.max(max, x);
        }
        N += relwt;
    }
```

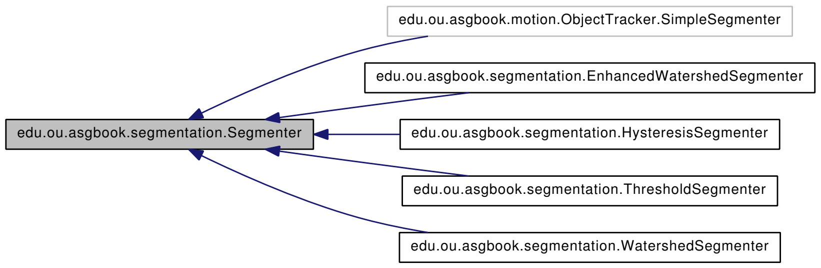
6.101.2.10 void edu.ou.asgbook.core.ScalarStatistic.update (double *x*)

```
                               {
        sumx += x;
        sumx2 += x*x;
        if ( N == 0 ){
            min = max = x;
        } else {
            min = Math.min(min, x);
            max = Math.max(max, x);
        }
        ++N;
    }
```

6.102 edu.ou.asgbook.segmentation.Segmenter Interface Reference

Object identification technique.

Inheritance diagram for edu.ou.asgbook.segmentation.Segmenter:


edu.ou.asgbook.motion.ObjectTracker.SimpleSegmenter
edu.ou.asgbook.segmentation.EnhancedWatershedSegmenter
edu.ou.asgbook.segmentation.Segmenter
edu.ou.asgbook.segmentation.HysteresisSegmenter
edu.ou.asgbook.segmentation.ThresholdSegmenter
edu.ou.asgbook.segmentation.WatershedSegmenter

Public Member Functions

- abstract LabelResult label (LatLonGrid data)

  Creates a labeled grid where background pixels are set to 0 and labels for objects go 1,2,3.

6.102.1 Detailed Description

Object identification technique.

Author:

valliappa.lakshmanan

6.102.2 Member Function Documentation

6.102.2.1 abstract LabelResult edu.ou.asgbook.segmentation.Segmenter.label (LatLonGrid *data*) `[pure virtual]`

Creates a labeled grid where background pixels are set to 0 and labels for objects go 1,2,3.

.. All pixels > thresh are part of an object.

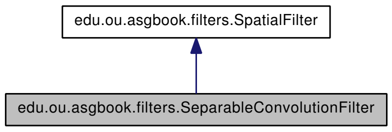
Implemented in edu.ou.asgbook.segmentation.Enhanced-WatershedSegmenter, edu.ou.asgbook.segmentation.Hysteresis-

Segmenter, edu.ou.asgbook.segmentation.ThresholdSegmenter, and edu.ou.asgbook.segmentation.WatershedSegmenter.

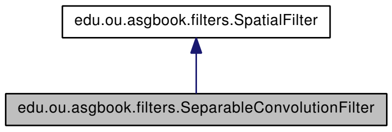
6.103 edu.ou.asgbook.filters.SeparableConvolution-Filter Class Reference

An optimized convolution filter.

Inheritance diagram for edu.ou.asgbook.filters.SeparableConvolutionFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.SeparableConvolutionFilter

Collaboration diagram for edu.ou.asgbook.filters.SeparableConvolutionFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.SeparableConvolutionFilter

Public Member Functions

- SeparableConvolutionFilter (double[ ] coeffs_x, double[ ] coeffs_y)
- SeparableConvolutionFilter (double[ ] coeffs)
- LatLonGrid smooth (final LatLonGrid input)
- Override LatLonGrid filter (LatLonGrid input)

Static Public Member Functions

- static SeparableConvolutionFilter boxcar (int numx, int numy)
- static SeparableConvolutionFilter gauss (int numx, int numy)
- static double[ ] gauss (int numx)
- static void main (String[ ] args) throws Exception

6.103.1 Detailed Description

An optimized convolution filter.

Author:

Valliappa.Lakshmanan

6.103.2 Constructor & Destructor Documentation

6.103.2.1 edu.ou.asgbook.filters.SeparableConvolutionFilter.SeparableConvolutionFilter (double[ ] *coeffs_x*, double[ ] *coeffs_y*)

```
                                                                             {
         this.coeffs_x = coeffs_x;
         if ( coeffs_x.length % 2 == 0 ){
             throw new IllegalArgumentException("Dimensions of coefficients array needs to be odd");
         }
         this.coeffs_y = coeffs_y;
         if ( coeffs_y.length % 2 == 0 ){
             throw new IllegalArgumentException("Dimensions of coefficients array needs to be odd");
         }
     }
```

6.103.2.2 edu.ou.asgbook.filters.SeparableConvolutionFilter.SeparableConvolutionFilter (double[ ] *coeffs*)

```
                                                      {
         this(coeffs,coeffs);
     }
```

6.103.3 Member Function Documentation

6.103.3.1 static SeparableConvolutionFilter edu.ou.asgbook.filters.SeparableConvolutionFilter.boxcar (int *numx*, int *numy*) [static]

```
                                                                        {
         double[] coeffs_x = new double[numx];
         double[] coeffs_y = new double[numy];
         for (int i=0; i < numx; ++i){
             coeffs_x[i] = 1.0/numx;
         }
         for (int i=0; i < numy; ++i){
             coeffs_y[i] = 1.0/numy;
         }
         return new SeparableConvolutionFilter(coeffs_x, coeffs_y);
     }
```

6.103.3.2 Override LatLonGrid edu.ou.asgbook.filters.SeparableConvolutionFilter.filter (LatLonGrid *input*)

```
                                                 {
         return smooth(input);
     }
```

6.103.3.3 static double [ ] edu.ou.asgbook.filters.SeparableConvolutionFilter.gauss (int *numx*) [static]

```
                                         {
        double[] coeffs = new double[numx];
        double sigmax = numx / 6.0; // 3-sigma on either side
        for (int i=0; i < coeffs.length; ++i){
            double x = (i - coeffs.length/2.0)/sigmax;
            coeffs[i] = Math.exp(-(x*x));
        }
        return coeffs;
    }
```

6.103.3.4 static SeparableConvolutionFilter edu.ou.asgbook.filters.SeparableConvolutionFilter.gauss (int *numx*, int *numy*) [static]

```
                                                                       {
        return new SeparableConvolutionFilter(gauss(numx), gauss(numy));
    }
```

6.103.3.5 static void edu.ou.asgbook.filters.SeparableConvolutionFilter.main (String[ ] *args*) throws Exception [static]

```
                                                             {
        // create output directory
        File out = OutputDirectory.getDefault("separable");

        // read input
        LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new G
        KmlWriter.write(popdensity, out, "orig", PngWriter.createCoolToWarmColormap());

        // boxcar
        {
            SeparableConvolutionFilter filter = SeparableConvolutionFilter.boxcar(5, 5);
            LatLonGrid sm = filter.smooth(popdensity);
            KmlWriter.write(sm, out, "boxcar", PngWriter.createCoolToWarmColormap());
        }

        // gauss
        {
            SeparableConvolutionFilter filter = SeparableConvolutionFilter.gauss(11, 11);
            LatLonGrid sm = filter.smooth(popdensity);
            KmlWriter.write(sm, out, "gauss", PngWriter.createCoolToWarmColormap());
        }
    }
```

6.103.3.6 LatLonGrid edu.ou.asgbook.filters.SeparableConvolutionFilter.smooth (final LatLonGrid *input*)

```
                                              {
        int[][] inData = input.getData();
        final int nx = input.getNumLat();
        final int ny = input.getNumLon();

        // filter the rows
        final int hx = coeffs_x.length / 2;
        int[][] rowResult = new int[nx][ny];
        for (int j=0; j < ny; ++j){
            for (int i=hx; i < (nx-hx); ++i){
                double tot = 0;
                double wt = 0;
                for (int m=-hx; m <= hx; ++m){
                    double coeff = coeffs_x[m+hx];
                    int inval = inData[i+m][j];
                    if (inval != input.getMissing()){
                        tot += inval*coeff;
                        wt += coeff;
                    }
                }
                if ( wt > 0 ){
                    rowResult[i][j] = (int)( Math.round(tot / wt) );
                }
            }
        }

        // now filter the columns of rowResult
        inData = rowResult;
        LatLonGrid output = LatLonGrid.copyOf(input);
        output.fill(output.getMissing());
        final int hy = coeffs_y.length / 2;
        int[][] outData = output.getData();
        for (int i=0; i < nx; ++i){
            for (int j=hy; j < (ny-hy); ++j){
                double tot = 0;
                double wt = 0;
                for (int n=-hy; n <= hy; ++n){
                    double coeff = coeffs_y[n+hy];
                    int inval = inData[i][j+n];
                    if (inval != input.getMissing()){
                        tot += inval*coeff;
                        wt += coeff;
                    }
                }
                if ( wt > 0 ){
                    outData[i][j] = (int)( Math.round(tot / wt) );
                }
            }
        }

        return output;
    }
```

6.104 edu.ou.asgbook.dataset.SeviriInfrared-Temperature Class Reference

To read binary dump output from WDSS-II (http://www.wdssii.org/).

Static Public Member Functions

- static Pair< LatLonGrid, Date > read (File f) throws IOException, Parse-Exception
- static Pair< LatLonGrid, Date >[ ] readAll (File dir) throws IOException, Parse-Exception
- static void main (String[ ] args) throws Exception

Static Package Functions

- [static initializer]

6.104.1 Detailed Description

To read binary dump output from WDSS-II (http://www.wdssii.org/).

Author:

v.lakshmanan

6.104.2 Member Function Documentation

6.104.2.1]

edu.ou.asgbook.dataset.SeviriInfraredTemperature.[static initializer] () `[static, package]`

6.104.2.2 static void edu.ou.asgbook.dataset.SeviriInfraredTemperature.main (String[ ] *args*) throws Exception `[static]`

```
                                                        {
          File f = new File("data/seviri");
          Pair<LatLonGrid,Date>[] grids = readAll(f);

          // create output directory
          File out = OutputDirectory.getDefault("seviri");
          // write out as image, for viewing
          for (int i=0; i < grids.length; ++i){
              KmlWriter.write(grids[i].first, out, "ir_"+i, PngWriter.createCoolToWarmColormap());
          }
      }
```

6.104.2.3 static Pair<LatLonGrid,Date> edu.ou.asgbook.dataset.SeviriInfraredTemperature.read (File *f*) throws IOException, ParseException [static]

```
                                                                                         {
        // parse filename: eg: MSG_20050105-150000_556x1111_60.00_-10.00_Channel_09_0.027_0.027.llg
        String[] pieces = f.getName().replace(".llg", "").split("_");
        int pieceno = 0;
        pieceno++; // typeName ignored
        String date = pieces[pieceno++];
        String[] dimpieces = pieces[pieceno++].split("x");
        int numrows = Integer.parseInt(dimpieces[0]);
        int numcols = Integer.parseInt(dimpieces[1]);
        float nwlat = Float.parseFloat(pieces[pieceno++]);
        float nwlon = Float.parseFloat(pieces[pieceno++]);
        ++pieceno; // subtype ignored
        int numleft = pieces.length - pieceno - 2;
        pieceno += numleft; // subtype has an underscore
        float deltalat = Float.parseFloat(pieces[pieceno++]);
        float deltalon = Float.parseFloat(pieces[pieceno++]);

        // read in LatLonGrid
        FileInputStream fis = new FileInputStream(f);
        byte[] bytes = new byte[numrows*numcols];
        fis.read(bytes);
        LatLonGrid grid = new LatLonGrid(numrows, numcols, 0, new LatLon(nwlat,nwlon), deltalat, deltal
        int index = 0;
        for (int i=0; i < numrows; ++i){
            for (int j=0; j < numcols; ++j){
                int value = 256 + bytes[index++]; // 0 to 255
                int reversed = 256 - value; // assign higher value to colder pixels
                grid.setValue(i,j, reversed);
            }
        }

        // format date
        SimpleDateFormat df = new SimpleDateFormat("yyyyMMdd-HHmmSS");
        Date gridTime = df.parse(date);

        System.out.println("Read grid at " + gridTime);
        return new Pair<LatLonGrid,Date>(grid,gridTime);
    }
```

6.104.2.4 static Pair<LatLonGrid,Date> [ ] edu.ou.asgbook.dataset.SeviriInfraredTemperature.readAll (File *dir*) throws IOException, ParseException [static]

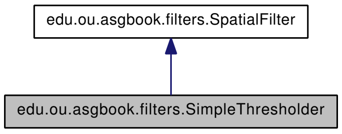
```
                                                                                                {
```

```
        File[] files = dir.listFiles(new FileFilter(){
            @Override
            public boolean accept(File f) {
                return f.getName().endsWith(".llg");
            }
        });
        @SuppressWarnings("unchecked")
        Pair<LatLonGrid,Date>[] result = new Pair[files.length];
        for (int i=0; i < result.length; ++i){
            result[i] = read(files[i]);
        }
        return result;
    }
```

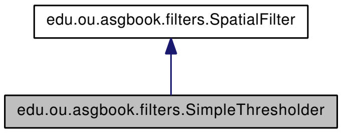
6.105 edu.ou.asgbook.filters.SimpleThresholder Class Reference

Replace pixel values with 1 or 0 depending on whether they are above or below a single threshold.

Inheritance diagram for edu.ou.asgbook.filters.SimpleThresholder:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.SimpleThresholder

Collaboration diagram for edu.ou.asgbook.filters.SimpleThresholder:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.SimpleThresholder

Public Member Functions

- SimpleThresholder (int thresh)
- Override LatLonGrid filter (LatLonGrid input)
- LatLonGrid threshold (final LatLonGrid input)

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.105.1 Detailed Description

Replace pixel values with 1 or 0 depending on whether they are above or below a single threshold.

Author:

Valliappa.Lakshmanan

6.105.2 Constructor & Destructor Documentation

6.105.2.1 edu.ou.asgbook.filters.SimpleThresholder.SimpleThresholder (int *thresh*)

```
                                    {
        this.thresh = thresh;
    }
```

6.105.3 Member Function Documentation

6.105.3.1 Override LatLonGrid edu.ou.asgbook.filters.SimpleThresholder.filter (LatLonGrid *input*)

```
                                          {
        return threshold(input);
    }
```

6.105.3.2 static void edu.ou.asgbook.filters.SimpleThresholder.main (String[ ] *args*) throws Exception [static]

```
                                                          {
        // create output directory
        File out = OutputDirectory.getDefault("threshold");

        // read input
        LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA);
        popdensity.setMissing(0); // transparent
        KmlWriter.write(popdensity, out, "popdensity", PngWriter.createCoolToWarmColormap()

        // threshold
        SimpleThresholder filter = new SimpleThresholder(100);
        LatLonGrid thresh = filter.threshold(popdensity);
        KmlWriter.write(thresh, out, "highdensity", PngWriter.createCoolToWarmColormap());
    }
```

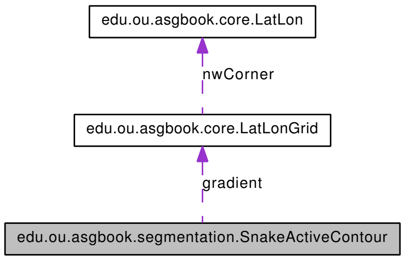
6.105.3.3 LatLonGrid edu.ou.asgbook.filters.SimpleThresholder.threshold (final LatLonGrid *input*)

```
                                                  {
        LatLonGrid output = LatLonGrid.copyOf(input);
        int[][] outData = output.getData();
        int[][] inData = input.getData();
        for (int i=0; i < output.getNumLat(); ++i){
            for (int j=0; j < output.getNumLon(); ++j){
                outData[i][j] = (inData[i][j] >= thresh)? 1 : 0;
            }
        }
        return output;
    }
```

6.106 edu.ou.asgbook.segmentation.SnakeActiveContour Class Reference

Active contour method of identifying objects.

Collaboration diagram for edu.ou.asgbook.segmentation.SnakeActiveContour:


edu.ou.asgbook.core.LatLon
nwCorner
edu.ou.asgbook.core.LatLonGrid
gradient
edu.ou.asgbook.segmentation.SnakeActiveContour

Public Member Functions

- SnakeActiveContour (LatLonGrid gradient)
- Snake pruneAndResample (Snake inputSnake)
- Snake resampleNodes (Snake inputSnake)
- Snake resample (Snake inputSnake)
- SnakeNode[ ] moveSnake (SnakeNode[ ] pixels, int numIter)

    Provide an initial guess of points.

Static Public Member Functions

- static void main (String[ ] args) throws Exception

Classes

- class Snake
- class **SnakeNode**

6.106.1 Detailed Description

Active contour method of identifying objects.

Author:

v.lakshmanan

6.106.2 Constructor & Destructor Documentation

6.106.2.1 edu.ou.asgbook.segmentation.SnakeActiveContour.SnakeActiveContour (LatLonGrid *gradient*)

```
                                                  {
        // Normalize the gradient grid to lie between 0 and 100
        this.gradient = LatLonGrid.copyOf(gradient);
        int maxgradient = 0;
        for (int i=0; i < gradient.getNumLat(); ++i) for (int j=0; j < gradient.getNumLon(); ++j){
            if ( gradient.getValue(i, j) > maxgradient ){
                maxgradient = gradient.getValue(i, j);
            }
        }
        for (int i=0; i < gradient.getNumLat(); ++i) for (int j=0; j < gradient.getNumLon(); ++j){
            if ( gradient.getValue(i, j) != gradient.getMissing() ){
                this.gradient.setValue(i, j, (gradient.getValue(i, j) * GRADIENT_SCALE) / maxgradient);
            } else {
                this.gradient.setValue(i, j, 0);
            }
        }
    }
```

6.106.3 Member Function Documentation

6.106.3.1 static void edu.ou.asgbook.segmentation.SnakeActiveContour.main (String[ ] *args*) throws Exception `[static]`

Parameters:

args

```
                                                            {
        File out = OutputDirectory.getDefault("snake");

        // data
        LatLonGrid grid = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new GlobalPopulation.Li
        KmlWriter.write(grid, out, "orig", PngWriter.createCoolToWarmColormap());

        // find cities > 10 px and more than 20,000 residents/km^2
        int thresh = 20;
        Segmenter seg = new HysteresisSegmenter(thresh, thresh-5);
        LabelResult labelResult = seg.label(grid);
        KmlWriter.write(labelResult.label, out, "allcities_"+thresh, PngWriter.createRandomColormap())
        int sizethresh = 10;
        LabelResult pruned = RegionProperty.pruneBySize(labelResult, grid, sizethresh);
        RegionProperty[] prop = RegionProperty.compute(pruned, grid);
        KmlWriter.write(pruned.label, out, "largecities_"+sizethresh, PngWriter.createRandomColormap(

        // threshold image and compute gradient of thresholded image
        LatLonGrid binaryImage = new SimpleThresholder(1).threshold(pruned.label);
        binaryImage = new DilationFilter(1).filter(binaryImage);
        KmlWriter.write(binaryImage, out, "thresh", PngWriter.createCoolToWarmColormap());
        LatLonGrid gradient = new LoGEdgeFilter(2,1).edgeFilter(binaryImage);
        KmlWriter.write(gradient, out, "gradient", PngWriter.createCoolToWarmColormap());

        // for each city, initialize a snake
        SnakeActiveContour alg = new SnakeActiveContour(gradient);
        int numiter = 30;
        for (int i=1; i < prop.length; ++i){
            double cx = prop[i].getCx();
            double cy = prop[i].getCy();
            // square box enclosing the center point that is larger than core area
            double initsize = 3 * Math.sqrt( prop[i].getSize() );
            SnakeNode[] snakepts = new SnakeNode[4];
            snakepts[0] = new SnakeNode(cx + initsize, cy - initsize, grid.getNumLat(), gr
            snakepts[1] = new SnakeNode(cx + initsize, cy + initsize, grid.getNumLat(), gr
            snakepts[2] = new SnakeNode(cx - initsize, cy + initsize, grid.getNumLat(), gr
            snakepts[3] = new SnakeNode(cx - initsize, cy - initsize, grid.getNumLat(), gr

            SnakeNode[] snake = alg.moveSnake(snakepts, numiter);
            // mark snake points on grid
            for (int k=0; k < snake.length; ++k){
                grid.setValue(snake[k].getX(), snake[k].getY(), 1000);
            }
        }

        // write out original grid with snake points marked
        KmlWriter.write(grid, out, "snakes", PngWriter.createCoolToWarmColormap());
    }
```

6.106.3.2 SnakeNode [ ] edu.ou.asgbook.segmentation.SnakeActiveContour.moveSnake (SnakeNode[ ] *pixels*, int *numIter*)

Provide an initial guess of points.

```
                                                                     {
        Snake snake = new Snake(pixels);
        snake = resample(snake); // get it to desired length, then start moving it
        snake = moveSnake(snake, numIter);
        return complete(snake.pts);
    }
```

6.106.3.3 Snake edu.ou.asgbook.segmentation.SnakeActiveContour.pruneAndResample (Snake *inputSnake*)

```
                                                {
        List<SnakeNode> nodes = new ArrayList<SnakeNode>(Arrays.asList(inputSnake.pts));
        int numNodes = nodes.size();
        if (numNodes <= 3) return inputSnake;


        int dist_thresh = SNAKE_DIST_BETWEEN_PTS * SNAKE_DIST_BETWEEN_PTS;
        for (int i = 0; i < numNodes; i++) {
            SnakeNode curPt = nodes.get(i);
            int next = i + 1; if (next == numNodes) next = 0;
            SnakeNode nextPt = nodes.get(next);

            int distsq = (nextPt.getX()-curPt.getX())*(nextPt.getX()-curPt.getX())+(nextPt.getX()-curP
            boolean currNotOnGradient = gradient.getData()[curPt.getX()][curPt.getY()] < 30;
            boolean nextNotOnGradient = gradient.getData()[nextPt.getX()][nextPt.getY()] < 30;

            boolean remove = numNodes > SNAKE_LENGTH && ( (distsq < 20) || (distsq < 80 && nextNotOnGr
            if (remove) {
                nodes.remove(next);
                --numNodes;
                --i; // retry this node
            } else if (distsq > dist_thresh && (currNotOnGradient || nextNotOnGradient)) {
                SnakeNode newPt = new SnakeNode((curPt.getX()+nextPt.getX())/2,(curPt.getY()+nextPt.ge
                nodes.add(i+1, newPt);
                numNodes++;
            }
        }
        return new Snake(nodes.toArray(new SnakeNode[0]));
    }
```

6.106.3.4 Snake edu.ou.asgbook.segmentation.SnakeActiveContour.resample (Snake *inputSnake*)

```
                                        {
        return pruneAndResample(inputSnake);
    }
```

6.106.3.5 Snake edu.ou.asgbook.segmentation.SnakeActiveContour.resampleNodes (Snake *inputSnake*)

```
                                             {
        SnakeNode[] full = complete(inputSnake.pts);
        if ( full.length <= SNAKE_LENGTH ) return inputSnake;
        System.out.println("Resampling " + full.length + " to " + SNAKE_LENGTH);
        int sampleInterval = full.length / SNAKE_LENGTH;
        SnakeNode[] sampled = new SnakeNode[SNAKE_LENGTH];
        for (int i=0; i < sampled.length; ++i){
            sampled[i] = full[i*sampleInterval];
        }
        return new Snake(sampled);
    }
```

6.107 edu.ou.asgbook.segmentation.SnakeActiveContour.Snake Class Reference

Public Member Functions

- Snake (SnakeNode[ ] pts)
- SnakeNode get (int k)

  the snake is a closed curve, so does modulo to get points

- SnakeNode[ ] getNodes ()
- double computeEnergy (int candx, int candy, SnakeNode current, SnakeNode previous, SnakeNode next)

6.107.1 Constructor & Destructor Documentation

6.107.1.1 edu.ou.asgbook.segmentation.SnakeActiveContour.Snake.Snake (SnakeNode[ ] *pts*)

```
                                 {
        this.pts = pts;
        meanDistBetweenPts = 0;
        if ( this.pts.length == 0 ) return;

        // compute mean dist
        for (int i=0; i < pts.length; ++i){
            SnakeNode curr = pts[i];
            SnakeNode next = pts[ (i+1)%(pts.length) ];
            meanDistBetweenPts += Math.sqrt( curr.getDistanceSquared(next) );
        }
        meanDistBetweenPts /= pts.length;
    }
```

6.107.2 Member Function Documentation

6.107.2.1 double edu.ou.asgbook.segmentation.SnakeActiveContour.Snake.computeEnergy (int *candx*, int *candy*, SnakeNode *current*, SnakeNode *previous*, SnakeNode *next*)

```

        double E_total, E_edgestrength, E_smoothness, E_continuity ;
        E_edgestrength = gradient.getData()[candx][candy];
        E_smoothness = Math.pow(previous.getX() - 2 * candx + next.getX(), 2) + Math.po
        E_continuity = Math.abs( Math.sqrt(previous.getDistanceSquared(candx,candy)) -
        E_total =  current.alpha * E_continuity + current.beta * E_smoothness - current
        return E_total;
    }
```

6.107.2.2 SnakeNode edu.ou.asgbook.segmentation.SnakeActive-Contour.Snake.get (int *k*)

the snake is a closed curve, so does modulo to get points

```
                                    {
                int len = pts.length;
                while ( k < 0 ){
                    k += len;
                }
                return pts[k%len];
            }
```

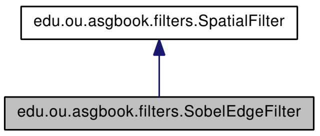
6.107.2.3 SnakeNode [ ] edu.ou.asgbook.segmentation.SnakeActive-Contour.Snake.getNodes ()

```
                                      {
                return pts;
            }
```

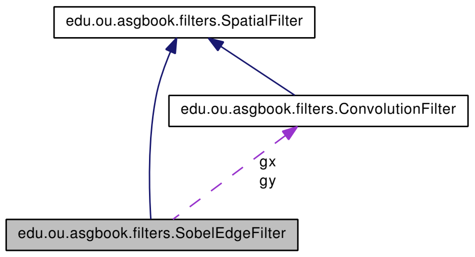
6.108 edu.ou.asgbook.filters.SobelEdgeFilter Class Reference

Find edges in a grid.

Inheritance diagram for edu.ou.asgbook.filters.SobelEdgeFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.SobelEdgeFilter

Collaboration diagram for edu.ou.asgbook.filters.SobelEdgeFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.ConvolutionFilter
gx
gy
edu.ou.asgbook.filters.SobelEdgeFilter

Public Member Functions

- SobelEdgeFilter ()
- LatLonGrid saturate (final LatLonGrid input, int thresh)

    treat values > thresh as equal to thresh

- Override LatLonGrid filter (LatLonGrid input)
- LatLonGrid edgeFilter (final LatLonGrid input)
- LatLonGrid edgeFilter (final LatLonGrid input, File out)

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.108.1 Detailed Description

Find edges in a grid.

Author:

valliappa.lakshmanan

6.108.2 Constructor & Destructor Documentation

6.108.2.1 edu.ou.asgbook.filters.SobelEdgeFilter.SobelEdgeFilter ()

```
                             {
        double[][] xc = new double[3][3];
        double[][] yc = new double[3][3];
        xc[0][0] = xc[0][2] = yc[0][0] = yc[2][0] = -1;
        xc[0][1] = yc[1][0] = -2;
        xc[2][0] = xc[2][2] = yc[0][2] = yc[2][2] = 1;
        xc[2][1] = yc[1][2] = 2;
        gx = new ConvolutionFilter(xc);
        gy = new ConvolutionFilter(yc);
    }
```

6.108.3 Member Function Documentation

6.108.3.1 LatLonGrid edu.ou.asgbook.filters.SobelEdgeFilter.edgeFilter (final LatLonGrid *input*, File *out*)

```
                                                                    {
        LatLonGrid g1 = gx.convolve(input);
        KmlWriter.debugWrite(g1, out, "gx");
        LatLonGrid g2 = gy.convolve(input);
        KmlWriter.debugWrite(g2, out, "gy");
        for (int i=0; i < g1.getNumLat(); ++i){
            for (int j=0; j < g1.getNumLon(); ++j){
                if (g1.getValue(i,j) != g1.getMissing() && g2.getValue(i,j) != g2.getMissing()){
                    int gradient = Math.abs( g1.getValue(i, j) ) + Math.abs( g2.getValue(i,j) );
                    g1.setValue(i, j, gradient);
                } else {
                    g1.setValue(i,j, g1.getMissing());
                }
            }
        }
        return g1;
    }
```

6.108.3.2 LatLonGrid edu.ou.asgbook.filters.SobelEdgeFilter.edgeFilter (final LatLonGrid *input*)

```
                                                          {
        return edgeFilter(input, null);
    }
```

6.108.3.3 Override LatLonGrid edu.ou.asgbook.filters.SobelEdgeFilter.filter (LatLonGrid *input*)

```
                                                {
        return edgeFilter(input);
    }
```

6.108.3.4 static void edu.ou.asgbook.filters.SobelEdgeFilter.main (String[ ] *args*) throws Exception [static]

```
                                                            {
        // create output directory
        File out = OutputDirectory.getDefault("sobel");

        // read input
        DataTransform t = new GlobalPopulation.LogScaling();
        // DataTransform t = new GlobalPopulation.LinearScaling();
        LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, t).crop
        KmlWriter.write(popdensity, out, "orig", PngWriter.createCoolToWarmColormap());

        SobelEdgeFilter filter = new SobelEdgeFilter();
        LatLonGrid edges = filter.edgeFilter(popdensity, out);
        KmlWriter.write(edges, out, "edge", PngWriter.createCoolToWarmColormap());

        ConvolutionFilter sm = new ConvolutionFilter(ConvolutionFilter.gauss(11, 11));
        popdensity = sm.smooth(popdensity);
        edges = filter.edgeFilter(popdensity, null);
        KmlWriter.write(edges, out, "smoothedge", PngWriter.createCoolToWarmColormap());

        LatLonGrid saturated = filter.saturate(edges, 5000);
        KmlWriter.write(saturated, out, "saturated", PngWriter.createCoolToWarmColormap());
    }
```

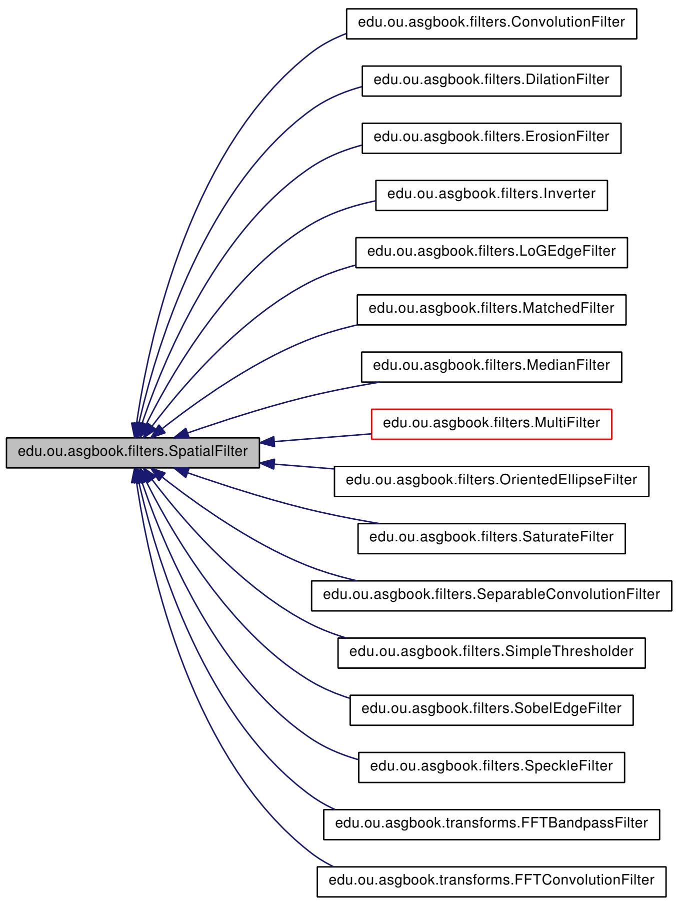
6.108.3.5 LatLonGrid edu.ou.asgbook.filters.SobelEdgeFilter.saturate (final LatLonGrid *input*, int *thresh*)

treat values > thresh as equal to thresh

```
                                                                  {
        LatLonGrid result = LatLonGrid.copyOf(input);
        for (int i=0; i < input.getNumLat(); ++i){
            for (int j=0; j < input.getNumLon(); ++j){
                if ( result.getValue(i, j) > thresh ){
                    result.setValue(i,j, thresh);
                }
            }
        }
        return result;
    }
```

6.109 edu.ou.asgbook.filters.SpatialFilter Interface Reference

Inheritance diagram for edu.ou.asgbook.filters.SpatialFilter:


edu.ou.asgbook.filters.ConvolutionFilter
edu.ou.asgbook.filters.DilationFilter
edu.ou.asgbook.filters.ErosionFilter
edu.ou.asgbook.filters.Inverter
edu.ou.asgbook.filters.LoGEdgeFilter
edu.ou.asgbook.filters.MatchedFilter
edu.ou.asgbook.filters.MedianFilter
edu.ou.asgbook.filters.MultiFilter
edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.OrientedEllipseFilter
edu.ou.asgbook.filters.SaturateFilter
edu.ou.asgbook.filters.SeparableConvolutionFilter
edu.ou.asgbook.filters.SimpleThresholder
edu.ou.asgbook.filters.SobelEdgeFilter
edu.ou.asgbook.filters.SpeckleFilter
edu.ou.asgbook.transforms.FFTBandpassFilter
edu.ou.asgbook.transforms.FFTConvolutionFilter

Public Member Functions

- LatLonGrid filter (final LatLonGrid input)

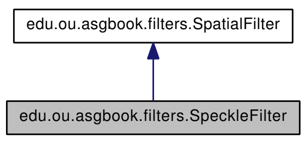
6.109.1 Member Function Documentation

6.109.1.1 LatLonGrid edu.ou.asgbook.filters.SpatialFilter.filter (final LatLonGrid *input*)

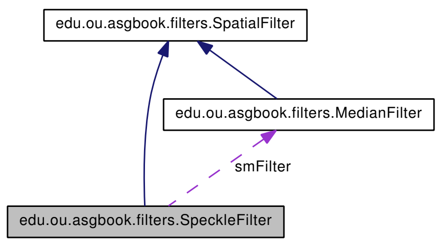
6.110 edu.ou.asgbook.filters.SpeckleFilter Class Reference

Denoising filter that removes speckle.

Inheritance diagram for edu.ou.asgbook.filters.SpeckleFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.SpeckleFilter

Collaboration diagram for edu.ou.asgbook.filters.SpeckleFilter:


edu.ou.asgbook.filters.SpatialFilter
edu.ou.asgbook.filters.MedianFilter
smFilter
edu.ou.asgbook.filters.SpeckleFilter

Public Member Functions

- SpeckleFilter (int halfSize, int maxChange)
- Override LatLonGrid filter (LatLonGrid input)
- LatLonGrid speckleFilter (final LatLonGrid input)

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.110.1 Detailed Description

Denoising filter that removes speckle.

Author:

Valliappa.Lakshmanan

6.110.2 Constructor & Destructor Documentation

6.110.2.1 edu.ou.asgbook.filters.SpeckleFilter.SpeckleFilter (int *halfSize*, int *maxChange*)

```
                                                      {
        this.smFilter = new MedianFilter(halfSize);
        this.maxChange = maxChange;
    }
```

6.110.3 Member Function Documentation

6.110.3.1 Override LatLonGrid edu.ou.asgbook.filters.SpeckleFilter.filter (LatLonGrid *input*)

```
                                               {
        return speckleFilter(input);
    }
```

6.110.3.2 static void edu.ou.asgbook.filters.SpeckleFilter.main (String[ ] *args*) throws Exception [static]

```
                                                            {
        // create output directory
        File out = OutputDirectory.getDefault("speckle");

        // read input
        LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new Gl
        KmlWriter.write(popdensity, out, "orig", PngWriter.createCoolToWarmColormap());

        // add noise
        Random rand = new Random();
        final int NOISE = 200;
        for (int i=0; i < 1000; ++i){
            int x = rand.nextInt(popdensity.getNumLat());
            int y = rand.nextInt(popdensity.getNumLon());
            int add = NOISE + rand.nextInt(NOISE/2);
            popdensity.setValue(x, y, popdensity.getValue(x,y) + add);
        }
        KmlWriter.write(popdensity, out, "noisy", PngWriter.createCoolToWarmColormap());

        // dilate
        LatLonGrid dilate1 = new SpeckleFilter(3,NOISE).filter(popdensity);
        KmlWriter.write(dilate1, out, "speckle_3", PngWriter.createCoolToWarmColormap());
    }
```

6.110.3.3 LatLonGrid edu.ou.asgbook.filters.SpeckleFilter.speckleFilter (final LatLonGrid *input*)

```
                                                          {
        LatLonGrid smoothed = smFilter.filter(input);
        LatLonGrid output = LatLonGrid.copyOf(input);
        int[][] inData = input.getData();
        int[][] smData = smoothed.getData();
        int nx = inData.length;
        int ny = inData[0].length;
        for (int i=0; i < nx; ++i){
            for (int j=0; j < ny; ++j){
                if (inData[i][j] != input.getMissing() &&
                    smData[i][j] != smoothed.getMissing()){
                    int diff = Math.abs(inData[i][j] - smData[i][j]);
                    if (diff > maxChange){ // noise
                        output.setValue(i,j, smData[i][j]);
                    }
                }
            }
        }
        return output;
    }
```

6.111 edu.ou.asgbook.usage.Sprawl Class Reference

Solution to a classroom assignment to identify regions of urban sprawl from the population density data.

Static Public Member Functions

- static void runOnPopDensity (boolean crop) throws Exception
- static void findSprawl (LatLonGrid grid1, LatLonGrid grid2, File out) throws Exception
- static void main (String[ ] args) throws Exception

6.111.1 Detailed Description

Solution to a classroom assignment to identify regions of urban sprawl from the population density data.

Author:

v.lakshmanan

6.111.2 Member Function Documentation

6.111.2.1 static void edu.ou.asgbook.usage.Sprawl.findSprawl (LatLonGrid *grid1*, LatLonGrid *grid2*, File *out*) throws Exception [static]

```

        // write out input grids
        KmlWriter.write(grid1, out, "pop_1990", PngWriter.createCoolToWarmColormap());
        KmlWriter.write(grid2, out, "pop_2010", PngWriter.createCoolToWarmColormap());
        System.out.println("Resolution of image = " + grid1.getLatRes() + "x" + grid1.getLo
        // Find optimal threshold on 2010 data using Otsu's method
        final int MIN = 0;
        final int MAX = 400;
        final int incr = 1;
        Histogram hist = new Histogram(MIN, incr, (MAX-MIN)/incr );
        hist.update(grid2);
        OtsuThresholdSelector thresholder = new OtsuThresholdSelector(hist);
        int thresh1 = thresholder.getOptimalThreshold();
        System.out.println("Optimal threshold=" + thresh1);

        // A city consists of point with this threshold and contiguous points > some reasona
        int thresh2 = thresh1 * 2;
        LabelResult label1990 =  new HysteresisSegmenter(thresh1, thresh2).label(grid1);
        KmlWriter.write(label1990.label, out, "label_1990", PngWriter.createRandomColormap(

        LabelResult label2010 =  new HysteresisSegmenter(thresh1, thresh2).label(grid2);
        KmlWriter.write(label2010.label, out, "label_2010", PngWriter.createRandomColormap(

        // grow regions and find region properties
        RegionProperty[] props1 = RegionProperty.compute(label1990, grid1);
        RegionProperty[] props2 = RegionProperty.compute(label2010, grid2);

        // create a new grid that consists of city sizes
        LatLonGrid citysize1990 = getCitySize(label1990, props1);
        LatLonGrid citysize2010 = getCitySize(label2010, props2);
        KmlWriter.write(citysize1990, out, "citysize_1990", PngWriter.createCoolToWarmColormap());
        KmlWriter.write(citysize2010, out, "citysize_2010", PngWriter.createCoolToWarmColormap());

        // compute and write out difference in size for every 2010 city
        LatLonGrid changeInSize = LatLonGrid.copyOf(citysize2010);
        Pixel[] sizechange = new Pixel[props2.length];
        for (int i=0; i < sizechange.length; ++i){
            sizechange[i] = new Pixel(0, 0,0);
        }
        for (int i=0; i < citysize2010.getNumLat(); ++i){
            for (int j=0; j < citysize2010.getNumLon(); ++j){
                 int sz1 = citysize1990.getValue(i,j);
                 int sz2 = citysize2010.getValue(i,j);
                 int percentChange = 0;
                 if (sz1 > 5 && sz2 > 5){ // reasonably big?
                     percentChange = (100 * (sz2 - sz1)) / sz1;
                     sizechange[ label2010.label.getValue(i,j) ] = new Pixel( i,j, percentChange);
                 }
                 changeInSize.setValue(i,j, percentChange);
             }
         }
         KmlWriter.write(changeInSize, out, "sprawl_1990_2010", PngWriter.createCoolToWarmColormap());

         // Print out the top cities
         Arrays.sort(sizechange, new Comparator<Pixel>(){
             @Override
             public int compare(Pixel arg0, Pixel arg1) {
                 return arg0.getValue() - arg1.getValue();
             }
         });
         for (int i=Math.max(0,sizechange.length-20); i < sizechange.length; ++i){
             LatLon loc = citysize2010.getLocation(sizechange[i].getX(), sizechange[i].getY());
             System.out.println(  loc +
                     " : " + citysize1990.getValue(sizechange[i]) +
                     " to " + citysize2010.getValue(sizechange[i]) +
                     "  " + getCityNear(loc)
             );
         }

         // Shrunk?
         System.out.println("Cities that have exhibited the least spatial growth:");
         for (int i=0; i < Math.min(20,sizechange.length); ++i){
             LatLon loc = citysize2010.getLocation(sizechange[i].getX(), sizechange[i].getY());
             System.out.println(  loc +
                     " : " + citysize1990.getValue(sizechange[i]) +
                     " to " + citysize2010.getValue(sizechange[i]) +
                     "  " + getCityNear(loc)
             );
         }
```

```
    }
```

6.111.2.2 static void edu.ou.asgbook.usage.Sprawl.main (String[ ] *args*) throws Exception [static]

```
                                                            {
        boolean crop = false; // if false, on USA; if true, on NYC area
        runOnPopDensity(crop);
    }
```

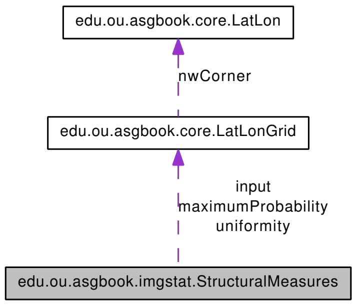
6.111.2.3 static void edu.ou.asgbook.usage.Sprawl.runOnPopDensity (boolean *crop*) throws Exception [static]

```
                                                                   {
        LatLonGrid popdensity1 = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA1990, ne
        LatLonGrid popdensity2 = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new Gl

        int x1, x2, x3, x4;
        if (crop){
            x1 = 900;
            x2 = 2500;
            x3 = 200;
            x4 = 200;
        } else {
            LatLon nwCorner = new LatLon(60, -130);
            LatLon seCorner = new LatLon(12, -52);
            x1 = popdensity1.getRow(nwCorner);
            x2 = popdensity1.getCol(nwCorner);
            x3 = popdensity1.getRow(seCorner) - popdensity1.getRow(nwCorner);
            x4 = popdensity1.getCol(seCorner) - popdensity1.getCol(nwCorner);
        }
        popdensity1 = popdensity1.crop(x1, x2, x3, x4);
        popdensity2 = popdensity2.crop(x1, x2, x3, x4);

        File out = OutputDirectory.getDefault("sprawl");

        findSprawl(popdensity1, popdensity2, out);
    }
```

6.112 edu.ou.asgbook.imgstat.StructuralMeasures Class Reference

Statistics computed in the neighborhood of a pixel.

Collaboration diagram for edu.ou.asgbook.imgstat.StructuralMeasures:


edu.ou.asgbook.core.LatLon
nwCorner
edu.ou.asgbook.core.LatLonGrid
input
maximumProbability
uniformity
edu.ou.asgbook.imgstat.StructuralMeasures

Public Member Functions

- StructuralMeasures (LatLonGrid input, int Nx, int Ny, int min, int incr, int bins)
- LatLonGrid[ ] getUniformity ()
- LatLonGrid[ ] getMaximumProbability ()

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.112.1 Detailed Description

Statistics computed in the neighborhood of a pixel.

Author:

valliappa.lakshmanan

6.112.2 Constructor & Destructor Documentation

6.112.2.1 edu.ou.asgbook.imgstat.StructuralMeasures.StructuralMeasures (LatLonGrid *input*, int *Nx*, int *Ny*, int *min*, int *incr*, int *bins*)

```
27                                                                            {
        this.hx = Nx/2;
        this.hy = Ny/2;
        this.input = input;
        this.min = min;
        this.incr = incr;
        this.bins = bins;
        for (int i=0; i < uniformity.length; ++i){
            this.uniformity[i] = LatLonGrid.copyOf(input);
            this.maximumProbability[i] = LatLonGrid.copyOf(input);
        }
        this.compute();
    }
```

6.112.3 Member Function Documentation

6.112.3.1 LatLonGrid [ ] edu.ou.asgbook.imgstat.StructuralMeasures.getMaximumProbability ()

```
                                                {
        return maximumProbability;
    }
```

6.112.3.2 LatLonGrid [ ] edu.ou.asgbook.imgstat.StructuralMeasures.getUniformity ()

```
                                        {
        return uniformity;
    }
```

6.112.3.3 static void edu.ou.asgbook.imgstat.StructuralMeasures.main (String[ ] *args*) throws Exception [static]

```
                                                             {
        // log-scaled population density
        LatLonGrid popdensity = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new Glo
        popdensity = popdensity.crop(900, 2500, 200, 200);
        File out = OutputDirectory.getDefault("glcmstat");

        KmlWriter.write(popdensity, out, "popdensity", PngWriter.createCoolToWarmColormap())
        StructuralMeasures stat = new StructuralMeasures(popdensity, 11, 11, 100, 500, 50);
        for (int i=0; i < Direction.values().length; ++i){
            KmlWriter.write(stat.getUniformity()[i], out, "uniformity_" + Direction.values()
        }
        for (int i=0; i < Direction.values().length; ++i){
            KmlWriter.write(stat.getMaximumProbability()[i], out, "maxprob_" + Direction.val
        }
    }
```

6.113 edu.ou.asgbook.dataset.SurfaceAlbedo Class Reference

Reads lambert-conformal ascii grid.

Static Public Member Functions

- static LatLonGrid read (Reader inputFile, int scaling)
- static LatLonGrid read (File file, int scaling) throws IOException

  reads data from a File.

- static void main (String[ ] args) throws Exception

Static Public Attributes

- static File CONUS = new File("data/sfcalbedo/sfcalbedo.txt.gz")

6.113.1 Detailed Description

Reads lambert-conformal ascii grid.

Author:

v.lakshmanan

6.113.2 Member Function Documentation

6.113.2.1 static void edu.ou.asgbook.dataset.SurfaceAlbedo.main (String[ ] *args*) throws Exception [static]

```
                                                            {
        // create output directory
        File out = OutputDirectory.getDefault("sfcalbedo");

        // read input
        LatLonGrid conus = SurfaceAlbedo.read(SurfaceAlbedo.CONUS, 100);

        // write out as image, for viewing
        KmlWriter.write(conus, out, "sfcalbedo", PngWriter.createCoolToWarmColormap());
    }
```

6.113.2.2 static LatLonGrid edu.ou.asgbook.dataset.SurfaceAlbedo.read (File *file*, int *scaling*) throws IOException [static]

reads data from a File.

The File can be gzipped or uncompressed.

```
                                                                                    {
        Reader f = null;
        System.out.println("Reading " + file.getAbsolutePath());
        if (file.getAbsolutePath().endsWith(".gz")) {
            f = new InputStreamReader(new GZIPInputStream(new FileInputStream(
                    file)));
        } else {
            f = new FileReader(file);
        }
        return read(f, scaling);
    }
```

6.113.2.3 static LatLonGrid edu.ou.asgbook.dataset.SurfaceAlbedo.read (Reader *inputFile*, int *scaling*) [static]

```
                                                                           {
        Scanner s = null;
        try {
            s = new Scanner(inputFile);

            // read header
            @SuppressWarnings("unused")
            String junk;
            junk = s.next(); String ellipsoid = s.next();
            junk = s.next(); String projection = s.next();
            if (! (ellipsoid.equals("WGS-84") && projection.equals("LAMBERT2SP"))){
                throw new IllegalArgumentException("Expect data to be in LAMBERT2SP and WGS-
            }
            junk = s.next(); double lat1 = s.nextDouble();
            junk = s.next(); double lat2 = s.nextDouble();
            junk = s.next(); double center_lat = s.nextDouble();
            junk = s.next(); double center_lon = s.nextDouble();
            junk = s.next(); double eastres = s.nextDouble(); // meters
            junk = s.next(); double northres = s.nextDouble();
            junk = s.next(); int nrows = s.nextInt();
            junk = s.next(); int ncols = s.nextInt();

            double center_northing = - nrows * 0.5 * northres;
            double center_easting = ncols * 0.5 * eastres;

            int missing = -999; // doesn't exist in the data

            // read in data (in Lambert projection)
            int[][] lamdata = new int[nrows][ncols];
            for (int i=0; i < nrows; ++i){
                for (int j=0; j < ncols; ++j){
                    try {
                        double value = s.nextDouble();
                        lamdata[i][j] = (int)(0.5 + value * scaling);
                    } catch (Exception e){
                        lamdata[i][j] = missing;
                    }
                }
            }

            // Find grid extent
            LambertConformal2SP proj = new LambertConformal2SP(Ellipsoid.WGS84(), new LatLon(center_lat
            double minlat = 180; double maxlat = -180;
            double minlon = 180; double maxlon = -180;
            for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
                LambertConformal2SP.Coord lam = new LambertConformal2SP.Coord(-(i+0.5)*northres, (j+0.5
                LatLon ll = proj.getLatLon(lam);
                if ( ll.getLat() < minlat ) minlat = ll.getLat();
                if ( ll.getLat() > maxlat ) maxlat = ll.getLat();
                if ( ll.getLon() < minlon ) minlon = ll.getLon();
                if ( ll.getLon() > maxlon ) maxlon = ll.getLon();
            }
            System.out.println("Grid extent: " + minlat + " " + minlon + " " + maxlat + " " + maxlon);

            // best latres, lonres
            int outrows = nrows;
            int outcols = ncols;
            double latres = (maxlat - minlat)/outrows;
            double lonres = (maxlon - minlon)/outcols;
            // lookup nearest neighbor in lat-lon space
            int[][] lldata = new int[outrows][outcols];
            for (int i=0; i < outrows; ++i){
                double lat = maxlat - i * latres;
                for (int j=0; j < outcols; ++j){
                    double lon = minlon + j * lonres;
                    LambertConformal2SP.Coord lam = proj.getLambert( new LatLon(lat,lon) );
                     double rowno = (0 - lam.northing)/northres;
                     rowno = nrows - rowno - 1; // row=0 is southmost row
                     double colno = (lam.easting - 0)/eastres;
                     // lldata[i][j] = Remapper.nearestNeighbor(rowno, colno, lamdata, missing);
                     lldata[i][j] = Remapper.bilinearInterpolation(rowno, colno, lamdata, missing);
                 }
             }

             return new LatLonGrid(lldata, missing, new LatLon(maxlat,minlon), latres, lonres);
        } catch (Exception e){
            System.err.println("Error reading file: " + e);
            throw new IllegalArgumentException(e);
        } finally {
            if (s != null) {
                try{
                    s.close();
                } catch (Exception e){
                    // okay
                }
            }
        }
    }
```

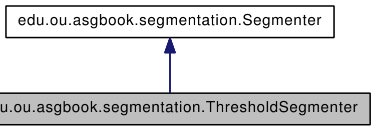
6.113.3 Member Data Documentation

6.113.3.1 File edu.ou.asgbook.dataset.SurfaceAlbedo.CONUS = new File("data/sfcalbedo/sfcalbedo.txt.gz") `[static]`

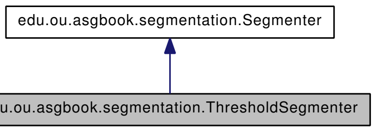
6.114 edu.ou.asgbook.segmentation.ThresholdSegmenter Class Reference

Simple object identification based on a single threshold.

Inheritance diagram for edu.ou.asgbook.segmentation.ThresholdSegmenter:


edu.ou.asgbook.segmentation.Segmenter
edu.ou.asgbook.segmentation.ThresholdSegmenter

Collaboration diagram for edu.ou.asgbook.segmentation.ThresholdSegmenter:


edu.ou.asgbook.segmentation.Segmenter
edu.ou.asgbook.segmentation.ThresholdSegmenter

Public Member Functions

- ThresholdSegmenter (int thresh)
- LabelResult label (LatLonGrid data)

  Creates a labeled grid where background pixels are set to 0 and labels for objects go 1,2,3.

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.114.1 Detailed Description

Simple object identification based on a single threshold.

Author:

v.lakshmanan

6.114.2 Constructor & Destructor Documentation

6.114.2.1 edu.ou.asgbook.segmentation.ThresholdSegmenter.ThresholdSegmenter (int *thresh*)

```
                                        {
        super();
        this.thresh = thresh;
    }
```

6.114.3 Member Function Documentation

6.114.3.1 LabelResult edu.ou.asgbook.segmentation.ThresholdSegmenter.label (LatLonGrid *data*) [virtual]

Creates a labeled grid where background pixels are set to 0 and labels for objects go 1,2,3.

.. All pixels > thresh are part of an object.

Implements edu.ou.asgbook.segmentation.Segmenter.

```
                                          {
        final int UNSET = 0;
        int nrows = data.getNumLat();
        int ncols = data.getNumLon();
        LatLonGrid label = new LatLonGrid(nrows,ncols,0,data.getNwCorner(),data.getLatRes(),
        // label.fill(UNSET); java default is to zero-out arrays
        int regno = 0;
        for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
            if ( data.getValue(i, j) > thresh && label.getValue(i, j) == UNSET ){
                ++regno;
                RegionGrowing.growRegion(i,j, data, thresh, label, regno);
            }
        }
        System.out.println("Found " + (regno+1) + " objects");
        return new LabelResult(label, regno);
    }
```

6.114.3.2 static void edu.ou.asgbook.segmentation.ThresholdSegmenter.main (String[ ] *args*) throws Exception [static]

```
                                                          {
        File out = OutputDirectory.getDefault("regiongrowing");

        // data
        LatLonGrid grid = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new GlobalPop
        KmlWriter.write(grid, out, "orig", PngWriter.createCoolToWarmColormap());
        KmlWriter.write(GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new GlobalPopul

        // global thresh
        for (int thresh = 10; thresh <= 30; thresh += 10){
            KmlWriter.write(new SimpleThresholder(thresh).threshold(grid), out, "thresh_"+thresh, PngWr
            ThresholdSegmenter seg = new ThresholdSegmenter(thresh);
            LatLonGrid label = seg.label(grid).label;
            // label.setMissing(-1); // so background is present
            KmlWriter.write(label, out, "cities_"+thresh, PngWriter.createRandomColormap());
        }
    }
```

6.115 edu.ou.asgbook.geocode.UsaZipcode Class Reference

Find the city for each zipcode in the USA.

Collaboration diagram for edu.ou.asgbook.geocode.UsaZipcode:

Public Member Functions

- Entry getEntryClosestTo (LatLon loc)

Static Public Member Functions

- static UsaZipcode getInstance ()
- static void main (String[ ] args)

Classes

- class **Entry**

6.115.1 Detailed Description

Find the city for each zipcode in the USA.

Author:

v.lakshmanan

6.115.2 Member Function Documentation

6.115.2.1 Entry edu.ou.asgbook.geocode.UsaZipcode.getEntryClosestTo (LatLon *loc*)

```
                                             {
        double mindistsq = 0.5*0.5; // within 50 km
        Entry best = null;
        for (Entry e : entries){
            double dist_lat = e.location.getLat() - loc.getLat();
            double dist_lon = e.location.getLon() - loc.getLon();
            double dist_sq = dist_lat*dist_lat + dist_lon*dist_lon;
            if ( dist_sq < mindistsq ){
                mindistsq = dist_sq;
                best = e;
            }
        }
        return best;
    }
```

6.115.2.2 static UsaZipcode edu.ou.asgbook.geocode.UsaZipcode.getInstance () [static]

```
                                          {
        return instance;
    }
```

6.115.2.3 static void edu.ou.asgbook.geocode.UsaZipcode.main (String[ ] *args*) [static]

```
                                           {
        Entry e = UsaZipcode.getInstance().getEntryClosestTo(new LatLon(18.31,-66.06));
        System.out.println(e);

        e = UsaZipcode.getInstance().getEntryClosestTo(new LatLon(35.2,-97.4));
        System.out.println(e);
    }
```

6.116 edu.ou.asgbook.imgstat.VectorQuantizer Class Reference

Develops a quantization scheme using vector quantization.

Public Member Functions

- VectorQuantizer (LatLonGrid[ ] params, int K)
- int getBinNumber (LatLonGrid[ ] params, int x, int y)
- Vector getCenterValue (int bin_no)
- LatLonGrid band (LatLonGrid[ ] params)

  replaces each pixel by the bin number it belongs to.
- Override String toString ()

Static Public Member Functions

- static LatLonGrid[ ] normalize (LatLonGrid[ ] params)
- static LatLonGrid normalize (LatLonGrid data)

  The output grid ranges from 0 to 100.
- static void main (String[ ] args) throws Exception

Classes

- class **Vector**

6.116.1 Detailed Description

Develops a quantization scheme using vector quantization.

Author:

valliappa.lakshmanan

6.116.2 Constructor & Destructor Documentation

6.116.2.1 edu.ou.asgbook.imgstat.VectorQuantizer.VectorQuantizer (LatLonGrid[ ] *params*, int *K*)

Parameters:

params where to get the vectors from. These grids should be normalized.

K

```
                                                        {
         int nrows = params[0].getNumLat();
         int ncols = params[0].getNumLon();
         // 1. initialize centroid with mean
         centroids = new Vector[1];
         centroids[0] = new Vector(params.length); // zero
         for (int p=0; p < params.length; ++p){
             int N = 0;
             for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
                 int val = params[p].getValue(i,j);
                 if ( val != params[p].getMissing() ){
                     centroids[0].values[p] += val;
                     ++N;
                 }
             }
             if (N > 0){
                 centroids[0].values[p] /= N;
             }
         }
         System.out.println(this);
         while ( centroids.length < K ){
             // 2. split the centroids
             final double epsilon = 0.1;
             centroids = split(centroids, epsilon);
             // System.out.println(this);
             // 3. update centroid
             centroids = computeCentroids(params);
             // System.out.println(this);
         }

         System.out.println(this);
     }
```

6.116.3 Member Function Documentation

6.116.3.1 LatLonGrid edu.ou.asgbook.imgstat.VectorQuantizer.band (LatLonGrid[ ] *params*)

replaces each pixel by the bin number it belongs to.

```
                                                {
         LatLonGrid result = LatLonGrid.copyOf(params[0]);
```

```
        result.setMissing(0);
        int nrows = result.getNumLat();
        int ncols = result.getNumLon();
        for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
            int bin_no = getBinNumber(params, i, j);
            result.setValue(i,j, bin_no+1);
        }
        return result;
    }
```

6.116.3.2 int edu.ou.asgbook.imgstat.VectorQuantizer.getBinNumber (LatLonGrid[ ] *params*, int *x*, int *y*)

```
                                                                    {
        // closest centroid wins
        int best = 0;
        double mindist = centroids[0].computeDist(params, x, y);
        for (int p=1; p < centroids.length; ++p){
            double dist = centroids[p].computeDist(params, x, y);
            if (dist < mindist){
                mindist = dist;
                best = p;
            }
        }
        return best;
    }
```

6.116.3.3 Vector edu.ou.asgbook.imgstat.VectorQuantizer.getCenterValue (int *bin_no*)

```
                                        {
        return centroids[bin_no];
    }
```

6.116.3.4 static void edu.ou.asgbook.imgstat.VectorQuantizer.main (String[ ] *args*) throws Exception [static]

```
                                                                  {
        // create output directory
        File outdir = OutputDirectory.getDefault("quantizer");

        // read input
        LatLonGrid conus = SurfaceAlbedo.read(SurfaceAlbedo.CONUS, 100);

        // compute a few local and texture measures
        LocalMeasures local = new LocalMeasures(conus, 11, 11);
        LatLonGrid mean = local.getMean();
        // LatLonGrid stdev = local.getStdDeviation();
        Histogram hist = HistogramBinSelection.createBasedOnRange(conus);
        StructuralMeasures texture = new StructuralMeasures(conus, 11, 11, hist.getMin(), hist.getIncr
        LatLonGrid uniformity = texture.getUniformity()[0];

        LatLonGrid[] params = new LatLonGrid[]{ conus, mean, uniformity };
        params = normalize(params);

        for (int k=4; k < 32; k *= 2){ // 4, 8, 16
            VectorQuantizer quant = new VectorQuantizer(params, k);
            LatLonGrid banded = quant.band(params);
            KmlWriter.write(banded, outdir, "vecquant_" + k, PngWriter.createCoolToWarmColormap());
        }
    }
```

6.116.3.5 static LatLonGrid edu.ou.asgbook.imgstat.VectorQuantizer.normalize (LatLonGrid *data*) [static]

The output grid ranges from 0 to 100.

Parameters:

input

Returns:

```
                                                           {
        LatLonGrid result = LatLonGrid.copyOf(data);
        result.setMissing(-1);
        // find range
        int min = data.getMissing();
        int max = data.getMissing();
        int nrows = data.getNumLat();
        int ncols = data.getNumLon();
        for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
            int val = data.getValue(i,j);
            if ( val != data.getMissing() ){
                if (min == data.getMissing() || val < min){
                    min = val;
                }
                if (max == data.getMissing() || val > max){
                    max = val;
                }
            }
        }
        for (int i=0; i < nrows; ++i) for (int j=0; j < ncols; ++j){
            int val = data.getValue(i,j);
            if ( val != data.getMissing() ){
                result.setValue(i, j, (int)Math.round((100.0*(val-min))/max) );
            } else {
                result.setValue(i, j, result.getMissing() );
            }
        }
        return result;
    }
```

6.116.3.6 static LatLonGrid [ ] edu.ou.asgbook.imgstat.VectorQuantizer.normalize (LatLonGrid[ ] *params*) [static]

```
                                                            {
        LatLonGrid[] norm = new LatLonGrid[params.length];
        for (int i=0; i < params.length; ++i){
            norm[i] = normalize(params[i]);
        }
        return norm;
    }
```

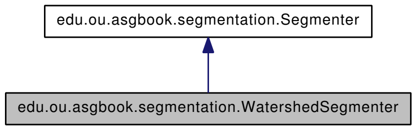
6.116.3.7 Override String edu.ou.asgbook.imgstat.VectorQuantizer.toString ()

```
                              {
        StringBuilder sb = new StringBuilder("Centroids: ");
        for (int p=0; p < centroids.length; ++p){
            sb.append(centroids[p]);
        }
        return sb.toString();
    }
```

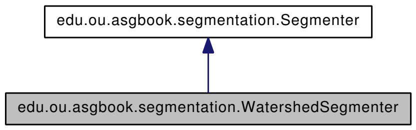
6.117 edu.ou.asgbook.segmentation.WatershedSegmenter Class Reference

Watershed approach of object identification.

Inheritance diagram for edu.ou.asgbook.segmentation.WatershedSegmenter:


edu.ou.asgbook.segmentation.Segmenter
edu.ou.asgbook.segmentation.WatershedSegmenter

Collaboration diagram for edu.ou.asgbook.segmentation.WatershedSegmenter:


edu.ou.asgbook.segmentation.Segmenter
edu.ou.asgbook.segmentation.WatershedSegmenter

Public Member Functions

- WatershedSegmenter (int thresh)
- LabelResult label (LatLonGrid data)

  Creates a labeled grid where background pixels are set to 0 and labels for objects go 1,2,3.

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.117.1 Detailed Description

Watershed approach of object identification.

Author:

valliappa.lakshmanan

6.117.2 Constructor & Destructor Documentation

6.117.2.1 edu.ou.asgbook.segmentation.WatershedSegmenter.WatershedSegmenter (int *thresh*)

```
                                            {
        super();
        this.thresh = thresh;
    }
```

6.117.3 Member Function Documentation

6.117.3.1 LabelResult edu.ou.asgbook.segmentation.WatershedSegmenter.label (LatLonGrid *data*) [virtual]

Creates a labeled grid where background pixels are set to 0 and labels for objects go 1,2,3.

.. All pixels > thresh are part of an object.

Implements edu.ou.asgbook.segmentation.Segmenter.

```
                                                {
        return vincent_segment(data);
    }
```

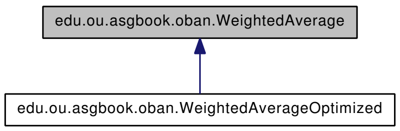
6.117.3.2 static void edu.ou.asgbook.segmentation.WatershedSegmenter.main (String[ ] *args*) throws Exception [static]

```
                                                                {
        File out = OutputDirectory.getDefault("wshed");

        // data
        LatLonGrid grid = GlobalPopulation.read(GlobalPopulation.NORTHAMERICA, new GlobalPo
        KmlWriter.write(grid, out, "orig", PngWriter.createCoolToWarmColormap());

        // int min_thresh = 0; int max_thresh = 400; int incr_thresh = 200; // log scaling
        int min_thresh = 0; int max_thresh = 20; int incr_thresh = 10; // linear scaling

        for (int thresh=min_thresh; thresh <= max_thresh; thresh += incr_thresh){
            WatershedSegmenter seg = new WatershedSegmenter(thresh);
            LatLonGrid label = seg.label(grid).label;
            KmlWriter.write(label, out, "wsheds_"+thresh, PngWriter.createRandomColormap()
        }

        grid = new ConvolutionFilter(ConvolutionFilter.gauss(9, 9)).smooth(grid);
        for (int thresh=min_thresh; thresh <= max_thresh; thresh += incr_thresh){
            WatershedSegmenter seg = new WatershedSegmenter(thresh);
            LatLonGrid label = seg.label(grid).label;
            KmlWriter.write(label, out, "urbanareas_"+thresh, PngWriter.createRandomColorma
        }
    }
```

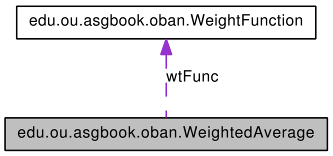
6.118 edu.ou.asgbook.oban.WeightedAverage Class Reference

Interpolation methods for point observations.

Inheritance diagram for edu.ou.asgbook.oban.WeightedAverage:


edu.ou.asgbook.oban.WeightedAverage
edu.ou.asgbook.oban.WeightedAverageOptimized

Collaboration diagram for edu.ou.asgbook.oban.WeightedAverage:


edu.ou.asgbook.oban.WeightFunction
wtFunc
edu.ou.asgbook.oban.WeightedAverage

Public Member Functions

- LatLonGrid analyze (PointObservations data)
- LatLonGrid analyze (PointObservations data, int numPasses, int physicalMin, int physicalMax)
- WeightedAverage (WeightFunction wtFunc, double latres, double lonres, int minPoints)

Static Public Member Functions

- static void main (String[ ] args) throws Exception

Protected Attributes

- final WeightFunction wtFunc
- final double latres
- final double lonres
- final int minPoints

6.118.1 Detailed Description

Interpolation methods for point observations.

This is here only for illustration; you should use the WeightedAverageOptimized

Author:

Valliappa.Lakshmanan

6.118.2 Constructor & Destructor Documentation

6.118.2.1 edu.ou.asgbook.oban.WeightedAverage.WeightedAverage (WeightFunction *wtFunc*, double *latres*, double *lonres*, int *minPoints*)

```
                                                                                       {
        this.wtFunc = wtFunc;
        this.latres = latres;
        this.lonres = lonres;
        this.minPoints = minPoints;
    }
```

6.118.3 Member Function Documentation

6.118.3.1 LatLonGrid edu.ou.asgbook.oban.WeightedAverage.analyze (PointObservations *data*, int *numPasses*, int *physicalMin*, int *physicalMax*)

```
                                                                                              {
        LatLonGrid result = analyze(data); // pass #1
        final PointObservations.ObservationPoint[] points = data.getPoints();
        for (int pass=1; pass < numPasses; ++pass){
            // find error at each point
            PointObservations.ObservationPoint[] errors = new PointObservations.ObservationPoint[points
            for (int k=0; k < points.length; ++k){
                int a = points[k].getValue();
                int b = result.getValue(points[k]);
                int error = 0;
                if ( a != data.getMissing() && b != result.getMissing() ){
                    error = a - b;
                }
                errors[k] = new PointObservations.ObservationPoint(points[k].getLat(), points[k].getLon
            }
            // create a grid of errors and add this to the original grid
            LatLonGrid errGrid = analyze(new PointObservations(errors,data.getMissing()));
            add( result, errGrid , physicalMin, physicalMax);
        }
        return result;
    }
```

6.118.3.2 LatLonGrid edu.ou.asgbook.oban.WeightedAverage.analyze (PointObservations *data*)

Reimplemented in edu.ou.asgbook.oban.WeightedAverageOptimized.

```
                                                     {
        LatLonGrid grid = ObjectiveAnalysisUtils.createBoundingGrid(data, latres, lonres);
        PointObservations.ObservationPoint[] points = data.getPoints();
        for (int i=0; i < grid.getNumLat(); ++i){
            for (int j=0; j < grid.getNumLon(); ++j){
                LatLon gridpt = grid.getLocation(i, j);
                double sum = 0;
                double sumwt = 0;
                int n = 0;
                for (int k=0; k < points.length; ++k){
                    if ( points[k].getValue() != data.getMissing() ){
                        double wt = wtFunc.computeWt( points[k].getLat() - gridpt.getLat(),
                        if ( wt > 0 ){
                            sum += wt * points[k].getValue();
                            sumwt += wt;
                            ++n;
                        }
                    }
                }
                if ( n >= minPoints ){
                    grid.setValue(i, j, (int) Math.round(sum/sumwt));
                } else {
                    grid.setValue(i, j, grid.getMissing());
                }
            }
        }
        return grid;
    }
```

6.118.3.3 static void edu.ou.asgbook.oban.WeightedAverage.main (String[ ] *args*) throws Exception [static]

Reimplemented in edu.ou.asgbook.oban.WeightedAverageOptimized.

```
                                                             {
         PointObservations data = DailyRainfall.read(DailyRainfall.TN_Oct2010);

         double meansep = ObjectiveAnalysisUtils.computeMeanDistance(data);
         System.out.println("Objectively analyzing " + data.getPoints().length + " pts with
         WeightFunction wtFunc = new CressmanWeighting(3*meansep);
         WeightedAverage analyzer = new WeightedAverage(wtFunc, 0.01, 0.01, 1);

         long startTime = System.nanoTime();
         final int numPasses = 2;
         LatLonGrid grid = analyzer.analyze(data, numPasses, 0, data.getMaxValue());
         System.out.println("Took " + (System.nanoTime() - startTime)/(1000*1000.0*1000) + "

         // write output
         File out = OutputDirectory.getDefault("oban");
         KmlWriter.write(grid, out, "Precip24H", PngWriter.createCoolToWarmColormap());
     }
```

6.118.4 Member Data Documentation

6.118.4.1 final double edu.ou.asgbook.oban.WeightedAverage.latres

`[protected]`

6.118.4.2 final double edu.ou.asgbook.oban.WeightedAverage.lonres

`[protected]`

6.118.4.3 final int edu.ou.asgbook.oban.WeightedAverage.minPoints

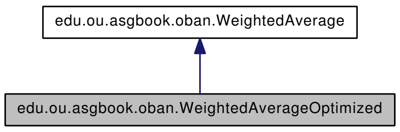
`[protected]`

6.118.4.4 final WeightFunction edu.ou.asgbook.oban.WeightedAverage.wtFunc

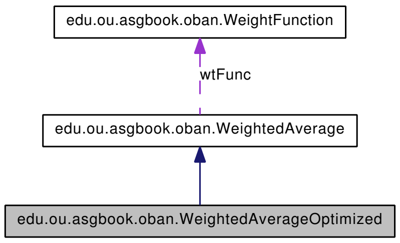
`[protected]`

6.119 edu.ou.asgbook.oban.WeightedAverage-Optimized Class Reference

Inheritance diagram for edu.ou.asgbook.oban.WeightedAverageOptimized:


edu.ou.asgbook.oban.WeightedAverage
edu.ou.asgbook.oban.WeightedAverageOptimized

Collaboration diagram for edu.ou.asgbook.oban.WeightedAverageOptimized:


edu.ou.asgbook.oban.WeightFunction
wtFunc
edu.ou.asgbook.oban.WeightedAverage
edu.ou.asgbook.oban.WeightedAverageOptimized

Public Member Functions

- Override LatLonGrid analyze (PointObservations data)
- WeightedAverageOptimized (WeightFunction wtFunc, double latres, double lon-res, int minPoints)

Static Public Member Functions

- static void main (String[ ] args) throws Exception

6.119.1 Detailed Description

Author:

Valliappa.Lakshmanan

6.119.2 Constructor & Destructor Documentation

6.119.2.1 edu.ou.asgbook.oban.WeightedAverageOptimized.WeightedAverageOptimized (WeightFunction *wtFunc*, double *latres*, double *lonres*, int *minPoints*)

```

        super(wtFunc, latres, lonres, minPoints);
        this.wtKernel = computeWeightKernel(wtFunc, latres, lonres);
    }
```

6.119.3 Member Function Documentation

6.119.3.1 Override LatLonGrid edu.ou.asgbook.oban.WeightedAverageOptimized.analyze (PointObservations *data*)

Reimplemented from edu.ou.asgbook.oban.WeightedAverage.

```
                                                             {
        LatLonGrid grid = ObjectiveAnalysisUtils.createBoundingGrid(data, latres, lonres);
        double[][] sum = new double[grid.getNumLat()][grid.getNumLon()];
        double[][] sumwt = new double[grid.getNumLat()][grid.getNumLon()];
        int[][] numpts = new int[grid.getNumLat()][grid.getNumLon()];
        PointObservations.ObservationPoint[] points = data.getPoints();

        final int half_rows = wtKernel.length / 2;
        final int half_cols = wtKernel.length / 2;
        for (int k=0; k < points.length; ++k){
            final int row = grid.getRow(points[k]);
            final int col = grid.getCol(points[k]);
            if ( points[k].getValue() != data.getMissing() ){
                for (int m=-half_rows; m <= half_rows; ++m){
                    for (int n=-half_cols; n <= half_cols; ++n){
                        final int i = row + m;
                        final int j = col + n;
                        final double wt = wtKernel[m+half_rows][n+half_rows];
                        if ( wt > 0 && grid.isValid(i, j)){
                            sum[i][j] += points[k].getValue() * wt;
                            sumwt[i][j] += wt;
                            numpts[i][j] ++;
                        }
                    }
                }
            }
        }

        for (int i=0; i < grid.getNumLat(); ++i){
            for (int j=0; j < grid.getNumLon(); ++j){
                if ( numpts[i][j] >= minPoints ){
                    grid.setValue(i, j, (int) Math.round(sum[i][j]/sumwt[i][j]));
                } else {
                    grid.setValue(i, j, grid.getMissing());
                }
            }
        }
        return grid;
    }
```

6.119.3.2 static void edu.ou.asgbook.oban.WeightedAverageOptimized.main (String[ ] *args*) throws Exception [static]

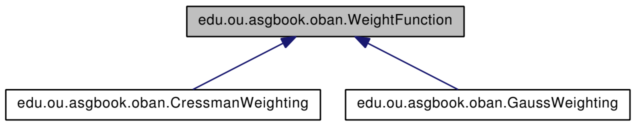
Reimplemented from edu.ou.asgbook.oban.WeightedAverage.

```
                                                       {
        File out = OutputDirectory.getDefault("obanopt");
        run(DailyRainfall.read(DailyRainfall.TN_Oct2010), out, "Precip24H", 2);
         run(MadisTemperature.read(MadisTemperature.TN_Oct2010), out, "Temperature_pass1", 
         run(MadisTemperature.read(MadisTemperature.TN_Oct2010), out, "Temperature_pass2", 
         run(MadisTemperature.read(MadisTemperature.TN_Oct2010), out, "Temperature_pass3", 
         run(MadisTemperature.read(MadisTemperature.TN_Oct2010), out, "Temperature_pass10",
     }
```

6.120 edu.ou.asgbook.oban.WeightFunction Interface Reference

Used by WeightedAverage.

Inheritance diagram for edu.ou.asgbook.oban.WeightFunction:


edu.ou.asgbook.oban.WeightFunction
edu.ou.asgbook.oban.CressmanWeighting
edu.ou.asgbook.oban.GaussWeighting

Public Member Functions

- abstract double computeWt (double latdist, double londist)

  Subclasses implement a weighting function.

Static Public Attributes

- static final double INVALID_WEIGHT = -100.0

6.120.1 Detailed Description

Used by WeightedAverage.

Author:

Valliappa.Lakshmanan

6.120.2 Member Function Documentation

6.120.2.1 abstract double edu.ou.asgbook.oban.WeightFunction.computeWt (double *latdist*, double *londist*) `[pure virtual]`

Subclasses implement a weighting function.

If -ve value is returned, then the point will be considered too far away and not used in weighting.

Implemented in edu.ou.asgbook.oban.CressmanWeighting, and edu.ou.asgbook.oban.GaussWeighting.

6.120.3 Member Data Documentation

6.120.3.1 final double edu.ou.asgbook.oban.WeightFunction.INVALID_-WEIGHT = -100.0 `[static]`

6.121 edu.ou.asgbook.dataset.WorldBankGDI Class Reference

Reads country-by-country Global development index from World Bank.

Public Types

- LowIncome
- enum DevelopmentCategory {
  LowIncome, LowerMiddleIncome, UpperMiddleIncome, HighIncomeNon-OECD,
  HighIncomeOECD, text }

Static Public Member Functions

- static CountryDI[ ] read (File file) throws IOException
  reads data from a CSV File.
- static LatLonGrid readGrid (File file) throws IOException
  reads data from a ESRI grid file.
- static DevelopmentLookup readAsMap (File f) throws Exception
- static void main (String[ ] args) throws Exception

Static Public Attributes

- static File WORLD_TABULAR = new File("data/development/WDI_GDF_-Country.csv")
- static File WORLD_GRID = new File("data/development/globaldevelopmentindex.txt.gz")

Classes

- class **CountryDI**
- class **DevelopmentLookup**

6.121.1 Detailed Description

Reads country-by-country Global development index from World Bank.

Author:

v.lakshmanan

6.121.2 Member Enumeration Documentation

6.121.2.1 enum edu::ou::asgbook::dataset::WorldBankGDI::DevelopmentCategory

Enumerator:

LowIncome

LowerMiddleIncome

UpperMiddleIncome

HighIncomeNonOECD

HighIncomeOECD

text

```
                                    {
        LowIncome, LowerMiddleIncome, UpperMiddleIncome, HighIncomeNonOECD, HighIncomeOECD,

        public static DevelopmentCategory getInstance(String text){
            if (text.equals("Low income")){
                return LowIncome;
            } else  if (text.equals("Lower middle income")){
                return LowerMiddleIncome;
            } else  if (text.equals("Upper middle income")){
                return UpperMiddleIncome;
            } else  if (text.equals("High income: nonOECD")){
                return HighIncomeNonOECD;
            } else  if (text.equals("High income: OECD")){
                    return HighIncomeOECD;
            }
            throw new IllegalArgumentException("Unknown category: {" + text + "}");
        }
    }
```

6.121.3 Member Function Documentation

6.121.3.1 static void edu.ou.asgbook.dataset.WorldBankGDI.main (String[ ] *args*) throws Exception [static]

```
                                                         {
        WorldBankGDI.CountryDI[] countries = WorldBankGDI.read(WorldBankGDI.WORLD_TABULAR)
        for (WorldBankGDI.CountryDI c : countries){
            System.out.println(c);
        }

        // some basic stats
        System.out.println("Distribution of income levels is as follows:");
        int[] count = new int[ DevelopmentCategory.values().length ];
        for (WorldBankGDI.CountryDI c : countries){
            count[ c.category.ordinal() ] ++;
        }
        for (int i=0; i < count.length; ++i){
            WorldBankGDI.DevelopmentCategory cat = WorldBankGDI.DevelopmentCategory.values()[i];
            System.out.println( cat + " " + count[i]);
        }

        // now combine with country polygons
        DevelopmentLookup lookup = WorldBankGDI.readAsMap(WorldBankGDI.WORLD_TABULAR);
        LatLonGrid countryGrid = CountryPolygons.readGrid(CountryPolygons.WORLD_GRID);
        DevelopmentCategory[] categories = lookup.getDevelopmentCategories(CountryPolygons.readKml(Cou
        for (int i=0; i < countryGrid.getNumLat(); ++i){
            for (int j=0; j < countryGrid.getNumLon(); ++j){
                int countryIndex = countryGrid.getValue(i,j);
                if ( countryIndex >= 0 ){
                    int devCategory = categories[countryIndex].ordinal();
                    countryGrid.setValue(i,j, devCategory);
                }
            }
        }
        File out = OutputDirectory.getDefault("countries");
        KmlWriter.write(countryGrid, out, "gdi", PngWriter.createCoolToWarmColormap());
        EsriGrid.write(countryGrid, out, "globaldevelopmentindex.txt.gz");
        EsriGrid.write(countryGrid, WorldBankGDI.WORLD_GRID);
    }
```

6.121.3.2 static CountryDI [ ] edu.ou.asgbook.dataset.WorldBankGDI.read (File *file*) throws IOException `[static]`

reads data from a CSV File.

The File can be gzipped or uncompressed.

```
                                                                  {
        Reader f = null;
        System.out.println("Reading " + file.getAbsolutePath());
        if (file.getAbsolutePath().endsWith(".gz")) {
            f = new InputStreamReader(new GZIPInputStream(new FileInputStream(
                    file)));
        } else {
            f = new FileReader(file);
        }
        return read(f);
    }
```

6.121.3.3 static DevelopmentLookup edu.ou.asgbook.dataset.WorldBankGDI.readAsMap (File *f*) throws Exception `[static]`

```
                                                                          {
        DevelopmentLookup result = new DevelopmentLookup();
        for (CountryDI c : read(f)){
            result.add(c);
        }
        return result;
    }
```

6.121.3.4 static LatLonGrid edu.ou.asgbook.dataset.WorldBankGDI.readGrid (File *file*) throws IOException [static]

reads data from a ESRI grid file.

The File can be gzipped or uncompressed.

```
                                                                  {
        return EsriGrid.read(file, new LinearScaling(1));
    }
```

6.121.4 Member Data Documentation

6.121.4.1 File edu.ou.asgbook.dataset.WorldBankGDI.WORLD_GRID = new File("data/development/globaldevelopmentindex.txt.gz") [static]

6.121.4.2 File edu.ou.asgbook.dataset.WorldBankGDI.WORLD_TABULAR = new File("data/development/WDI_GDF_Country.csv") [static]

Chapter 7

Automating Analysis of Spatial Grids File Documentation

7.1 AlignAndDifference.java File Reference

Namespaces

- namespace edu.ou.asgbook.motion

Classes

- class edu.ou.asgbook.motion.AlignAndDifference
  Aligns two grids and then computes their difference.

7.2 AlignmentEstimator.java File Reference

Namespaces

- namespace edu.ou.asgbook.transforms

Classes

- class edu.ou.asgbook.transforms.AlignmentEstimator

  Estimate the degree of spatial displacement between two similar grids.

7.3 Assignment4.java File Reference

Namespaces

- namespace edu.ou.asgbook.usage

Classes

- class edu.ou.asgbook.usage.Assignment4

  (1) Find optimal threshold on log(pop) Find distance of every grid point to a point < thresh Find optimal threshold of distance values Threshold image to keep only values < threshold

7.4 BoundingBox.java File Reference

Namespaces

- namespace edu.ou.asgbook.rasterization

Classes

- class edu.ou.asgbook.rasterization.BoundingBox

  A rectangular bounding box of a polygon.

7.5 CatmullRom.java File Reference

Namespaces

- namespace edu.ou.asgbook.rasterization

Classes

- class edu.ou.asgbook.rasterization.CatmullRom

  A Catmull-Rom spline, a local spline.

- class **edu.ou.asgbook.rasterization.CatmullRom.XtoY**

7.6 CityCategories.java File Reference

Namespaces

- namespace edu.ou.asgbook.datamining

Classes

- class edu.ou.asgbook.datamining.CityCategories

  Obtains city data for clustering.

7.7 CityGdiModels.java File Reference

Namespaces

- namespace edu.ou.asgbook.datamining

Classes

- class edu.ou.asgbook.datamining.CityGdiModels

  Applies different data mining models to each city.

7.8 ContiguityEnhancedKMeansSegmenter.java File Reference

Namespaces

- namespace edu.ou.asgbook.segmentation

Classes

- class edu.ou.asgbook.segmentation.ContiguityEnhancedKMeansSegmenter

  Objects consist of pixels that are grown from initial centers using K-means.

- class **edu.ou.asgbook.segmentation.ContiguityEnhancedKMeans-Segmenter.Cluster**

7.9 ConvolutionFilter.java File Reference

Namespaces

- namespace edu.ou.asgbook.filters

Classes

- class edu.ou.asgbook.filters.ConvolutionFilter

  Convolve an image by a window.

7.10 CountryPolygons.java File Reference

Namespaces

- namespace edu.ou.asgbook.dataset

Classes

- class edu.ou.asgbook.dataset.CountryPolygons

  Reads country-by-country coordinates from a KML placemarks file.

- class **edu.ou.asgbook.dataset.CountryPolygons.Country**

7.11 CressmanWeighting.java File Reference

Namespaces

- namespace edu.ou.asgbook.oban

Classes

- class edu.ou.asgbook.oban.CressmanWeighting

  An interpolation method that uses 1/r^2.

7.12 CrossCorrelation.java File Reference

Namespaces

- namespace edu.ou.asgbook.motion

Classes

- class edu.ou.asgbook.motion.CrossCorrelation

  Estimates motion using cross-correlation.

7.13 CumulativeDistributionFunction.java File Reference

Namespaces

- namespace edu.ou.asgbook.histogram

Classes

- class edu.ou.asgbook.histogram.CumulativeDistributionFunction

  Forms a CDF from a Histogram.

7.14 DailyRainfall.java File Reference

Namespaces

- namespace edu.ou.asgbook.dataset

Classes

- class edu.ou.asgbook.dataset.DailyRainfall

  Reads the ASCII precipitation data available at http://madis-data.noaa.gov/public/hydrodumpguest.html.

7.15 DataSimulator.java File Reference

Namespaces

- namespace edu.ou.asgbook.rbf

Classes

- class edu.ou.asgbook.rbf.DataSimulator

  Simulates RBF data to be fit.

7.16 DataTransform.java File Reference

Namespaces

- namespace edu.ou.asgbook.linearity

Classes

- class edu.ou.asgbook.linearity.DataTransform

  Transform pixel values, usually to meet linearity requirements.

7.17 Differencer.java File Reference

Namespaces

- namespace edu.ou.asgbook.motion

Classes

- class edu.ou.asgbook.motion.Differencer

  Just computes a pixel-by-pixel difference.

7.18 DilateErodeFilter.java File Reference

Namespaces

- namespace edu.ou.asgbook.filters

Classes

- class edu.ou.asgbook.filters.DilateErodeFilter

  Carries out paired dilation followed by erosion for filling in holes.

7.19 DilationFilter.java File Reference

Namespaces

- namespace edu.ou.asgbook.filters

Classes

- class edu.ou.asgbook.filters.DilationFilter

  Expands entities by taking a local maximum.

7.20 EdgeBased.java File Reference

Namespaces

- namespace edu.ou.asgbook.motion

Classes

- class edu.ou.asgbook.motion.EdgeBased

  Estimates motion based on the displacement of edges.

7.21 Ellipsoid.java File Reference

Namespaces

- namespace edu.ou.asgbook.projections

Classes

- class edu.ou.asgbook.projections.Ellipsoid

  An ellipsoidal approximation to the earth.

7.22 EnhancedWatershedSegmenter.java File Reference

Namespaces

- namespace edu.ou.asgbook.segmentation

Classes

- class edu.ou.asgbook.segmentation.EnhancedWatershedSegmenter

  Enhanced watershed segmentation following Lakshmanan, Hondl and Rabin.

- class **edu.ou.asgbook.segmentation.EnhancedWatershedSegmenter.Glob**

7.23 Entropy.java File Reference

Namespaces

- namespace edu.ou.asgbook.histogram

Classes

- class edu.ou.asgbook.histogram.Entropy

  Compute entropy from a histogram.

7.24 ErodeDilateFilter.java File Reference

Namespaces

- namespace edu.ou.asgbook.filters

Classes

- class edu.ou.asgbook.filters.ErodeDilateFilter

  Carries out paired erosion followed by dilation for denoising.

7.25 ErosionFilter.java File Reference

Namespaces

- namespace edu.ou.asgbook.filters

Classes

- class edu.ou.asgbook.filters.ErosionFilter
  Reduces the size of entities by taking a local mininum.

7.26 EsriGrid.java File Reference

Namespaces

- namespace edu.ou.asgbook.io

Classes

- class edu.ou.asgbook.io.EsriGrid

  Read an ESRI grid.

7.27 EuclideanDT.java File Reference

Namespaces

- namespace edu.ou.asgbook.distance

Classes

- interface edu.ou.asgbook.distance.EuclideanDT

7.28 EuclideanDTPropagation.java File Reference

Namespaces

- namespace edu.ou.asgbook.distance

Classes

- class edu.ou.asgbook.distance.EuclideanDTPropagation

  Implementation of Euclidean distance that updates the distance instead of computing it afresh each time.

7.29 EuclideanDTRecursivePropagation.java File Reference

Namespaces

- namespace edu.ou.asgbook.distance

Classes

- class edu.ou.asgbook.distance.EuclideanDTRecursivePropagation

  Note that this class is only for illustrative purposes.

7.30 EuclideanDTSaito.java File Reference

Namespaces

- namespace edu.ou.asgbook.distance

Classes

- class edu.ou.asgbook.distance.EuclideanDTSaito

  The Saito technique of computing the distance transform by calculating in the two directions separately.

7.31 FFT.java File Reference

Namespaces

- namespace edu.ou.asgbook.transforms

Classes

- class edu.ou.asgbook.transforms.FFT
  FFT based on Sedgewick and Wayne.
- class **edu.ou.asgbook.transforms.FFT.Complex**

7.32 FFT2D.java File Reference

Namespaces

- namespace edu.ou.asgbook.transforms

Classes

- class edu.ou.asgbook.transforms.FFT2D

  Two-dimensional FFT.

7.33 FFTBandpassFilter.java File Reference

Namespaces

- namespace edu.ou.asgbook.transforms

Classes

- class edu.ou.asgbook.transforms.FFTBandpassFilter

  Removes noise (high frequencies) and the gross signal (low frequencies).

7.34 FFTConvolutionFilter.java File Reference

Namespaces

- namespace edu.ou.asgbook.transforms

Classes

- class edu.ou.asgbook.transforms.FFTConvolutionFilter
  An optimization for convolution using FFTs.

7.35 FuzzyCandidateMarket.java File Reference

Namespaces

- namespace edu.ou.asgbook.datamining

Classes

- class edu.ou.asgbook.datamining.FuzzyCandidateMarket

  Uses heuristic rules to choose the next market to enter.

7.36 FuzzyLogic.java File Reference

Namespaces

- namespace edu.ou.asgbook.datamining

Classes

- class edu.ou.asgbook.datamining.FuzzyLogic

    A simple fuzzy logic engine.

- class **edu.ou.asgbook.datamining.FuzzyLogic.Fuzzy**
- interface edu.ou.asgbook.datamining.FuzzyLogic.Rule
- class **edu.ou.asgbook.datamining.FuzzyLogic.IsHigh**
- class **edu.ou.asgbook.datamining.FuzzyLogic.IsLow**
- class **edu.ou.asgbook.datamining.FuzzyLogic.IsAbout**
- class **edu.ou.asgbook.datamining.FuzzyLogic.Aggregate**

    Simply applies an equal weighting to all of the values.

7.37 GaussianComponent.java File Reference

Namespaces

- namespace edu.ou.asgbook.gmm

Classes

- class edu.ou.asgbook.gmm.GaussianComponent

  Component of a Gaussian Mixture Model.

- class **edu.ou.asgbook.gmm.GaussianComponent.Expectation**

7.38 GaussianMixtureModel.java File Reference

Namespaces

- namespace edu.ou.asgbook.gmm

Classes

- class edu.ou.asgbook.gmm.GaussianMixtureModel

  A parametric approximation of a spatial grid as a sum of Gaussians.

7.39 GaussWeighting.java File Reference

Namespaces

- namespace edu.ou.asgbook.oban

Classes

- class edu.ou.asgbook.oban.GaussWeighting

  An interpolation method that uses exp(-1/r^2).

7.40 GdiPattern.java File Reference

Namespaces

- namespace edu.ou.asgbook.datamining

Classes

- class edu.ou.asgbook.datamining.GdiPattern

  The training pattern for each city.

7.41 GlobalPopulation.java File Reference

Namespaces

- namespace edu.ou.asgbook.dataset

Classes

- class edu.ou.asgbook.dataset.GlobalPopulation

  Reads the ASCII population data available at http://sedac.ciesin.columbia.edu/gpw.

- class **edu.ou.asgbook.dataset.GlobalPopulation.LogScaling**
- class **edu.ou.asgbook.dataset.GlobalPopulation.LinearScaling**

7.42 GraylevelCooccurenceMatrix.java File Reference

Namespaces

- namespace edu.ou.asgbook.imgstat

Classes

- class edu.ou.asgbook.imgstat.GraylevelCooccurenceMatrix

  Computes texture properties from a GLCM.

7.43 HilditchSkeletonization.java File Reference

Namespaces

- namespace edu.ou.asgbook.thinning

Classes

- class edu.ou.asgbook.thinning.HilditchSkeletonization

  Hilditch method of skeletonizing a grid.

- class **edu.ou.asgbook.thinning.HilditchSkeletonization.State**

7.44 Histogram.java File Reference

Namespaces

- namespace edu.ou.asgbook.histogram

Classes

- class edu.ou.asgbook.histogram.Histogram

  A histogram is an empirical probability distribution.

7.45 HistogramBinSelection.java File Reference

Namespaces

- namespace edu.ou.asgbook.histogram

Classes

- class edu.ou.asgbook.histogram.HistogramBinSelection

  Tries out different values for the number of bins and replaces each pixel value by the center of its bin.

7.46 HornSchunk.java File Reference

Namespaces

- namespace edu.ou.asgbook.motion

Classes

- class edu.ou.asgbook.motion.HornSchunk

  Horn-Schunk optical flow method of motion estimation.

7.47 HoughTransform.java File Reference

Namespaces

- namespace edu.ou.asgbook.transforms

Classes

- class edu.ou.asgbook.transforms.HoughTransform

  Finds lines in image.

- class **edu.ou.asgbook.transforms.HoughTransform.Line**

7.48 HungarianAssigner.java File Reference

Namespaces

- namespace edu.ou.asgbook.motion

Classes

- class edu.ou.asgbook.motion.HungarianAssigner

  Optimal assignment algorithm.

- class **edu.ou.asgbook.motion.HungarianAssigner.HungarianMatch**

7.49 HybridTracker.java File Reference

Namespaces

- namespace edu.ou.asgbook.motion

Classes

- class edu.ou.asgbook.motion.HybridTracker

  Estimates motion by finding cross-correlation of objects in one frame to the pixels in the previous frame.

- class **edu.ou.asgbook.motion.HybridTracker.Centroid**

7.50 HysteresisSegmenter.java File Reference

Namespaces

- namespace edu.ou.asgbook.segmentation

Classes

- class edu.ou.asgbook.segmentation.HysteresisSegmenter

  Objects consist of pixels that are > thresh2 but have at least one pixel > thresh1.

7.51 Inverter.java File Reference

Namespaces

- namespace edu.ou.asgbook.filters

Classes

- class edu.ou.asgbook.filters.Inverter

  at every pixel, replaces its value (val) by (A - val)

7.52 KalmanFilter.java File Reference

Namespaces

- namespace edu.ou.asgbook.motion

Classes

- class edu.ou.asgbook.motion.KalmanFilter
  For the time smoothing of motion vectors.

7.53 KmlWriter.java File Reference

Namespaces

- namespace edu.ou.asgbook.io

Classes

- class edu.ou.asgbook.io.KmlWriter

  Writes data out in KML form, for display in Google Earth or similar program.

7.54 LabelResult.java File Reference

Namespaces

- namespace edu.ou.asgbook.segmentation

Classes

- class edu.ou.asgbook.segmentation.LabelResult

  Result of segmentation.

7.55 LambertConformal2SP.java File Reference

Namespaces

- namespace edu.ou.asgbook.projections

Classes

- class edu.ou.asgbook.projections.LambertConformal2SP

  Lambert Conformation 2 Standard Parallels map projection.

- class **edu.ou.asgbook.projections.LambertConformal2SP.Coord**

7.56 LatLon.java File Reference

Namespaces

- namespace edu.ou.asgbook.core

Classes

- class edu.ou.asgbook.core.LatLon

  A point on the earth's surface typically in WGS84.

7.57 LatLonGrid.java File Reference

Namespaces

- namespace edu.ou.asgbook.core

Classes

- class edu.ou.asgbook.core.LatLonGrid

  A geospatial grid of data in equilat equilon coordinates typically in WGS84 ellipsoid.

7.58 LevelSet.java File Reference

Namespaces

- namespace edu.ou.asgbook.core

Classes

- class edu.ou.asgbook.core.LevelSet

  A representation of a spatial grid as a set of levels.

7.59 Line.java File Reference

Namespaces

- namespace edu.ou.asgbook.rasterization

Classes

- class edu.ou.asgbook.rasterization.Line

  A line that connects two points on the earth's surface.

7.60 LinearityVerifier.java File Reference

Namespaces

- namespace edu.ou.asgbook.linearity

Classes

- class edu.ou.asgbook.linearity.LinearityVerifier

  Given a 2D array of points, reports error measures of assuming linearity.

- interface **edu.ou.asgbook.linearity.LinearityVerifier.DataSelector**
- class **edu.ou.asgbook.linearity.LinearityVerifier.NotMissing**
- class **edu.ou.asgbook.linearity.LinearityVerifier.InRange**

7.61 LinearScaling.java File Reference

Namespaces

- namespace edu.ou.asgbook.linearity

Classes

- class edu.ou.asgbook.linearity.LinearScaling

  Scales pixel values as Ax.

7.62 LocalMeasures.java File Reference

Namespaces

- namespace edu.ou.asgbook.imgstat

Classes

- class edu.ou.asgbook.imgstat.LocalMeasures
  Statistics computed in the neighborhood of a pixel.

7.63 LoGEdgeFilter.java File Reference

Namespaces

- namespace edu.ou.asgbook.filters

Classes

- class edu.ou.asgbook.filters.LoGEdgeFilter

  Laplacian of a Gaussian edge filter.

7.64 LogScaling.java File Reference

Namespaces

- namespace edu.ou.asgbook.linearity

Classes

- class edu.ou.asgbook.linearity.LogScaling
  Transforms pixel values as log(x).

7.65 MadisTemperature.java File Reference

Namespaces

- namespace edu.ou.asgbook.dataset

Classes

- class edu.ou.asgbook.dataset.MadisTemperature

  Reads the ASCII temperature data available at http://madis-data.noaa.gov/public/sfcdumpguest.html.

7.66 MatchedFilter.java File Reference

Namespaces

- namespace edu.ou.asgbook.filters

Classes

- class edu.ou.asgbook.filters.MatchedFilter

  Convolve an image by a window that is akin to the features we want to extract.

7.67 MaxValueFilter.java File Reference

Namespaces

- namespace edu.ou.asgbook.filters

Classes

- class edu.ou.asgbook.filters.MaxValueFilter

  Finds the highest value pixel in the image.

- class **edu.ou.asgbook.filters.MaxValueFilter.Result**

7.68 MedialAxisSkeletonization.java File Reference

Namespaces

- namespace edu.ou.asgbook.thinning

Classes

- class edu.ou.asgbook.thinning.MedialAxisSkeletonization

  The MAT method of skeletonizing a grid.

7.69 MedianFilter.java File Reference

Namespaces

- namespace edu.ou.asgbook.filters

Classes

- class edu.ou.asgbook.filters.MedianFilter
  A smoothing operation that involves replacing a pixel by the local median.

7.70 MotionEstimator.java File Reference

Namespaces

- namespace edu.ou.asgbook.motion

Classes

- interface edu.ou.asgbook.motion.MotionEstimator

7.71 MultiFilter.java File Reference

Namespaces

- namespace edu.ou.asgbook.filters

Classes

- class edu.ou.asgbook.filters.MultiFilter

  Carries out multiple operations.

7.72 MultiscaleKMeansSegmenter.java File Reference

Namespaces

- namespace edu.ou.asgbook.segmentation

Classes

- class edu.ou.asgbook.segmentation.MultiscaleKMeansSegmenter

  Quantizes image into K levels, then does multiscale segmentation Does not implement the pruning techniques discussed in the paper.

- class **edu.ou.asgbook.segmentation.MultiscaleKMeansSegmenter.Cluster**

7.73 NHighest.java File Reference

Namespaces

- namespace edu.ou.asgbook.filters

Classes

- class edu.ou.asgbook.filters.NHighest

  Finds the N highest valued-pixels in image.

7.74 NHighestLevelSetImpl.java File Reference

Namespaces

- namespace edu.ou.asgbook.filters

Classes

- class edu.ou.asgbook.filters.NHighestLevelSetImpl

  Finds the N highest valued-pixels in image using a levelset implementation.

7.75 NightimeLights.java File Reference

Namespaces

- namespace edu.ou.asgbook.dataset

Classes

- class edu.ou.asgbook.dataset.NightimeLights

  Reads night-time lights data in ESRI grid format.

7.76 ObjectiveAnalysisUtils.java File Reference

Namespaces

- namespace edu.ou.asgbook.oban

Classes

- class edu.ou.asgbook.oban.ObjectiveAnalysisUtils

  Utility functions for objective analysis.

7.77 ObjectTracker.java File Reference

Namespaces

- namespace edu.ou.asgbook.motion

Classes

- class edu.ou.asgbook.motion.ObjectTracker

  Estimates motion based on assigning objects in one frame to objects in the previous frame.

- class **edu.ou.asgbook.motion.ObjectTracker.CentroidDistance**
- class **edu.ou.asgbook.motion.ObjectTracker.GreedyAssigment**
- class **edu.ou.asgbook.motion.ObjectTracker.SimpleSegmenter**
- interface edu.ou.asgbook.motion.ObjectTracker.CostEstimator
- interface edu.ou.asgbook.motion.ObjectTracker.Assigner

7.78 OrientedEllipseFilter.java File Reference

Namespaces

- namespace edu.ou.asgbook.filters

Classes

- class edu.ou.asgbook.filters.OrientedEllipseFilter

    A non-isotropic smoothing filter.

7.79 OtsuThresholdSelector.java File Reference

Namespaces

- namespace edu.ou.asgbook.histogram

Classes

- class edu.ou.asgbook.histogram.OtsuThresholdSelector

  Uses Otsu (1979) to select optimal threshold.

7.80 OutputDirectory.java File Reference

Namespaces

- namespace edu.ou.asgbook.io

Classes

- class edu.ou.asgbook.io.OutputDirectory

  Change this to change the output directory that is used by all the main().

7.81 package-info.java File Reference

Namespaces

- namespace edu.ou.asgbook.core

7.82 package-info.java File Reference

Namespaces

- namespace edu.ou.asgbook.datamining

7.83 package-info.java File Reference

Namespaces

- namespace edu.ou.asgbook.dataset

7.84 package-info.java File Reference

Namespaces

- namespace edu.ou.asgbook.distance

7.85 package-info.java File Reference

Namespaces

- namespace edu.ou.asgbook.filters

7.86 package-info.java File Reference

Namespaces

- namespace edu.ou.asgbook.geocode

7.87 package-info.java File Reference

Namespaces

- namespace edu.ou.asgbook.gmm

7.88 package-info.java File Reference

Namespaces

- namespace edu.ou.asgbook.histogram

7.89 package-info.java File Reference

Namespaces

- namespace edu.ou.asgbook.imgstat

7.90 package-info.java File Reference

Namespaces

- namespace edu.ou.asgbook.io

7.91 package-info.java File Reference

Namespaces

- namespace edu.ou.asgbook.linearity

7.92 package-info.java File Reference

Namespaces

- namespace edu.ou.asgbook.motion

7.93 package-info.java File Reference

Namespaces

- namespace edu.ou.asgbook.oban

7.94 package-info.java File Reference

Namespaces

- namespace edu.ou.asgbook.projections

7.95 package-info.java File Reference

Namespaces

- namespace edu.ou.asgbook.rasterization

7.96 package-info.java File Reference

Namespaces

- namespace edu.ou.asgbook.rbf

7.97 package-info.java File Reference

Namespaces

- namespace edu.ou.asgbook.segmentation

7.98 package-info.java File Reference

Namespaces

- namespace edu.ou.asgbook.thinning

7.99 package-info.java File Reference

Namespaces

- namespace edu.ou.asgbook.transforms

7.100 package-info.java File Reference

Namespaces

- namespace edu.ou.asgbook.usage

7.101 Pair.java File Reference

Namespaces

- namespace edu.ou.asgbook.core

Classes

- class edu.ou.asgbook.core.Pair< X, Y >

  An utility class so that methods can return two objects.

7.102 PhaseCorrelation.java File Reference

Namespaces

- namespace edu.ou.asgbook.motion

Classes

- class edu.ou.asgbook.motion.PhaseCorrelation

  Estimate motion based on FFT.

7.103 Pixel.java File Reference

Namespaces

- namespace edu.ou.asgbook.core

Classes

- class edu.ou.asgbook.core.Pixel

  A grid point in a spatial grid consists of a location and value.

- class **edu.ou.asgbook.core.Pixel.CompareLocation**
- class **edu.ou.asgbook.core.Pixel.CompareValue**

7.104 PngWriter.java File Reference

Namespaces

- namespace edu.ou.asgbook.io

Classes

- class edu.ou.asgbook.io.PngWriter

  Writes a spatial grid out as PNG file.

7.105 PointObservations.java File Reference

Namespaces

- namespace edu.ou.asgbook.core

Classes

- class edu.ou.asgbook.core.PointObservations

  A set of observation points.

- class **edu.ou.asgbook.core.PointObservations.ObservationPoint**

  An observation point is a value at a given location.

7.106 Polygon.java File Reference

Namespaces

- namespace edu.ou.asgbook.rasterization

Classes

- class edu.ou.asgbook.rasterization.Polygon

  A polygon consisting of straight edges along the earth's surface.

7.107 PrimaryCities.java File Reference

Namespaces

- namespace edu.ou.asgbook.datamining

Classes

- class edu.ou.asgbook.datamining.PrimaryCities

  Identifies the primary cities in each country.

7.108 ProjectionPursuit.java File Reference

Namespaces

- namespace edu.ou.asgbook.rbf

Classes

- class edu.ou.asgbook.rbf.ProjectionPursuit

  Approximates a spatial grid by a RBF when nothing is known beyond the number of Gaussians desired.

- interface edu.ou.asgbook.rbf.ProjectionPursuit.NextRBF
- class **edu.ou.asgbook.rbf.ProjectionPursuit.SpatialMean**
- class **edu.ou.asgbook.rbf.ProjectionPursuit.LocalMax**

7.109 PyramidalCrossCorrelation.java File Reference

Namespaces

- namespace edu.ou.asgbook.motion

Classes

- class edu.ou.asgbook.motion.PyramidalCrossCorrelation

  Cross-correlation at muliple resolutions.

7.110 Quantizer.java File Reference

Namespaces

- namespace edu.ou.asgbook.imgstat

Classes

- class edu.ou.asgbook.imgstat.Quantizer

  Develops a quantization scheme using histogram equalization.

7.111 QuickSelect.java File Reference

Namespaces

- namespace edu.ou.asgbook.filters

Classes

- class edu.ou.asgbook.filters.QuickSelect

  From Numerical Recipes, a fast way to find the kth smallest item in a list Useful to implement rank filters.

7.112 RadialBasisFunction.java File Reference

Namespaces

- namespace edu.ou.asgbook.rbf

Classes

- class edu.ou.asgbook.rbf.RadialBasisFunction

  Finds best fit of a spatial grid to a sum of Gaussians when the centers and sigmas of the Gaussians are known.

7.113 RegionGrowing.java File Reference

Namespaces

- namespace edu.ou.asgbook.segmentation

Classes

- class edu.ou.asgbook.segmentation.RegionGrowing

  Common object-identification utility.

7.114 RegionProperty.java File Reference

Namespaces

- namespace edu.ou.asgbook.segmentation

Classes

- class edu.ou.asgbook.segmentation.RegionProperty

  Properties of a region such as geometric (centroid, area, etc) and physical (based on other grid values).

- class **edu.ou.asgbook.segmentation.RegionProperty.Ellipse**

7.115 Remapper.java File Reference

Namespaces

- namespace edu.ou.asgbook.projections

Classes

- class edu.ou.asgbook.projections.Remapper

  Utilities to remap one map projection to another.

7.116 SaturateFilter.java File Reference

Namespaces

- namespace edu.ou.asgbook.filters

Classes

- class edu.ou.asgbook.filters.SaturateFilter

    Sets all values < MIN to MIN and all values > MAX to MAX.

7.117 ScalarStatistic.java File Reference

Namespaces

- namespace edu.ou.asgbook.core

Classes

- class edu.ou.asgbook.core.ScalarStatistic

  A utility class to compute mean, variance of a streaming set of inputs.

7.118 Segmenter.java File Reference

Namespaces

- namespace edu.ou.asgbook.segmentation

Classes

- interface edu.ou.asgbook.segmentation.Segmenter

  Object identification technique.

7.119 SeparableConvolutionFilter.java File Reference

Namespaces

- namespace edu.ou.asgbook.filters

Classes

- class edu.ou.asgbook.filters.SeparableConvolutionFilter
  An optimized convolution filter.

7.120 SeviriInfraredTemperature.java File Reference

Namespaces

- namespace edu.ou.asgbook.dataset

Classes

- class edu.ou.asgbook.dataset.SeviriInfraredTemperature
  To read binary dump output from WDSS-II (http://www.wdssii.org/).

7.121 SimpleThresholder.java File Reference

Namespaces

- namespace edu.ou.asgbook.filters

Classes

- class edu.ou.asgbook.filters.SimpleThresholder

  Replace pixel values with 1 or 0 depending on whether they are above or below a single threshold.

7.122 SnakeActiveContour.java File Reference

Namespaces

- namespace edu.ou.asgbook.segmentation

Classes

- class edu.ou.asgbook.segmentation.SnakeActiveContour

    Active contour method of identifying objects.

- class **edu.ou.asgbook.segmentation.SnakeActiveContour.SnakeNode**
- class edu.ou.asgbook.segmentation.SnakeActiveContour.Snake

7.123 SobelEdgeFilter.java File Reference

Namespaces

- namespace edu.ou.asgbook.filters

Classes

- class edu.ou.asgbook.filters.SobelEdgeFilter

  Find edges in a grid.

7.124 SpatialFilter.java File Reference

Namespaces

- namespace edu.ou.asgbook.filters

Classes

- interface edu.ou.asgbook.filters.SpatialFilter

7.125 SpeckleFilter.java File Reference

Namespaces

- namespace edu.ou.asgbook.filters

Classes

- class edu.ou.asgbook.filters.SpeckleFilter

  Denoising filter that removes speckle.

7.126 Sprawl.java File Reference

Namespaces

- namespace edu.ou.asgbook.usage

Classes

- class edu.ou.asgbook.usage.Sprawl

  Solution to a classroom assignment to identify regions of urban sprawl from the population density data.

7.127 StructuralMeasures.java File Reference

Namespaces

- namespace edu.ou.asgbook.imgstat

Classes

- class edu.ou.asgbook.imgstat.StructuralMeasures
  Statistics computed in the neighborhood of a pixel.

7.128 SurfaceAlbedo.java File Reference

Namespaces

- namespace edu.ou.asgbook.dataset

Classes

- class edu.ou.asgbook.dataset.SurfaceAlbedo

  Reads lambert-conformal ascii grid.

7.129 ThresholdSegmenter.java File Reference

Namespaces

- namespace edu.ou.asgbook.segmentation

Classes

- class edu.ou.asgbook.segmentation.ThresholdSegmenter

  Simple object identification based on a single threshold.

7.130 UsaZipcode.java File Reference

Namespaces

- namespace edu.ou.asgbook.geocode

Classes

- class edu.ou.asgbook.geocode.UsaZipcode

  Find the city for each zipcode in the USA.

- class **edu.ou.asgbook.geocode.UsaZipcode.Entry**

7.131 VectorQuantizer.java File Reference

Namespaces

- namespace edu.ou.asgbook.imgstat

Classes

- class edu.ou.asgbook.imgstat.VectorQuantizer

  Develops a quantization scheme using vector quantization.

- class **edu.ou.asgbook.imgstat.VectorQuantizer.Vector**

7.132 WatershedSegmenter.java File Reference

Namespaces

- namespace edu.ou.asgbook.segmentation

Classes

- class edu.ou.asgbook.segmentation.WatershedSegmenter

  Watershed approach of object identification.

7.133 WeightedAverage.java File Reference

Namespaces

- namespace edu.ou.asgbook.oban

Classes

- class edu.ou.asgbook.oban.WeightedAverage

  Interpolation methods for point observations.

7.134 WeightedAverageOptimized.java File Reference

Namespaces

- namespace edu.ou.asgbook.oban

Classes

- class edu.ou.asgbook.oban.WeightedAverageOptimized

7.135 WeightFunction.java File Reference

Namespaces

- namespace edu.ou.asgbook.oban

Classes

- interface edu.ou.asgbook.oban.WeightFunction

  Used by WeightedAverage.

7.136 WorldBankGDI.java File Reference

Namespaces

- namespace edu.ou.asgbook.dataset

Classes

- class edu.ou.asgbook.dataset.WorldBankGDI

  Reads country-by-country Global development index from World Bank.

- class **edu.ou.asgbook.dataset.WorldBankGDI.CountryDI**
- class **edu.ou.asgbook.dataset.WorldBankGDI.DevelopmentLookup**

Index