

Reference Manual

Generated by Doxygen 1.8.15

Contents

Chapter 1

SKIZ Voronoi Diagram Tool and Matlab Bindings for VOISE Algorithm

1.1 Introduction

The SKIZ operator algorithm, described by Sequeira and Preteux in [1], is an efficient way of calculating and maintaining a Voronoi diagram in discrete 2D space. Unlike other, more popular algorithms such as Fortune's Sweep, the SKIZ algorithm is dynamic so adding and removing seeds does not require recalculation of the entire graph. Moreover, checking the bounds of a region $R(s)$ and checking whether a pixel p belongs to a region $R(s)$ is reduced to evaluation of a small number of inequalities - a number which is bounded above by twice the number of neighbouring seeds.

The discrete nature of the SKIZ algorithm makes it well suited to image segmentation. The VOISE algorithm [2] relies on a fast and dynamic method of recalculation of Voronoi diagrams upon addition and removal of seeds, so the two are a natural fit. Although a standalone version of the SKIZ algorithm is included here (mostly for testing), the main functionality is provided by the matlab bindings in `pushVD`, `grabVD`, `addSeedToVD` and `removeSeedFromVD`, which are compiled into Matlab-readable MEX binaries and are tailored specifically for the code built and maintained by P. Guio and N. Achilleos to aid faster image analysis through VOISE.

1.2 Installation

1.2.1 Step 1: Installation

Instructions: Cmake etc.

1.3 References

[1] R. E. Sequeira and F. J. Preteux. Discrete voronoi diagrams and the skiz operator: a dynamic algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1165–1170, 1997. [doi: 10.1109/34.625128]

[2] P. Guio and N. Achilleos. The voice algorithm: a versatile tool for automatic segmentation of astronomical images. *Monthly Notices of the Royal Astronomical Society*, 398(3):1254–1262, 2009. [doi: 10.1111/j.1365-2966.2009.15218.x]

Chapter 2

Test List

Global TEST_CASE ("Check whether the addSeed method correctly recalculates the lambda matrix")

AddSeedCheckLambda

Global TEST_CASE ("Check squared distance between points, some of which have negative coordinates")

SquaredDistanceNegativePoints

Global TEST_CASE ("Check squared distance between points which are neither vertically nor horizontally aligned")

SquaredDistanceNonAlignedPoints

Global TEST_CASE ("Check squared distance between points with non-integer coordinates")

SquaredDistanceNonIntegerPoints

Global TEST_CASE ("Check squared distance on horizontally aligned points")

SquaredDistanceHorizontalPoints

Global TEST_CASE ("Check squared distance on vertically aligned points")

SquaredDistanceVerticalPoints

Global TEST_CASE ("Square distance of identical points")

SquaredDistanceIdenticalPoints

Global TEST_CASE ("Check whether the removeSeed method correctly recalculates the v matrix")

RemoveSeedCheckV

Global TEST_CASE ("Check whether the removeSeed method correctly recalculates the lambda matrix")

RemoveSeedCheckLambda

Global TEST_CASE ("Upper bounds of getRegion")

GetRegionUpperBound

Global TEST_CASE ("Lower bounds of getRegion")

GetRegionLowerBound

Global TEST_CASE ("Check if inVector can handle empty vectors")

InEmptyVector

Global TEST_CASE ("Check if inVector correctly identifies lack of item in vector of reals")

NotInVectorOfReals

Global TEST_CASE ("Check if inVector correctly identifies item in vector of reals")

InVectorOfReals

Global TEST_CASE ("Check if inVector correctly identifies lack of item in vector of ints")

NotInVectorOfInts

Global **TEST_CASE** ("Check if inVector correctly identifies item in vector of ints")

 InVectorOfInts

Global **TEST_CASE** ("Check whether the addSeed method correctly recalculates the v matrix.")

 AddSeedCheckV

Chapter 3

Module Index

3.1 Global Functions

All non class method functions (links contain more details):

addSeed	??
getRegion	??
grabVD	??
nsStar	??
pointInRegion	??
pushVD	??
circumcentre	??
inVector	??
readMatrix	??
readSeeds	??
sqDist	??
updateDict	??

Chapter 4

Hierarchical Index

4.1 Class Hierarchy

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

exception	
SKIZException	??
SKIZIdenticalSeedsException	??
SKIZIndexException	??
SKIZIOException	??
SKIZLinearSeedsException	??
V_struct	??
vd	??
W_struct	??

Chapter 5

Data Structure Index

5.1 Data Structures

Here are the data structures with brief descriptions:

SKIZException	Parent class for all SKIZ exceptions	??
SKIZIdenticalSeedsException	Thrown if addSeed is given a seed to add to Voronoi diagram where one already exists	??
SKIZIndexException	Thrown when trying to access a non-existent entry in a std::vector or std::map	??
SKIZIOException	Thrown in case of failure to open a file for reading or writing	??
SKIZLinearSeedsException	Thrown by circumcentre if input coordinates form a line	??
V_struct	As defined in [1], Section 3 (Vk)	??
vd	Contains all information about voronoi diagram needed to perform SKIZ algorithm from [1] . . .	??
W_struct	As defined in [1], Section 2.2. Only for use with VOISE algorithm matlab interface. Unused but here for consistency	??

Chapter 6

File Index

6.1 File List

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

addSeed.cpp	Adds seed to Voronoi diagram	??
addSeed.h	??
getRegion.cpp	Finds the voronoi region R(s) of a seed s	??
getRegion.h	??
grabVD.cpp	Allocates memory and populates vd object with data from matlab VD struct. Only for use with Matlab mex compiler	??
grabVD.h	??
NSStar.cpp	Finds neighbouring Voronoi regions for new seeds	??
NSStar.h	??
pointInRegion.cpp	Checks whether a point is within region C(s, A) according to [1] Definition 2.5	??
pointInRegion.h	Checks whether a point is within region C(s, A) according to [1] Definition 2.5.	??
pushVD.cpp	Allocates memory and populates Matlab struct with data from vd object. Only for use with Matlab mex compiler	??
pushVD.h	??
removeSeed.cpp	Removes seed from voronoi diagram	??
removeSeed.h	??
skizException.cpp	Exception class for SKIZ Operator Tool	??
skizException.h	??
typedefs.cpp	Type definitions (all in one place)	??
vd.cpp	Voronoi diagram class	??
vd.h	??
aux-functions/circumcentre.h	Finds the circumcentre of the triangle formed by three given points (templated). Header only for templating/linking reasons	??

aux-functions/ inVector.h	Checks whether item exists within a vector. Header only for templating/linking reasons	??
aux-functions/ readMatrix.cpp	Reads matrix from ascii-formatted files generated by Matlab's 'save' function	??
aux-functions/ readMatrix.h		??
aux-functions/ readSeeds.cpp	Reads seed coordinates from ascii-formatted files generated by Matlab's 'save' function	??
aux-functions/ readSeeds.h		??
aux-functions/ sqDist.h	Finds the squared difference between two points (templated). Header only for templating/linking reasons	??
aux-functions/ updateDict.h	Routine for adding to the vector in a dictionary of vectors only if the item does not already exist (templated). Header only for templating/linking reasons	??
test/ testAddSeedCheckLambda.cpp	Unit tests for whether the addSeed method correctly recalculates the λ matrix	??
test/ testAddSeedCheckV.cpp	Unit tests for whether the addSeed method correctly recalculates the ν matrix	??
test/ testInVector.cpp	Unit tests for whether inVector correctly identifies the presence or otherwise of numeric values in vectors	??
test/ testPointInRegion.cpp	Unit tests for various normal and pathological cases for the pointInRegion function	??
test/ testRemoveSeedCheckLambda.cpp	Unit tests for whether the removeSeed method correctly recalculates the λ matrix	??
test/ testRemoveSeedCheckV.cpp	Unit tests for whether the removeSeed method correctly recalculates the ν matrix	??
test/ testSqDist.cpp	Unit tests for sqDist function	??

Chapter 7

Module Documentation

7.1 addSeed

Adds seed to voronoi diagram.

Adds seed to voronoi diagram.

Parameters

<i>VD</i>	vd object (definition in vd.h)
<i>s1</i>	First coordinate of seed to be added
<i>s2</i>	Second coordinate of seed to be added

Method used is taken from "Discrete Voronoi Diagrams and the SKIZ Operator: A Dynamic Algorithm" [1], Section 3.1

7.2 getRegion

Finds the voronoi region $R(s)$ of a seed s .

Finds the voronoi region $R(s)$ of a seed s .

Parameters

<i>VD</i>	Voronoi diagram
<i>s</i>	ID of seed for which $R(s)$ is to be found

Returns

($m \times 2$) Eigen::Array. Each row is either (-1, -1) if there are no pixels in the corresponding row in W that are also in $R(s)$, or (lb, ub) where $0 \leq lb \leq ub < n$, indicating that the pixels in the i^{th} row in the interval (lb, ub] are in $R(s)$.

$R(s)$ is as defined in [1], Definition 1.1.

7.3 grabVD

Allocates memory and populates vd object with data from matlab VD struct. Only for use with Matlab mex compiler.

Allocates memory and populates vd object with data from matlab VD struct. Only for use with Matlab mex compiler.

Parameters

in	<i>prhs</i>	Voronoi diagram in the form of a Matlab struct with the relevant fields filled in the correct manner.
----	-------------	---

Returns

Voronoi diagram (vd) object containing all relevant information.

The larger matrices (λ , ν in [1] as well as px and py) are not copied but mapped using Eigen's map class for reasons of speed.

This is part of the Matlab bindings for the VOISE algorithm [2], and is only compatible with the code written to this end by P. Guio and N. Achilleos.

7.4 nsStar

Finds neighbouring Voronoi regions for new seeds.

Finds neighbouring Voronoi regions for new seeds.

Parameters

<i>VD</i>	Voronoi diagram
-----------	-----------------

Returns

Vector of the IDs of seeds with Voronoi regions bordering the Voronoi region of the seed last added to the Voronoi diagram

Method used is taken from "Discrete Voronoi Diagrams and the SKIZ Operator: A Dynamic Algorithm" [1], Section 3.1

7.5 pointInRegion

Checks whether a point is within region $C(s, A)$ according to [1] Definition 2.5.

Checks whether a point is within region $C(s, A)$ according to [1] Definition 2.5.

Parameters

<i>vd</i>	Voronoi Diagram
<i>pt</i>	x and y coordinates of point to check
<i>s</i>	Index of seed which defines the region being checked
<i>A</i>	Vector of seeds which together form half-planes that make up $C(s, A)$

Returns

true: Point is in $C(s, A)$
false: Point is not in $C(s, A)$

7.6 pushVD

Allocates memory and populates Matlab struct with data from vd object. Only for use with Matlab mex compiler.

Allocates memory and populates Matlab struct with data from vd object. Only for use with Matlab mex compiler.

Parameters

in	<i>outputVD</i>	Voronoi diagram from which data is read
out	<i>plhs</i>	Pointer to mxArray object which is the start of the section of memory to be populated with data and which Matlab will interpret as a struct containing all of the information from outputVD. This is part of the Matlab bindings for the VOISE algorithm [2], and is only compatible with the code written to this end by P. Guio and N. Achilleos.

7.7 circumcentre

Finds the circumcentre of the triangle formed by three given points.

Functions

- `template<class T1 , class T2 , class T3 , class T4 , class T5 , class T6 >
std::array< real, 2 > circumcentre (const T1 &ax, const T2 &ay, const T3 &bx, const T4 &by, const T5 &cx,
const T6 &cy)`

7.7.1 Detailed Description

Finds the circumcentre of the triangle formed by three given points.

Parameters

<i>ax,ay</i>	x and y coordinates of first vertex
<i>bx,by</i>	x and y coordinates of second vertex
<i>cx,cy</i>	x and y coordinates of third vertex

Returns

Circumcentre of points a, b and c

The circumcentre of a triangle is the unique point in R^2 that is equidistant from its three vertices. This is the equivalent of $X(a, b, c)$ as defined in Section 2 of reference [1].

7.8 inVector

Checks whether or not item is in vector. Templated so inputs don't have to be of the same (numeric) type.

Functions

- `template<class T1 , class T2 >`
`bool inVector (const std::vector< T1 > &vec, const T2 &item)`

7.8.1 Detailed Description

Checks whether or not item is in vector. Templated so inputs don't have to be of the same (numeric) type.

Parameters

<i>vec</i>	Vector to be checked for item
<i>item</i>	Item to be looked for

Returns

true: item is in vector
false: item is not in vector

7.9 readMatrix

Reads matrix from ascii-formatted files generated by Matlab's 'save' function.

Reads matrix from ascii-formatted files generated by Matlab's 'save' function.

Parameters

<i>filename</i>	Name of text file to be read
-----------------	------------------------------

Returns

Eigen array with matrix from text file

7.10 readSeeds

Reads seed coordinates from ascii-formatted files generated by Matlab's 'save' function.

Reads seed coordinates from ascii-formatted files generated by Matlab's 'save' function.

Parameters

<i>filename</i>	Name of text file to be read
-----------------	------------------------------

Returns

Vector containing two `std::vector<double>`: the x and y coordinates of the seeds

7.11 sqDist

Finds the squared difference between two points.

Functions

- `template<class T1 , class T2 , class T3 , class T4 >`
`real sqDist` (const T1 &p1, const T2 &p2, const T3 &q1, const T4 &q2)

7.11.1 Detailed Description

Finds the squared difference between two points.

Parameters

<i>p1,p2</i>	x and y coordinates of first point
<i>q1,q2</i>	x and y coordinates of second point

Returns

Squared distance between points p and q

Using squared distance gives integer results when inputs are limited to W as defined in doi: 10.1109/34.625128, Section 2.2 which avoids floating point precision errors.

7.12 updateDict

Custom routine for adding to the vector in a dictionary of vectors only if the item does not already exist.

Custom routine for adding to the vector in a dictionary of vectors only if the item does not already exist.

Parameters

<i>d</i>	Dictionary
<i>key</i>	Key to be added
<i>value</i>	Value to be added to vector

Chapter 8

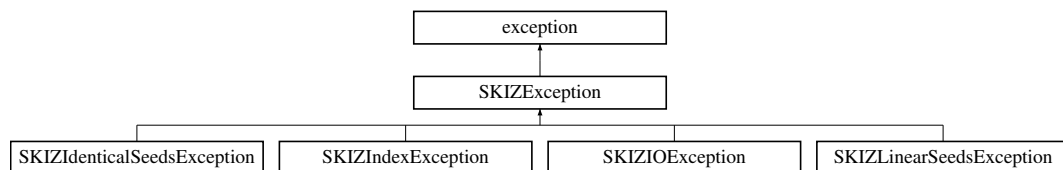
Data Structure Documentation

8.1 SKIZException Class Reference

Parent class for all SKIZ exceptions.

```
#include <skizException.h>
```

Inheritance diagram for SKIZException:



Public Member Functions

- [SKIZException](#) (const std::string s)
Constructor takes string as argument which is stored in msg.
- virtual [~SKIZException](#) () throw ()
Destructor.
- const char * [what](#) ()
Extract message stored in msg.

8.1.1 Detailed Description

Parent class for all SKIZ exceptions.

Parameters

s	Message to be given when thrown
---	---------------------------------

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

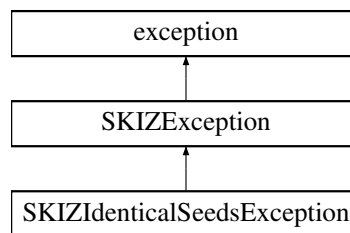
- [skizException.h](#)
- [skizException.cpp](#)

8.2 SKIZIdenticalSeedsException Class Reference

Thrown if addSeed is given a seed to add to Voronoi diagram where one already exists.

```
#include <skizException.h>
```

Inheritance diagram for SKIZIdenticalSeedsException:



Public Member Functions

- [SKIZIdenticalSeedsException](#) (const std::string s)
Constructor takes string as argument which is stored in msg.
- virtual [~SKIZIdenticalSeedsException](#) () throw ()
Destructor.
- const char * [what](#) ()
Extract message stored in msg.

8.2.1 Detailed Description

Thrown if addSeed is given a seed to add to Voronoi diagram where one already exists.

Parameters

s	Message to be given when thrown
---	---------------------------------

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

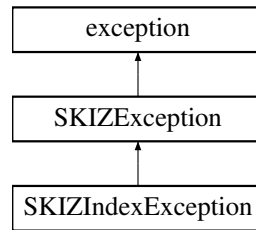
- [skizException.h](#)
- [skizException.cpp](#)

8.3 SKIZIndexException Class Reference

Thrown when trying to access a non-existent entry in a std::vector or std::map.

```
#include <skizException.h>
```

Inheritance diagram for SKIZIndexException:



Public Member Functions

- [SKIZIndexException](#) (const std::string s)
Constructor takes string as argument which is stored in msg.
- virtual [~SKIZIndexException](#) () throw ()
Destructor.
- const char * **what** ()

8.3.1 Detailed Description

Thrown when trying to access a non-existent entry in a std::vector or std::map.

Parameters

s	Message to be given when thrown
---	---------------------------------

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

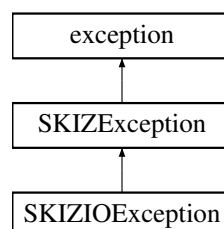
- [skizException.h](#)
- [skizException.cpp](#)

8.4 SKIZIOException Class Reference

Thrown in case of failure to open a file for reading or writing.

```
#include <skizException.h>
```

Inheritance diagram for SKIZIOException:



Public Member Functions

- [SKIZIOException](#) (const std::string s)
Constructor takes string as argument which is stored in msg.
- virtual [~SKIZIOException](#) () throw ()
Destructor.
- const char * [what](#) ()
Extract message stored in msg.

8.4.1 Detailed Description

Thrown in case of failure to open a file for reading or writing.

Parameters

s	Message to be given when thrown
---	---------------------------------

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

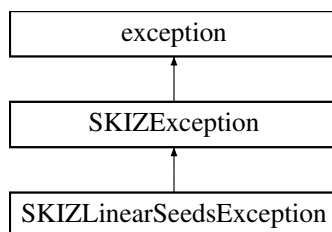
- [skizException.h](#)
- [skizException.cpp](#)

8.5 SKIZLinearSeedsException Class Reference

Thrown by circumcentre if input coordinates form a line.

```
#include <skizException.h>
```

Inheritance diagram for SKIZLinearSeedsException:



Public Member Functions

- [SKIZLinearSeedsException](#) (const std::string s)
Constructor takes string as argument which is stored in msg.
- virtual [~SKIZLinearSeedsException](#) () throw ()
Destructor.
- const char * [what](#) ()
Extract message stored in msg.

8.5.1 Detailed Description

Thrown by circumcentre if input coordinates form a line.

Parameters

s	Message to be given when thrown
---	---------------------------------

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

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

8.6 V_struct Struct Reference

as defined in [1], Section 3 (Vk).

```
#include <vd.h>
```

Data Fields

- [Mat lam](#)
- [Mat v](#)

8.6.1 Detailed Description

as defined in [1], Section 3 (Vk).

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

- [vd.h](#)

8.7 vd Class Reference

Contains all information about voronoi diagram needed to perform SKIZ algorithm from [1].

```
#include <vd.h>
```

Public Member Functions

- void `setVk` (`V_struct` val)
- void `setW` (`W_struct` val)
- void `setS` (`W_struct` val)
- void `setLam` (`Mat` newLam)
- void `setV` (`Mat` newV)
- void `setLamByIdx` (`uint32` i, `uint32` j, `real` val)
- void `setVByIdx` (`uint32` i, `uint32` j, `real` val)
- void `setSeeds` (`Mat` s)
- void `setPx` (`Mat` x)
- void `setPy` (`Mat` y)
- void `setK` (`real` val)
- void `setSx` (`std::map< real, real >` val)
- void `setSy` (`std::map< real, real >` val)
- void `setSk` (`std::map< real, real >` val)
- void `setSxByIdx` (`uint32` idx, `real` val)
- void `setSyByIdx` (`uint32` idx, `real` val)
- void `setSkByIdx` (`uint32` idx, `real` val)
- void `setNk` (`std::map< real, RealVec >` val)
- void `setNkByIdx` (`uint32` idx, `RealVec` val)
- void `incrementK` ()
- void `eraseSk` (`uint32` idx)
- `V_struct` `getVk` () const
- `W_struct` `getW` () const
- `W_struct` `getS` () const
- `Mat` `getLam` () const
- `Mat` `getV` () const
- `real` `getLamByIdx` (`uint32` i, `uint32` j) const
- `real` `getVByIdx` (`uint32` i, `uint32` j) const
- `Mat` `getSeeds` () const
- `Mat` `getPx` () const
- `Mat` `getPy` () const
- `real` `getK` () const
- `real` `getNr` () const
- `real` `getNc` () const
- `std::map< real, real >` `getSx` () const
- `std::map< real, real >` `getSy` () const
- `std::map< real, real >` `getSk` () const
- `std::map< real, RealVec >` `getNk` () const
- `real` `getSxByIdx` (`uint32` idx) const
- `real` `getSyByIdx` (`uint32` idx) const
- `real` `getSkByIdx` (`uint32` idx) const
- `RealVec` `getNkByIdx` (`uint32` idx) const
- `real` `getPxByIdx` (`uint32` i, `uint32` j)
- `real` `getPyByIdx` (`uint32` i, `uint32` j)
- `vd` (`real` rows, `real` cols)

Data Fields

- `W_struct` `W`
- `W_struct` `S`

8.7.1 Detailed Description

Contains all information about voronoi diagram needed to perform SKIZ algorithm from [1].

8.7.2 Constructor & Destructor Documentation

8.7.2.1 vd()

```
vd::vd (
    real rows,
    real cols )
```

Parameters

<i>rows</i>	Number of rows in pixel matrix
<i>cols</i>	Number of cols in pixel matrix

8.7.3 Member Function Documentation

8.7.3.1 eraseSk()

```
void vd::eraseSk (
    uint32 idx )
```

Erases an entry in the seed index dictionary Sk

Parameters

<i>idx</i>	Entry to erase
------------	----------------

8.7.3.2 getK()

```
real vd::getK ( ) const
```

Get iteration count (k) as defined in [1] Section 3.

Returns

Iteration count (k)

8.7.3.3 getLam()

```
Mat vd::getLam ( ) const
```

Get λ matrix, as defined in [1] Section 3

Returns

Eigen::Array of values of λ_{ij} at each pixel. λ_{ij} is the seed which is closest to pixel (i, j)

8.7.3.4 getLamByIdx()

```
real vd::getLamByIdx (
    uint32 i,
    uint32 j ) const
```

Get element of λ matrix, as defined in [1] Section 3

Parameters

i	Row
j	Column

Returns

Value of λ_{ij} of pixel in the i^{th} row and j^{th} column. λ_{ij} is the seed which is closest to pixel (i, j).

8.7.3.5 getNc()

```
real vd::getNc ( ) const
```

Get the number of columns of pixels

Returns

nc: the number of columns of pixels

8.7.3.6 getNk()

```
std::map< real, RealVec > vd::getNk ( ) const
```

Get neighbour relationships dictionary, as defined in [1] Section 3.

Returns

Nk, a dictionary of vectors { seed ID : vector of neighbouring seed IDs }

8.7.3.7 getNkByIdx()

```
RealVec vd::getNkByIdx (
    uint32 idx ) const
```

Get neighbour relationship for seed, as defined in [1] Section 3.

Parameters

<i>idx</i>	ID of seed for which neighbour relationships are to be retrieved
------------	--

Returns

Vector of neighbouring seeds

8.7.3.8 getNr()

```
real vd::getNr ( ) const
```

Get the number of rows of pixels

Returns

nr: the number of rows of pixels

8.7.3.9 getPx()

```
Mat vd::getPx ( ) const
```

Returns a (nr x nc) Eigen:Array of x coordinates of each pixel

Returns

(nr x nc) Eigen:Array of x coordinates of each pixel

8.7.3.10 getPxByIdx()

```
real vd::getPxByIdx (
    uint32 i,
    uint32 j )
```

Returns x coordinate of pixel

Returns

x coordinate of pixel (i, j)

8.7.3.11 getPy()

```
Mat vd::getPy ( ) const
```

Returns a (nr x nc) Eigen::Array of y coordinates of each pixel

Returns

(nr x nc) Eigen::Array of y coordinates of each pixel

8.7.3.12 getPyByIdx()

```
real vd::getPyByIdx (
    uint32 i,
    uint32 j )
```

Returns y coordinate of pixel

Returns

y coordinate of pixel (i, j)

8.7.3.13 getS()

```
W_struct vd::getS ( ) const
```

Get S struct, as defined in [1] Section 3. Only for use with VOISE algorithm matlab interface.

Returns

S

8.7.3.14 getSeeds()

```
Mat vd::getSeeds ( ) const
```

Returns a (ns x 2) Eigen::Array of all seed coordinates, where ns is the number of seeds in the Voronoi diagram.

Returns

(ns x 2) Eigen::Array of all seed coordinates. The first column is x coordinates; the second is y coordinates.

8.7.3.15 getSk()

```
std::map< real, real > vd::getSk ( ) const
```

Get Sk, the dictionary which maps seed ID to seed ID. Dictionary is used for consistency with other methods and functions and to ensure uniqueness of keys/values.

Returns

Sk dictionary { seed ID : seed ID }

8.7.3.16 getSkByIdx()

```
real vd::getSkByIdx (
    uint32 idx ) const
```

Get element of Sk, the dictionary which maps seed ID to seed ID. Dictionary is used for consistency with other methods and functions and to ensure uniqueness of keys/values. (Unused)

Parameters

<i>idx</i>	Index of seed
------------	---------------

Returns

Index of seed

8.7.3.17 getSx()

```
std::map< real, real > vd::getSx ( ) const
```

Get Sx, the dictionary which maps seed ID to x coordinate

Returns

Sx dictionary { seed ID : x coordinate }

8.7.3.18 getSxByIdx()

```
real vd::getSxByIdx (
    uint32 idx ) const
```

Get element of Sx, the dictionary which maps seed ID to x coordinate.

Parameters

<i>idx</i>	Index of seed for which coordinate is to be retrieved
------------	---

Returns

x coordinate if seed with ID = idx

8.7.3.19 getSy()

```
std::map< real, real > vd::getSy ( ) const
```

Get Sy, the dictionary which maps seed ID to y coordinate

Returns

Sy dictionary { seed ID : y coordinate }

8.7.3.20 getSyByIdx()

```
real vd::getSyByIdx (
    uint32 idx ) const
```

Get element of Sy, the dictionary which maps seed ID to y coordinate.

Parameters

<i>idx</i>	Index of seed for which coordinate is to be retrieved
------------	---

Returns

y coordinate if seed with ID = idx

8.7.3.21 getV()

```
Mat vd::getV ( ) const
```

Get ν matrix, as defined in [1] Section 3

Returns

Eigen::Array of values of ν at each pixel. $\nu_{ij} = 1$ iff there exist two or more closest seeds, else 0.

8.7.3.22 getVByIdx()

```
real vd::getVByIdx (
    uint32 i,
    uint32 j ) const
```

Get element of ν matrix, as defined in [1] Section 3

Returns

Value of ν at each pixel in the i^{th} row and j^{th} column. $\nu_{ij} = 1$ iff there exist two or more closest seeds, else 0

8.7.3.23 getVk()

```
V_struct vd::getVk ( ) const
```

Get Vk struct, as defined in [1] Section 3. Only for use with VOISE algorithm matlab interface.

Returns

Vk

8.7.3.24 getW()

```
W_struct vd::getW ( ) const
```

Get W struct, as defined in [1] Section 3. Only for use with VOISE algorithm matlab interface.

Returns

W

8.7.3.25 incrementK()

```
void vd::incrementK ( )
```

Increment iteration count (k) by 1, as defined in [1] Section 3

8.7.3.26 setK()

```
void vd::setK (
    real val )
```

Set iteration number (k) as defined in [1], Section 3

Parameters

<i>val</i>	Value to which k is set
------------	-------------------------

8.7.3.27 setLam()

```
void vd::setLam (
    Mat newLam )
```

Set λ matrix, as defined in [1] Section 3

Parameters

<i>newLam</i>	Eigen array containing λ
---------------	----------------------------------

8.7.3.28 setLamByIdx()

```
void vd::setLamByIdx (
    uint32 i,
    uint32 j,
    real val )
```

Set individual element of λ matrix, as defined in [1] Section 3

Parameters

<i>i</i>	Row index of λ
<i>j</i>	Column index of λ
<i>val</i>	Value to set λ_{ij}

8.7.3.29 setNk()

```
void vd::setNk (
    std::map< real, RealVec > val )
```

Set neighbour relationships dictionary, as defined in [1] Section 3

Parameters

<i>val</i>	Dictionary of vectors { seed ID : Vector of neighbouring seed IDs }
------------	---

8.7.3.30 setNkByIdx()

```
void vd::setNkByIdx (
    uint32 idx,
    RealVec val )
```

Set individual element in neighbour relationships dictionary, as defined in [1] Section 3

Parameters

<i>idx</i>	Key (seed ID)
<i>val</i>	Vector of neighbouring seed IDs

8.7.3.31 setPx()

```
void vd::setPx (
    Mat x )
```

Set x coordinates of each pixel.

Parameters

<i>x</i>	Eigen::Array<double> containing x coordinates of each pixel
----------	---

8.7.3.32 setPy()

```
void vd::setPy (
    Mat y )
```

Set y coordinates of each pixel

Parameters

<i>y</i>	Eigen::Array<double> containing y coordinates of each pixel
----------	---

8.7.3.33 setS()

```
void vd::setS (
    W_struct val )
```

Set S. Only for use with VOISE algorithm matlab interface.

Parameters

<i>val</i>	W_struct containing information about S
------------	---

8.7.3.34 setSeeds()

```
void vd::setSeeds (
    Mat s )
```

Set coordinates of all seeds in Voronoi diagram

Parameters

<i>s</i>	(ns x 2) Eigen::Array<double> containing coordinates of ns seeds
----------	--

8.7.3.35 setSk()

```
void vd::setSk (
    std::map< real, real > val )
```

Set Sk, the dictionary which maps seed ID to seed ID. A map is used for uniqueness and consistency with other functions and methods.

Parameters

<i>val</i>	Dictionary (Sk). All keys should be equal to their values.
------------	--

8.7.3.36 setSkByIdx()

```
void vd::setSkByIdx (
    uint32 idx,
    real val )
```

Set individual element in Sk dictionary

Parameters

<i>idx</i>	Key
<i>val</i>	k

8.7.3.37 setSx()

```
void vd::setSx (
    std::map< real, real > val )
```

Set Sx, the dictionary which maps seed ID to x coordinate

Parameters

<i>val</i>	Dictionary (Sx)
------------	-----------------

8.7.3.38 setSxByIdx()

```
void vd::setSxByIdx (
    uint32 idx,
    real val )
```

Set individual element in Sx dictionary

Parameters

<i>idx</i>	Key
<i>val</i>	x coordinate

8.7.3.39 setSy()

```
void vd::setSy (
    std::map< real, real > val )
```

Set Sy, the dictionary which maps seed ID to y coordinate

Parameters

<i>val</i>	Dictionary (Sy)
------------	-----------------

8.7.3.40 setSyByIdx()

```
void vd::setSyByIdx (
    uint32 idx,
    real val )
```

Set individual element in Sy dictionary

Parameters

<i>idx</i>	Key
<i>val</i>	y coordinate

8.7.3.41 setV()

```
void vd::setV (
    Mat newV )
```

Set ν matrix, as defined in [1] Section 3

Parameters

<i>newV</i>	Eigen array containing ν
-------------	------------------------------

8.7.3.42 setVByIdx()

```
void vd::setVByIdx (
    uint32 i,
    uint32 j,
    real val )
```

Set individual element of ν matrix, as defined in [1] Section 3

Parameters

<i>i</i>	Row index of ν
<i>j</i>	Column index of ν
<i>val</i>	Value to set ν_{ij}

8.7.3.43 setVk()

```
void vd::setVk (
    V_struct val )
```

Set Vk

Parameters

<i>val</i>	
------------	--

8.7.3.44 setW()

```
void vd::setW (
    W_struct val )
```

Set W, as defined in [1], Section 2.2. Only for use with VOISE algorithm matlab interface.

Parameters

<i>val</i>	W_struct containing information about restricted space
------------	--

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

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

8.8 W_struct Struct Reference

as defined in [1], Section 2.2. Only for use with VOISE algorithm matlab interface. Unused but here for consistency.

```
#include <vd.h>
```

Data Fields

- [real xm](#)
- [real ym](#)
- [real xM](#)
- [real yM](#)

8.8.1 Detailed Description

as defined in [1], Section 2.2. Only for use with VOISE algorithm matlab interface. Unused but here for consistency.

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

- [vd.h](#)

Chapter 9

File Documentation

9.1 addSeed.cpp File Reference

Adds seed to Voronoi diagram.

```
#include "addSeed.h"
#include "skizException.h"
#include "NSStar.h"
#include "pointInRegion.h"
#include "getRegion.h"
#include "aux-functions/inVector.h"
#include "aux-functions/sqDist.h"
#include "aux-functions/circumcentre.h"
#include "aux-functions/updateDict.h"
#include "typedefs.cpp"
#include <iostream>
```

Macros

- `#define INF std::numeric_limits<real>::infinity()`

Functions

- `bool addSeed (vd &VD, real s1, real s2)`

9.1.1 Detailed Description

Adds seed to Voronoi diagram.

9.2 addSeed.h File Reference

```
#include "vd.h"
```

Functions

- bool **addSeed** (vd &VD, real s1, real s2)

9.2.1 Detailed Description

9.3 aux-functions/circumcentre.h File Reference

Finds the circumcentre of the triangle formed by three given points (templated). Header only for templating/linking reasons.

Functions

- template<class T1 , class T2 , class T3 , class T4 , class T5 , class T6 >
std::array< real, 2 > **circumcentre** (const T1 &ax, const T2 &ay, const T3 &bx, const T4 &by, const T5 &cx, const T6 &cy)

9.3.1 Detailed Description

Finds the circumcentre of the triangle formed by three given points (templated). Header only for templating/linking reasons.

9.4 aux-functions/inVector.h File Reference

Checks whether item exists within a vector. Header only for templating/linking reasons.

```
#include <vector>
#include "../typedefs.cpp"
```

Functions

- template<class T1 , class T2 >
bool **inVector** (const std::vector< T1 > &vec, const T2 &item)

9.4.1 Detailed Description

Checks whether item exists within a vector. Header only for templating/linking reasons.

9.5 aux-functions/readMatrix.cpp File Reference

Reads matrix from ascii-formatted files generated by Matlab's 'save' function.

```
#include "readMatrix.h"
#include <fstream>
```

Functions

- [Mat](#) **readMatrix** (std::string filename, int nr, int nc)

9.5.1 Detailed Description

Reads matrix from ascii-formatted files generated by Matlab's 'save' function.

9.6 aux-functions/readSeeds.cpp File Reference

Reads seed coordinates from ascii-formatted files generated by Matlab's 'save' function.

```
#include "readSeeds.h"
#include <string>
#include <fstream>
```

Functions

- std::vector< [RealVec](#) > **readSeeds** (std::string filename)

9.6.1 Detailed Description

Reads seed coordinates from ascii-formatted files generated by Matlab's 'save' function.

9.7 aux-functions/sqDist.h File Reference

Finds the squared difference between two points (templated). Header only for templating/linking reasons.

```
#include "../typedefs.cpp"
```

Functions

- template<class T1 , class T2 , class T3 , class T4 >
[real](#) **sqDist** (const T1 &p1, const T2 &p2, const T3 &q1, const T4 &q2)

9.7.1 Detailed Description

Finds the squared difference between two points (templated). Header only for templating/linking reasons.

9.8 aux-functions/updateDict.h File Reference

Routine for adding to the vector in a dictionary of vectors only if the item does not already exist (templated). Header only for templating/linking reasons.

```
#include <vector>
#include <map>
```

Functions

- `template<class T1 , class T2 , class T3 , class T4 >`
`void updateDict (std::map< T1, std::vector< T2 >> &d, const T3 &key, const T4 &value)`

9.8.1 Detailed Description

Routine for adding to the vector in a dictionary of vectors only if the item does not already exist (templated). Header only for templating/linking reasons.

9.8.2 Function Documentation

9.8.2.1 `updateDict()`

```
template<class T1 , class T2 , class T3 , class T4 >
void updateDict (
    std::map< T1, std::vector< T2 >> & d,
    const T3 & key,
    const T4 & value )
```

If key and corresponding vector do not exist, we create both and populate vector with value

9.9 `getRegion.cpp` File Reference

Finds the voronoi region R(s) of a seed s.

```
#include <eigen3/Eigen/Dense>
#include <math.h>
#include "getRegion.h"
```

Functions

- `Mat getRegion (const vd &VD, const real &s)`

9.9.1 Detailed Description

Finds the voronoi region $R(s)$ of a seed s .

9.10 getRegion.h File Reference

```
#include <eigen3/Eigen/Dense>
#include "vd.h"
#include "typedefs.cpp"
```

Functions

- **Mat** **getRegion** (const **vd** &VD, const **real** &s)

9.10.1 Detailed Description

9.11 grabVD.cpp File Reference

Allocates memory and populates vd object with data from matlab VD struct. Only for use with Matlab mex compiler.

```
#include <eigen3/Eigen/Dense>
#include <string>
#include <map>
#include "grabVD.h"
#include "skizException.h"
#include "typedefs.cpp"
```

Functions

- **vd** **grabVD** (const mxArray *prhs[])

9.11.1 Detailed Description

Allocates memory and populates vd object with data from matlab VD struct. Only for use with Matlab mex compiler.

9.12 grabVD.h File Reference

```
#include "vd.h"
```

Functions

- [vd grabVD](#) (const mxArray *prhs[])

9.12.1 Detailed Description

9.13 NSStar.cpp File Reference

Finds neighbouring Voronoi regions for new seeds.

```
#include <map>
#include "NSStar.h"
#include "pointInRegion.h"
#include "skizException.h"
#include "aux-functions/inVector.h"
#include "aux-functions/circumcentre.h"
```

Functions

- [RealVec nsStar](#) (const [vd](#) &VD)

9.13.1 Detailed Description

Finds neighbouring Voronoi regions for new seeds.

9.14 NSStar.h File Reference

```
#include <vector>
#include "vd.h"
```

Functions

- [RealVec nsStar](#) (const [vd](#) &VD)

9.14.1 Detailed Description

9.15 pointInRegion.cpp File Reference

Checks whether a point is within region C(s, A) according to [1] Definition 2.5.

```
#include "pointInRegion.h"
#include "skizException.h"
```

Macros

- `#define INF std::numeric_limits<real>::infinity()`

Functions

- `bool pointInRegion (const vd &VD, std::array< real, 2 > pt, real s, RealVec A)`

9.15.1 Detailed Description

Checks whether a point is within region C(s, A) according to [1] Definition 2.5.

9.16 pointInRegion.h File Reference

Checks whether a point is within region C(s, A) according to [1] Definition 2.5.

```
#include <vector>
#include "vd.h"
#include "typedefs.cpp"
```

Functions

- `bool pointInRegion (const vd &VD, std::array< real, 2 > pt, real s, RealVec A)`

9.16.1 Detailed Description

Checks whether a point is within region C(s, A) according to [1] Definition 2.5.

9.17 pushVD.cpp File Reference

Allocates memory and populates Matlab struct with data from vd object. Only for use with Matlab mex compiler.

```
#include <eigen3/Eigen/Dense>
#include <string>
#include <map>
#include "pushVD.h"
#include "grabVD.h"
#include "skizException.h"
#include "typedefs.cpp"
```

Functions

- `void pushVD (vd outputVD, mxArray *plhs[])`

9.17.1 Detailed Description

Allocates memory and populates Matlab struct with data from vd object. Only for use with Matlab mex compiler.

9.18 pushVD.h File Reference

```
#include "vd.h"
```

Functions

- void **pushVD** ([vd](#) outputVD, mxArray *plhs[])

9.18.1 Detailed Description

9.19 removeSeed.cpp File Reference

Removes seed from voronoi diagram.

```
#include <set>
#include "addSeed.h"
#include "skizException.h"
#include "NSStar.h"
#include "pointInRegion.h"
#include "getRegion.h"
#include "typedefs.cpp"
#include "removeSeed.h"
#include "aux-functions/inVector.h"
#include "aux-functions/sqDist.h"
#include "aux-functions/circumcentre.h"
#include "aux-functions/updateDict.h"
```

Macros

- #define **INF** std::numeric_limits<[real](#)>::infinity()

Functions

- bool **removeSeed** ([vd](#) &VD, [real](#) Sk)
Removes seed from voronoi diagram.

9.19.1 Detailed Description

Removes seed from voronoi diagram.

9.19.2 Function Documentation

9.19.2.1 removeSeed()

```
bool removeSeed (
    vd & VD,
    real Sk )
```

Removes seed from voronoi diagram.

Parameters

<i>VD</i>	Voronoi Diagram
<i>Sk</i>	ID of seed to be removed Method used is taken from "Discrete Voronoi Diagrams and the SKIZ Operator: A Dynamic Algorithm" [1], Section 3.2.

9.20 removeSeed.h File Reference

```
#include "vd.h"
```

Functions

- bool `removeSeed` (`vd` &VD, `real` Sk)
Removes seed from voronoi diagram.

9.20.1 Detailed Description

9.20.2 Function Documentation

9.20.2.1 removeSeed()

```
bool removeSeed (
    vd & VD,
    real Sk )
```

Removes seed from voronoi diagram.

Parameters

<i>VD</i>	Voronoi Diagram
<i>Sk</i>	ID of seed to be removed Method used is taken from "Discrete Voronoi Diagrams and the SKIZ Operator: A Dynamic Algorithm" [1], Section 3.2.

9.21 skizException.cpp File Reference

Exception class for SKIZ Operator Tool.

```
#include "skizException.h"
```

9.21.1 Detailed Description

Exception class for SKIZ Operator Tool.

9.22 skizException.h File Reference

```
#include <exception>
#include <string>
```

Data Structures

- class [SKIZException](#)
Parent class for all SKIZ exceptions.
- class [SKIZLinearSeedsException](#)
Thrown by circumcentre if input coordinates form a line.
- class [SKIZIndexException](#)
Thrown when trying to access a non-existent entry in a std::vector or std::map.
- class [SKIZIdenticalSeedsException](#)
Thrown if addSeed is given a seed to add to Voronoi diagram where one already exists.
- class [SKIZIOException](#)
Thrown in case of failure to open a file for reading or writing.

9.22.1 Detailed Description

9.23 test/testAddSeedCheckLambda.cpp File Reference

Unit tests for whether the addSeed method correctly recalculates the λ matrix.

```
#include <string>
#include "Catch2/catch.hpp"
#include "../addSeed.h"
#include "../removeSeed.h"
#include "../getRegion.h"
#include "../skizException.h"
#include "../typedefs.cpp"
#include "../vd.h"
#include "test-help-fns/loadVD.h"
#include "test-help-fns/loadStruct.h"
#include "test-help-fns/bruteForceCheckLambda.h"
```

Functions

- [TEST_CASE](#) ("Check whether the addSeed method correctly recalculates the lambda matrix")
Add seeds to VD and check (in a greedy fashion) whether the closest seed to each pixel is the one held in its λ matrix entry.

9.23.1 Detailed Description

Unit tests for whether the addSeed method correctly recalculates the λ matrix.

9.23.2 Function Documentation

9.23.2.1 TEST_CASE()

```
TEST_CASE (
    "Check whether the addSeed method correctly recalculates the lambda matrix" )
```

Add seeds to VD and check (in a greedy fashion) whether the closest seed to each pixel is the one held in its λ matrix entry.

Test AddSeedCheckLambda

9.24 test/testAddSeedCheckV.cpp File Reference

Unit tests for whether the addSeed method correctly recalculates the ν matrix.

```
#include <string>
#include "Catch2/catch.hpp"
#include "../addSeed.h"
#include "../removeSeed.h"
#include "../getRegion.h"
#include "../skizException.h"
#include "../typedefs.cpp"
#include "../vd.h"
#include "test-help-fns/loadVD.h"
#include "test-help-fns/loadStruct.h"
#include "test-help-fns/bruteForceCheckV.h"
```

Functions

- [TEST_CASE](#) ("Check whether the addSeed method correctly recalculates the ν matrix.")
Adds seeds to VD and checks (in a greedy fashion) each pixel for whether there exists one or more closest seeds, and whether this corresponds to the relevant ν matrix entry.

9.24.1 Detailed Description

Unit tests for whether the addSeed method correctly recalculates the ν matrix.

9.24.2 Function Documentation

9.24.2.1 TEST_CASE()

```
TEST_CASE (
    "Check whether the addSeed method correctly recalculates the  $\nu$  matrix." )
```

Adds seeds to VD and checks (in a greedy fashion) each pixel for whether there exists one or more closest seeds, and whether this corresponds to the relevant ν matrix entry.

Test AddSeedCheckV

9.25 test/testInVector.cpp File Reference

Unit tests for whether inVector correctly identifies the presence or otherwise of numeric values in vectors.

```
#include "../aux-functions/inVector.h"
#include "../typedefs.cpp"
#include "Catch2/catch.hpp"
```

Functions

- **TEST_CASE** ("Check if inVector correctly identifies item in vector of ints")
Check if inVector correctly identifies item in vector of ints.
- **TEST_CASE** ("Check if inVector correctly identifies lack of item in vector of ints")
Check if inVector correctly identifies item in vector of ints.
- **TEST_CASE** ("Check if inVector correctly identifies item in vector of reals")
Check if inVector correctly identifies item in vector of reals.
- **TEST_CASE** ("Check if inVector correctly identifies lack of item in vector of reals")
Check if inVector correctly identifies lack of item in vector of reals.
- **TEST_CASE** ("Check if inVector can handle empty vectors")
Check if inVector can handle empty vectors.

9.25.1 Detailed Description

Unit tests for whether inVector correctly identifies the presence or otherwise of numeric values in vectors.

9.25.2 Function Documentation

9.25.2.1 TEST_CASE() [1/5]

```
TEST_CASE (
    "Check if inVector correctly identifies item in vector of ints" )
```

Check if inVector correctly identifies item in vector of ints.

Test InVectorOfInts

9.25.2.2 TEST_CASE() [2/5]

```
TEST_CASE (
    "Check if inVector correctly identifies lack of item in vector of ints" )
```

Check if inVector correctly identifies item in vector of ints.

Test NotInVectorOfInts

9.25.2.3 TEST_CASE() [3/5]

```
TEST_CASE (
    "Check if inVector correctly identifies item in vector of reals" )
```

Check if inVector correctly identifies item in vector of reals.

Test InVectorOfReals

9.25.2.4 TEST_CASE() [4/5]

```
TEST_CASE (
    "Check if inVector correctly identifies lack of item in vector of reals" )
```

Check if inVector correctly identifies lack of item in vector of reals.

Test NotInVectorOfReals

9.25.2.5 TEST_CASE() [5/5]

```
TEST_CASE (
    "Check if inVector can handle empty vectors" )
```

Check if inVector can handle empty vectors.

Test InEmptyVector

9.26 test/testPointInRegion.cpp File Reference

Unit tests for various normal and pathological cases for the pointInRegion function.

```
#include <eigen3/Eigen/Dense>
#include <string>
#include <iostream>
#include "../getRegion.h"
#include "../typedefs.cpp"
#include "../vd.h"
#include "../pointInRegion.h"
#include "test-help-fns/loadVD.h"
#include "test-help-fns/loadStruct.h"
#include "Catch2/catch.hpp"
```

Functions

- **TEST_CASE** ("Lower bounds of getRegion")
Checks whether the pixels on and around the lower bound calculated by getRegion are in said region.
- **TEST_CASE** ("Upper bounds of getRegion")
Checks whether the pixels on and around the upper bound calculated by getRegion are in said region.

Variables

- std::string **path** = "../cpp/test/resources/"
- loadStruct **loadResults** = loadVD(path)
- **RealVec Sx** = loadResults.Sx
- **RealVec Sy** = loadResults.Sy
- **vd VD** = loadResults.VD
- std::array< **real**, 2 > **pt**

9.26.1 Detailed Description

Unit tests for various normal and pathological cases for the pointInRegion function.

9.26.2 Function Documentation

9.26.2.1 TEST_CASE() [1/2]

```
TEST_CASE (
    "Lower bounds of getRegion" )
```

Checks whether the pixels on and around the lower bound calculated by getRegion are in said region.

Test GetRegionLowerBound

9.26.2.2 TEST_CASE() [2/2]

```
TEST_CASE (
    "Upper bounds of getRegion" )
```

Checks whether the pixels on and around the upper bound calculated by getRegion are in said region.

Test GetRegionUpperBound

9.27 test/testRemoveSeedCheckLambda.cpp File Reference

Unit tests for whether the removeSeed method correctly recalculates the λ matrix.

```
#include <string>
#include "Catch2/catch.hpp"
#include "../addSeed.h"
#include "../removeSeed.h"
#include "../getRegion.h"
#include "../skizException.h"
#include "../typedefs.cpp"
#include "../vd.h"
#include "test-help-fns/loadVD.h"
#include "test-help-fns/loadStruct.h"
#include "test-help-fns/bruteForceCheckLambda.h"
#include <iostream>
```

Functions

- **TEST_CASE** ("Check whether the [removeSeed](#) method correctly recalculates the lambda matrix")
Remove seeds to VD and check (in a greedy fashion) whether the closest seed to each pixel is the one held in its λ matrix entry.

9.27.1 Detailed Description

Unit tests for whether the removeSeed method correctly recalculates the λ matrix.

9.27.2 Function Documentation

9.27.2.1 TEST_CASE()

```
TEST_CASE (
    "Check whether the removeSeed method correctly recalculates the lambda matrix" )
```

Remove seeds to VD and check (in a greedy fashion) whether the closest seed to each pixel is the one held in its λ matrix entry.

Test RemoveSeedCheckLambda

9.28 test/testRemoveSeedCheckV.cpp File Reference

Unit tests for whether the `removeSeed` method correctly recalculates the ν matrix.

```
#include <string>
#include "Catch2/catch.hpp"
#include "../addSeed.h"
#include "../removeSeed.h"
#include "../getRegion.h"
#include "../skizException.h"
#include "../typedefs.cpp"
#include "../vd.h"
#include "test-help-fns/loadVD.h"
#include "test-help-fns/loadStruct.h"
#include "test-help-fns/bruteForceCheckV.h"
```

Functions

- **TEST_CASE** ("Check whether the `removeSeed` method correctly recalculates the ν matrix")
Remove seeds to VD and check (in a greedy fashion) whether the closest seed to each pixel is the one held in its ν matrix entry.

9.28.1 Detailed Description

Unit tests for whether the `removeSeed` method correctly recalculates the ν matrix.

9.28.2 Function Documentation

9.28.2.1 TEST_CASE()

```
TEST_CASE (
    "Check whether the removeSeed method correctly recalculates the v matrix" )
```

Remove seeds to VD and check (in a greedy fashion) whether the closest seed to each pixel is the one held in its v matrix entry.

Test RemoveSeedCheckV

9.29 test/testSqDist.cpp File Reference

Unit tests for sqDist function.

```
#include "../aux-functions/sqDist.h"
#include "Catch2/catch.hpp"
```

Functions

- **TEST_CASE** ("Square distance of identical points")
Test whether squared distance between identical points returns zero.
- **TEST_CASE** ("Check squared distance on vertically aligned points")
Checks squared distance on vertically aligned points.
- **TEST_CASE** ("Check squared distance on horizontally aligned points")
Checks squared distance on horizontally aligned points.
- **TEST_CASE** ("Check squared distance between points with non-integer coordinates")
Checks squared distance between points with non-integer coordinates.
- **TEST_CASE** ("Check squared distance between points which are neither vertically nor horizontally aligned")
Checks squared distance between points which are neither vertically nor horizontally aligned.
- **TEST_CASE** ("Check squared distance between points, some of which have negative coordinates")
Checks squared distance between points, some of which have negative coordinates.

9.29.1 Detailed Description

Unit tests for sqDist function.

9.29.2 Function Documentation

9.29.2.1 TEST_CASE() [1/6]

```
TEST_CASE (
    "Square distance of identical points" )
```

Test whether squared distance between identical points returns zero.

Test SquaredDistanceIdenticalPoints

9.29.2.2 TEST_CASE() [2/6]

```
TEST_CASE (
    "Check squared distance on vertically aligned points" )
```

Checks squared distance on vertically aligned points.

Test SquaredDistanceVerticalPoints

9.29.2.3 TEST_CASE() [3/6]

```
TEST_CASE (
    "Check squared distance on horizontally aligned points" )
```

Checks squared distance on horizontally aligned points.

Test SquaredDistanceHorizontalPoints

9.29.2.4 TEST_CASE() [4/6]

```
TEST_CASE (
    "Check squared distance between points with non-integer coordinates" )
```

Checks squared distance between points with non-integer coordinates.

Test SquaredDistanceNonIntegerPoints

9.29.2.5 TEST_CASE() [5/6]

```
TEST_CASE (
    "Check squared distance between points which are neither vertically nor horizontally
    aligned" )
```

Checks squared distance between points which are neither vertically nor horizontally aligned.

Test SquaredDistanceNonAlignedPoints

9.29.2.6 TEST_CASE() [6/6]

```
TEST_CASE (
    "Check squared distance between points,
    some of which have negative coordinates" )
```

Checks squared distance between points, some of which have negative coordinates.

Test SquaredDistanceNegativePoints

9.30 typedefs.cpp File Reference

Type definitions (all in one place)

```
#include <vector>
#include <eigen3/Eigen/Dense>
```

Typedefs

- typedef double [real](#)
All matlab inputs are doubles.
- typedef unsigned int [uint32](#)
Used for counters.
- typedef std::vector< [real](#) > [RealVec](#)
std::vector of reals
- typedef Eigen::Array< [real](#), Eigen::Dynamic, Eigen::Dynamic > [Mat](#)
Dynamically (determined at runtime) sized Eigen::Array of reals.

9.30.1 Detailed Description

Type definitions (all in one place)

9.31 vd.cpp File Reference

Voronoi diagram class.

```
#include "vd.h"
```

9.31.1 Detailed Description

Voronoi diagram class.

9.32 vd.h File Reference

```
#include <vector>
#include <map>
#include <algorithm>
#include <eigen3/Eigen/Dense>
#include "typedefs.cpp"
```

Data Structures

- struct [W_struct](#)
as defined in [1], Section 2.2. Only for use with VOISE algorithm matlab interface. Unused but here for consistency.
- struct [V_struct](#)
as defined in [1], Section 3 (Vk).
- class [vd](#)
Contains all information about voronoi diagram needed to perform SKIZ algorithm from [1].

9.32.1 Detailed Description