Tennis Assistant - Computer Vision

Generated by Doxygen 1.8.9.1

Fri Mar 27 2015 22:20:43

Contents

Chapter 1

Namespace Index

1.1 Packages

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

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 basket Namespace Reference

Functions

- def is_basket_middle (img)

  Single entry function returning True/False if basket is in the middle of the screen.
- def run_middle ()

  Runs continuously and prints if the best detected blob is in the middle.
- def run

  Runs continuously outlines best matched blob if it is in the middle.

Variables

- particle_filter = None
- int image_half_size = -1
- int save_count = 1
- tuple base_filename = datetime.now()

5.1.1 Function Documentation

5.1.1.1 def basket.is_basket_middle (*img*)

Single entry function returning True/False if basket is in the middle of the screen.

Parameters

img	SimpleCV.Image The image to test

5.1.1.2 def basket.run (*bestBlobCallback* = `False`)

Runs continuously outlines best matched blob if it is in the middle.

Parameters

bestBlob←↩ Callback	function Callback called passing the best blob found

5.1.1.3 def basket.run_middle ( )

Runs continuously and prints if the best detected blob is in the middle.

5.1.2 Variable Documentation

5.1.2.1 tuple basket.base_filename = datetime.now()

5.1.2.2 int basket.image_half_size = -1

5.1.2.3 basket.particle_filter = None

5.1.2.4 int basket.save_count = 1

5.2 basket_runner Namespace Reference

5.3 basket_test Namespace Reference

Functions

- def unitTest (actual, expected, name)
- def basketPresent ()
- def basketMissing ()

5.3.1 Function Documentation

5.3.1.1 def basket_test.basketMissing ( )

5.3.1.2 def basket_test.basketPresent ( )

5.3.1.3 def basket_test.unitTest (*actual, expected, name*)

5.4 experiment Namespace Reference

Functions

- def experiment
- def hard_threshold (img)
- def binary_mask (img)
- def dilation_and_blur (img)
- def blobs_by_mask (img)

5.4.1 Detailed Description

```
A utility file for testing out computer vision techniques on preset images.
The purpose of this is to avoid using the webcam, and test on consistent test
cases.
```

5.4.2 Function Documentation

5.4.2.1 def experiment.binary_mask (*img*)

5.4.2.2 def experiment.blobs_by_mask (*img*)

5.4.2.3 def experiment.dilation_and_blur (*img*)

5.4.2.4 def experiment.experiment (*image_function* = `None`, *blob_function* = `None`, *directory* = `"./"`)

5.4.2.5 def experiment.hard_threshold (*img*)

5.5 image_support Namespace Reference

Functions

- def external_init_particle_filter (img)

  Initializes particle filter.
- def image_hue_filter

  Converts given image to HSV based on the given color.
- def get_hue_blobs (img)

  Gets basket blobs after hue distance filtering.
- def get_best_blob (blobs, particle_filter)

  Returns the best blob out of the provided set and particle filter.
- def is_blob_in_middle_helper (img, blob)

  Determines whether the given blob is in ceter of image.

5.5.1 Function Documentation

5.5.1.1 def image_support.external_init_particle_filter (*img*)

Initializes particle filter.

Parameters

img	SimpleCV.Image captured image

Returns

A ParticleFilter object

5.5.1.2 def image_support.get_best_blob (*blobs*, *particle_filter*)

Returns the best blob out of the provided set and particle filter.

Parameters

blobs	list of potential HSV blobs
particle_filter	initialized ParticleFilter object

Returns

The largest blob found or None.

5.5.1.3 def image_support.get_hue_blobs (*img*)

Gets basket blobs after hue distance filtering.

Parameters

img	SimpleCV.Image captured image.

Returns

Set of 'black' potential blobs.

5.5.1.4 def image_support.image_hue_filter (*img*, *ball* = `True`)

Converts given image to HSV based on the given color.

Parameters

img	SimpleCV.Image captured image
color	tuple of RGB values of singe 'H' value of HSV

Returns

HSV converted image

5.5.1.5 def image_support.is_blob_in_middle_helper (*img*, *blob*)

Determines whether the given blob is in ceter of image.

Parameters

img	SimpleCV.Image caputed image
blob	SimpleCV.Blob Blob object

Returns

True if blob in middle of image, false otherwise.

5.6 particlefilter Namespace Reference

Classes

- class ParticleFilter

5.7 prquadtree Namespace Reference

Classes

- class Box

  Class defining a square on the coordinate system via a center point and half of square width.
- class Particle

  Represents particle point.
- class Point

  Represents an (x,y) coordinate point on a grid.
- class PRQuadTree

  Class representing a Point Range Quadtree.

5.7.1 Detailed Description

```
Implementation of a Point Range Quadtree.

Author: Pawel Szczurko
```

5.8 prquadtree_test Namespace Reference

Classes

- class TestBox
- class TestParticle
- class TestPoint
- class TestPrQuadTree

5.9 prquadtree_test_example Namespace Reference

Variables

- tuple b = Box(Point(5,5), 50)
- tuple b2 = Box(Point(50,50), 50)
- tuple qt = PRQuadTree(b2)
- tuple pt = Point(2,2)
- tuple nearby = qt.query_k_nearest(pt, 20)
- int c = 1

5.9.1 Detailed Description

```
File testing the capabilities of the PRQuadTree.

Author: Pawel Szczurko
```

5.9.2 Variable Documentation

5.9.2.1 tuple prquadtree_test_example.b = Box(Point(5,5), 50)

5.9.2.2 tuple prquadtree_test_example.b2 = Box(Point(50,50), 50)

5.9.2.3 int prquadtree_test_example.c = 1

5.9.2.4 tuple prquadtree_test_example.nearby = qt.query_k_nearest(pt, 20)

5.9.2.5 tuple prquadtree_test_example.pt = Point(2,2)

5.9.2.6 tuple prquadtree_test_example.qt = PRQuadTree(b2)

5.10 tennis_ball Namespace Reference

Functions

- def is_ball_middle (img)

  Entry point for module which determines whether tennis ball is in the middle of the image.

- def run ()

  Continuously captures image from computer camera and feeds it to the is_ball_middle method to detect whether tennis ball is in the middle of the screen.

Variables

- particle_filter = None

5.10.1 Detailed Description

```
Simple detection of ball using SimpleCV (much easier than OpenCV). The run method
identifies a tennis ball in the camera stream image. 'is_ball_middle' function
can be used to determine whether a ball is horizontally centered based on a specified
threshold.

-Pawel Szczurko
```

5.10.2 Function Documentation

5.10.2.1 def tennis_ball.is_ball_middle (*img*)

Entry point for module which determines whether tennis ball is in the middle of the image.

Parameters

img	SimpleCV.Image

Returns

boolean. True if tennis ball is in middle, false otherwise.

5.10.2.2 def tennis_ball.run ( )

Continuously captures image from computer camera and feeds it to the is_ball_middle method to detect whether tennis ball is in the middle of the screen.

5.10.3 Variable Documentation

5.10.3.1 tennis_ball.particle_filter = None

5.11 tennis_ball_runner Namespace Reference

5.12 tennis_ball_test Namespace Reference

Functions

- def unitTest (actual, expected, name)
- def ballPresent ()
- def ballMissing ()

5.12.1 Function Documentation

5.12.1.1 def tennis_ball_test.ballMissing ( )

5.12.1.2 def tennis_ball_test.ballPresent ( )

5.12.1.3 def tennis_ball_test.unitTest (*actual*, *expected*, *name*)

Chapter 6

Class Documentation

6.1 prquadtree.Box Class Reference

Class defining a square on the coordinate system via a center point and half of square width.

Public Member Functions

- def __init__ (self, center, half_size)

  Construct a Box object.
- def contains_point (self, point)

  Verifies that the given point is within this square.
- def intersect (self, other_box)

  Checks if the provided box/square intersects with this square.

Public Attributes

- center
- half_size

6.1.1 Detailed Description

Class defining a square on the coordinate system via a center point and half of square width.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 def prquadtree.Box.__init__ (*self,* *center,* *half_size*)

Construct a Box object.

Parameters

center	Point type specifying the center of the square
half_size	int half the length of the square

6.1.3 Member Function Documentation

6.1.3.1 def prquadtree.Box.contains_point (*self, point*)

Verifies that the given point is within this square.

Parameters

point	Point type to check if it's in the square

Returns

boolean indicating whether the point is within the square

6.1.3.2 def prquadtree.Box.intersect (*self, other_box*)

Checks if the provided box/square intersects with this square.

Parameters

other_box	Box object

Returns

Boolean indicating if the two intersect anywhere

6.1.4 Member Data Documentation

6.1.4.1 prquadtree.Box.center

6.1.4.2 prquadtree.Box.half_size

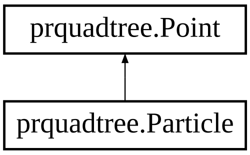
The documentation for this class was generated from the following file:

- prquadtree.py

6.2 prquadtree.Particle Class Reference

Represents particle point.

Inheritance diagram for prquadtree.Particle:



Public Member Functions

- def __init__ (self, x, y)

  Constructs a Particle.

Public Attributes

- x
- y
- score

6.2.1 Detailed Description

Represents particle point.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 def prquadtree.Particle.__init__ (*self*, *x*, *y*)

Constructs a Particle.

Parameters

x	float/int x-position
y	float/int y-position

6.2.3 Member Data Documentation

6.2.3.1 prquadtree.Particle.score

6.2.3.2 prquadtree.Particle.x

6.2.3.3 prquadtree.Particle.y

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

- prquadtree.py

6.3 particlefilter.ParticleFilter Class Reference

Public Member Functions

- def __init__ (self, box)

  Constructor.
- def iterate (self, blobs)

  For each blob, it updates the points in the tree increasing the score of those which are within the bounding square of the blob.
- def score (self, blob)

  Returns the sum of the scores of the points found within this blob by querying the quadtree.
- def clear_scores (self)

  Resets all scores of blobs This should be used when changing the webcam view.

Public Attributes

- pr_tree
- image_box
- iterations
- iterations_before_clearing

6.3.1 Constructor & Destructor Documentation

6.3.1.1 def particlefilter.ParticleFilter.__init__ (*self*, *box*)

Constructor.

Parameters

box	Box the box representing the web cam view

6.3.2 Member Function Documentation

6.3.2.1 def particlefilter.ParticleFilter.clear_scores (*self*)

Resets all scores of blobs This should be used when changing the webcam view.

6.3.2.2 def particlefilter.ParticleFilter.iterate (*self, blobs*)

For each blob, it updates the points in the tree increasing the score of those which are within the bounding square of the blob.

Parameters

blobs	array An array of blob objects which were matched

6.3.2.3 def particlefilter.ParticleFilter.score (*self, blob*)

Returns the sum of the scores of the points found within this blob by querying the quadtree.

Parameters

blob	Blob A single blob

Returns

int The sum of the scores of the points contained in the passed blob

6.3.3 Member Data Documentation

6.3.3.1 particlefilter.ParticleFilter.image_box

6.3.3.2 particlefilter.ParticleFilter.iterations

6.3.3.3 particlefilter.ParticleFilter.iterations_before_clearing

6.3.3.4 particlefilter.ParticleFilter.pr_tree

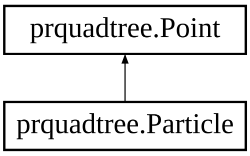
The documentation for this class was generated from the following file:

- particlefilter.py

6.4 prquadtree.Point Class Reference

Represents an (x,y) coordinate point on a grid.

Inheritance diagram for prquadtree.Point:



Public Member Functions

- def __init__ (self, x, y)

  Constructs a coordinate Point.
- def __str__ (self)

  Overwritting the default to string method of the Point class.
- def __repr__ (self)

  Needed for printing via 'print'.

Public Attributes

- x
- y

6.4.1 Detailed Description

Represents an (x,y) coordinate point on a grid.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 def prquadtree.Point.__init__ (*self,* *x,* *y*)

Constructs a coordinate Point.

Parameters

x	float/int x-position
y	float/int y-position

6.4.3 Member Function Documentation

6.4.3.1 def prquadtree.Point.__repr__ (*self*)

Needed for printing via 'print'.

6.4.3.2 def prquadtree.Point.__str__ (*self*)

Overwritting the default to string method of the Point class.

Returns

String representation of Point

6.4.4 Member Data Documentation

6.4.4.1 prquadtree.Point.x

6.4.4.2 prquadtree.Point.y

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

- prquadtree.py

6.5 prquadtree.PRQuadTree Class Reference

Class representing a Point Range Quadtree.

Public Member Functions

- def __init__ (self, box)

  Constructs a PR Quadtree given an initial square.
- def insert (self, point)

  Inserts a point into the PRQuadtree.
- def query_range (self, rng)

  Returns the points in the provided range.
- def query_k_nearest (self, point, k)

  Returns k points closest to the provided point.
- def print_all_points (self, root)

  Prints all points stored in the PRQuadtree.
- def __str__ (self)

  Prints the points of the nw,ne,sw,se blocks of the given PRQuadTree node.

Static Public Member Functions

- def size (prtree)

  Static method that determines the size of the given tree.

Public Attributes

- box
- points
- nw
- ne
- sw
- se

Static Public Attributes

- int QT_NODE_CAPACITY = 20

6.5.1 Detailed Description

Class representing a Point Range Quadtree.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 def prquadtree.PRQuadTree.__init__ (*self*, *box*)

Constructs a PR Quadtree given an initial square.

Parameters

box	Box representing initial square

6.5.3 Member Function Documentation

6.5.3.1 def prquadtree.PRQuadTree.__str__ (*self*)

Prints the points of the nw,ne,sw,se blocks of the given PRQuadTree node.

Returns

String A string of points in the blocks Generates string based on the number of points stored in the provided node.

Parameters

loc	PRQuadTree a PRQuadTree node (ie nw,ne,sw,se)
name	String

Returns

String A string with point and name

6.5.3.2 def prquadtree.PRQuadTree.insert (*self*, *point*)

Inserts a point into the PRQuadtree.

Parameters

point	Point

Returns

A boolean returning true on success, false on failure.

6.5.3.3 def prquadtree.PRQuadTree.print_all_points (*self*, *root*)

Prints all points stored in the PRQuadtree.

Parameters

root	PRQuadTree start point, or the root of the Quadtree

Returns

String a string with coordinates

6.5.3.4 def prquadtree.PRQuadTree.query_k_nearest (*self, point, k*)

Returns k points closest to the provided point.

Parameters

point	Point a Point from which to search for other points.
k	int number of closest points to return

Returns

array A coordinate distance between the search point and the provided point Internal method used to provide python method with a key (coordinate distance) on which to sort.

Parameters

p	Point

Returns

float

6.5.3.5 def prquadtree.PRQuadTree.query_range (*self*, *rng*)

Returns the points in the provided range.

Parameters

rng	Box a Box range from which to retrieve points

Returns

A list of points within the provided range

6.5.3.6 def prquadtree.PRQuadTree.size (*prtree*) `[static]`

Static method that determines the size of the given tree.

Keeping an insertion count in the client code would be preferred to this due to heavy recursion.

Parameters

prtree	PRQuadTree

Returns

int An integer representing the number of points in the given tree.

6.5.4 Member Data Documentation

6.5.4.1 prquadtree.PRQuadTree.box

6.5.4.2 prquadtree.PRQuadTree.ne

6.5.4.3 prquadtree.PRQuadTree.nw

6.5.4.4 prquadtree.PRQuadTree.points

6.5.4.5 int prquadtree.PRQuadTree.QT_NODE_CAPACITY = 20 [static]

6.5.4.6 prquadtree.PRQuadTree.se

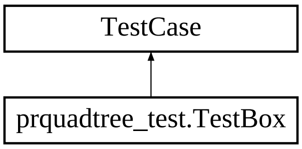
6.5.4.7 prquadtree.PRQuadTree.sw

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

- prquadtree.py

6.6 prquadtree_test.TestBox Class Reference

Inheritance diagram for prquadtree_test.TestBox:



Public Member Functions

- def test_box_insert (self)
- def test_box_contains (self)

6.6.1 Member Function Documentation

6.6.1.1 def prquadtree_test.TestBox.test_box_contains (*self*)

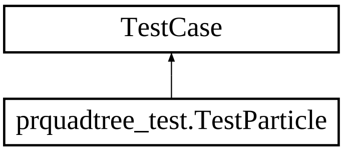
6.6.1.2 def prquadtree_test.TestBox.test_box_insert (*self*)

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

- prquadtree_test.py

6.7 prquadtree_test.TestParticle Class Reference

Inheritance diagram for prquadtree_test.TestParticle:



Public Member Functions

- def test_particle_insert (self)

6.7.1 Member Function Documentation

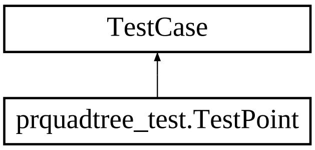
6.7.1.1 def prquadtree_test.TestParticle.test_particle_insert (*self*)

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

- prquadtree_test.py

6.8 prquadtree_test.TestPoint Class Reference

Inheritance diagram for prquadtree_test.TestPoint:



Public Member Functions

- def test_point_insert (self)

6.8.1 Member Function Documentation

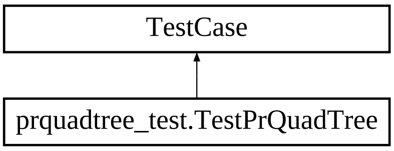
6.8.1.1 def prquadtree_test.TestPoint.test_point_insert (*self*)

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

- prquadtree_test.py

6.9 prquadtree_test.TestPrQuadTree Class Reference

Inheritance diagram for prquadtree_test.TestPrQuadTree:



Public Member Functions

- def test_insert (self)
- def test_nearby (self)

6.9.1 Member Function Documentation

6.9.1.1 def prquadtree_test.TestPrQuadTree.test_insert (*self*)

6.9.1.2 def prquadtree_test.TestPrQuadTree.test_nearby (*self*)

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

- prquadtree_test.py

Chapter 7

File Documentation

7.1 basket.py File Reference

Namespaces

- basket

Functions

- def basket.is_basket_middle (img)

  Single entry function returning True/False if basket is in the middle of the screen.
- def basket.run_middle ()

  Runs continuously and prints if the best detected blob is in the middle.
- def basket.run

  Runs continuously outlines best matched blob if it is in the middle.

Variables

- basket.particle_filter = None
- int basket.image_half_size = -1
- int basket.save_count = 1
- tuple basket.base_filename = datetime.now()

7.2 basket_runner.py File Reference

Namespaces

- basket_runner

7.3 basket_test.py File Reference

Namespaces

- basket_test

Functions

- def basket_test.unitTest (actual, expected, name)
- def basket_test.basketPresent ()
- def basket_test.basketMissing ()

7.4 experiment.py File Reference

Namespaces

- experiment

Functions

- def experiment.experiment
- def experiment.hard_threshold (img)
- def experiment.binary_mask (img)
- def experiment.dilation_and_blur (img)
- def experiment.blobs_by_mask (img)

7.5 image_support.py File Reference

Namespaces

- image_support

Functions

- def image_support.external_init_particle_filter (img)

  Initializes particle filter.
- def image_support.image_hue_filter

  Converts given image to HSV based on the given color.
- def image_support.get_hue_blobs (img)

  Gets basket blobs after hue distance filtering.
- def image_support.get_best_blob (blobs, particle_filter)

  Returns the best blob out of the provided set and particle filter.
- def image_support.is_blob_in_middle_helper (img, blob)

  Determines whether the given blob is in ceter of image.

7.6 particlefilter.py File Reference

Classes

- class particlefilter.ParticleFilter

Namespaces

- particlefilter

7.7 prquadtree.py File Reference

Classes

- class prquadtree.Point

  Represents an (x,y) coordinate point on a grid.
- class prquadtree.Particle

  Represents particle point.
- class prquadtree.Box

  Class defining a square on the coordinate system via a center point and half of square width.
- class prquadtree.PRQuadTree

  Class representing a Point Range Quadtree.

Namespaces

- prquadtree

7.8 prquadtree_test.py File Reference

Classes

- class prquadtree_test.TestPoint
- class prquadtree_test.TestParticle
- class prquadtree_test.TestBox
- class prquadtree_test.TestPrQuadTree

Namespaces

- prquadtree_test

7.9 prquadtree_test_example.py File Reference

Namespaces

- prquadtree_test_example

Variables

- tuple prquadtree_test_example.b = Box(Point(5,5), 50)
- tuple prquadtree_test_example.b2 = Box(Point(50,50), 50)
- tuple prquadtree_test_example.qt = PRQuadTree(b2)
- tuple prquadtree_test_example.pt = Point(2,2)
- tuple prquadtree_test_example.nearby = qt.query_k_nearest(pt, 20)
- int prquadtree_test_example.c = 1

7.10 tennis_ball.py File Reference

Namespaces

- tennis_ball

Functions

- def tennis_ball.is_ball_middle (img)

  Entry point for module which determines whether tennis ball is in the middle of the image.
- def tennis_ball.run ()

  Continuously captures image from computer camera and feeds it to the is_ball_middle method to detect whether tennis ball is in the middle of the screen.

Variables

- tennis_ball.particle_filter = None

7.11 tennis_ball_runner.py File Reference

Namespaces

- tennis_ball_runner

7.12 tennis_ball_test.py File Reference

Namespaces

- tennis_ball_test

Functions

- def tennis_ball_test.unitTest (actual, expected, name)
- def tennis_ball_test.ballPresent ()
- def tennis_ball_test.ballMissing ()

7.13 visual.h File Reference

Functions

- int start_visual (void)
- void set_objects (object_t ∗objs)
- void get_objects (object_t ∗objs, point_t ∗locations)
- void stop_visual (void)

7.13.1 Function Documentation

7.13.1.1 void get_objects (object_t ∗ *objs,* point_t ∗ *locations*)

7.13.1.2 void set_objects (object_t ∗ *objs*)

7.13.1.3 int start_visual (void)

7.13.1.4 void stop_visual (void)

Index