
Fiducial Registration Educational Demonstration Documentation

Stephen Thompson

Jan 26, 2022

CONTENTS

1	Citing	3
2	Developing	5
3	Licensing and copyright	7
4	Acknowledgements	9
	Python Module Index	15
	Index	17

Author: Stephen Thompson

This is the Fiducial Registration Educational Demonstration (SciKit-SurgeryFRED), part of the [SciKit-Surgery](#) software project, developed at the [Wellcome EPSRC Centre for Interventional and Surgical Sciences](#), part of [University College London \(UCL\)](#).

Fiducial Registration Educational Demonstration is intended to be used as part of an online tutorial in using fiducial based registration. The tutorial covers the basic theory of fiducial based registration, which is used widely in image guided interventions. The tutorial aims to help the students develop an intuitive understanding of key concepts in fiducial based registration, including Fiducial Localisation Error, Fiducial Registration Error, and Target Registration Error.

CITING

If you use SciKit-SurgeryFRED in your research or teaching please cite our paper:

Stephen Thompson, Tom Dowrick, Mian Ahmad, Jeremy Opie, and Matthew J. Clarkson “Are fiducial registration error and target registration error correlated? SciKit-SurgeryFRED for teaching and research”, Proc. SPIE 11598, Medical Imaging 2021: Image-Guided Procedures, Robotic Interventions, and Modeling, 115980U (15 February 2021); <https://doi.org/10.1117/12.2580159>

Specific releases can be cited via the Zenodo tag.

SciKit-Surgery can also be cited as:

Thompson S, Dowrick T, Ahmad M, et al. “SciKit-Surgery: compact libraries for surgical navigation.” International Journal of Computer Assisted Radiology and Surgery. 2020 May. <https://doi.org/10.1007/s11548-020-02180-5>

2.1 Cloning

You can clone the repository using the following command:

```
git clone https://github.com/SciKit-Surgery/scikit-surgeryfred
```

2.2 Contributing

Please see the [contributing guidelines](#).

2.3 Useful links

- [Source code repository](#)

LICENSING AND COPYRIGHT

Copyright 2020 University College London. Fiducial Registration Educational Demonstration is released under the BSD-3 license. Please see the [license file](#) for details.

ACKNOWLEDGEMENTS

Supported by [Wellcome](#) and [EPSRC](#).

4.1 Dependency Graph

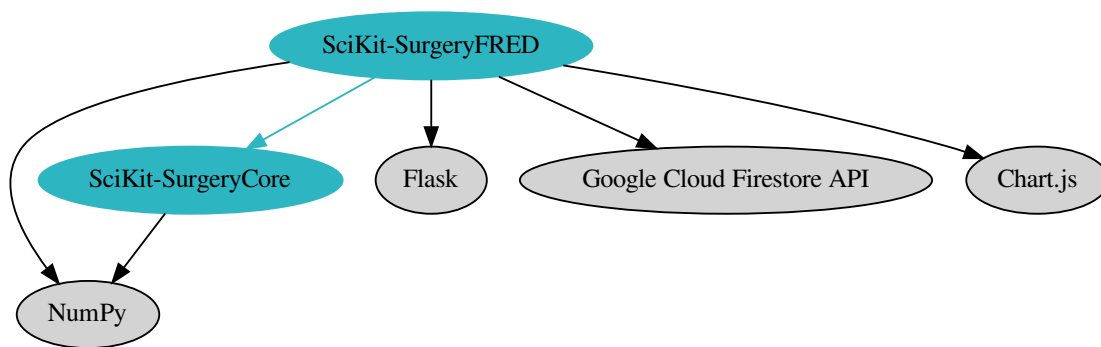


Fig. 1: SciKit-SurgeryFRED's Dependencies

4.2 stable

4.2.1 get_results module

4.2.2 main module

Module to handle communication between client (static/main.js) and sksurgeryfred server

`main.addhighscore()`
add your score to the high scores

`main.calculatescore()`
Delegates to `sksurgery.algorithms.score` to calculate an ablation score.

main.correlation()

Takes in 2d array, and does linear fit and correlation for each column against the first returns slope, intercept and correlation coefficient if there are less than 4 data points it returns false.

main.defaultcontour()

Returns a pre-calculated contour image to represent the intraoperative image.

main.favicon()

returns the icon

main.getfle()

Returns values for fiducial localisation errors Values are randomly selected from a uniform distribution from 0.5 to 5.0 pixels

main.gethighscores()

return the sorted high scores, the ranking and the ref to the lowest score

main.gettarget()

Returns a target point for the simulated intervention

main.index()

returns the main page, template/index.html

main.initdatabase()

here we will create a new document in collection results and return the name of the document. Write some stuff about the date and the versions of fred, core, and fredweb. Create a sub collection of results within the document

main.placefiducial()

Returns the location of a fiducial marker on the pre- and intra-operative images. FLE is added to each marker location.

main.register()

Performs point based registration and returns registration data as json.

main.startfred()

returns the fred page

main.writegameresults()

write the game results to a firestore database

main.writeresults()

write the results to a firestore database

4.2.3 sksurgeryfred package

Subpackages

sksurgeryfred.algorithms package

Submodules

sksurgeryfred.algorithms.errors module

Functions for point based registration using Orthogonal Procrustes.

sksurgeryfred.algorithms.errors.expected_absolute_value(*std_devs*)

Returns the expected absolute value of a normal distribution with mean 0 and standard deviations *std_dev*

skurgeryfred.algorithms.fle module

Functions for adding fiducial localisation error

class skurgeryfred.algorithms.fle.**FLE**(*independent_fle=None, ind_fle_function=None, systematic_fle=None, sys_fle_function=None, dimension=3*)

Bases: object

Provides methods to add Fiducial Localisation Error to a point

Parameters

- **independent_fle** – the magnitude(s) of the independent FLE's, used for the default `ind_fle_function`. Do not use if using your own `ind_fle_function`. A single float will yield isotropic error, or an array can be passed for anisotropic errors.
- **ind_fle_function** – the function to use for sampling the independent fle. Defaults to `numpy.random.normal`
- **systematic_fle** – the magnitude(s) of the systematic FLE's, used for the default `sys_fle_function`. Do not use if using your own `sys_fle_function`. A single float will yield isotropic error, or an array can be passed for anisotropic errors.
- **sys_fle_function** – the function to use for sampling the independent fle. Defaults to `numpy.add`
- **dimension** – the dimensions to use, defaults to 3.

Raises

- **ValueError** – If `independent_fle` is not single value or array of length dimension.
- **ValueError** – If both fle function and fle value are set.
- **TypeError** – If either error function is invalid.

perturb_fiducial(*fiducial_marker*)

Adds the FLE to the marker position

Parameters **fiducial_marker** – the true position of the marker.

Returns The perturbed position of the marker

skurgeryfred.algorithms.fred module

Functions to support MedPhys Taught Module workshop on calibration and tracking

skurgeryfred.algorithms.fred.is_valid_fiducial(*fiducial_location*)

Checks the x, y, and z location of a fiducial :returns: true if a valid fiducial

skurgeryfred.algorithms.fred.make_target_point(*outline, edge_buffer=0.9*)

returns a target point, that should lie within the outline.

sksurgeryfred.algorithms.point_based_reg module

Functions to support MedPhys Taught Module workshop on calibration and tracking

class sksurgeryfred.algorithms.point_based_reg.**PointBasedRegistration**(*target, fixed_fle_esv, moving_fle_esv*)

Bases: object

Does the registration and associated measures

get_transformed_target()

Returns transformed target and status

register(*fixed_points, moving_points*)

Does the registration

reinit(*target, fixed_fle_esv, moving_fle_esv*)

reinitiatilises the target and errors

sksurgeryfred.algorithms.scores module

Functions for calculating the score for ablation game

sksurgeryfred.algorithms.scores.**calculate_score**(*target_centre, est_target_centre, target_radius, margin*)

Calculates the score for a given simulated ablation :params target_centre: The known target position :params est_target_centre: The target centre estimated by registration :target_radius: The radius of the target :margin: The margin to add (treatment radius = target_radius + margin :returns: the score

sksurgeryfred.algorithms.scores.**sphere_volume**(*radius*)

Returns the volume of a sphere of radius

sksurgeryfred.algorithms.scores.**two_sphere_overlap_volume**(*centre0, centre1, radius0, radius1*)

Calculates the overlapping volume of two spheres from <https://math.stackexchange.com/questions/297751/overlapping-spheres> :param: centre0 centre of sphere0 (1x3) :param: centre1 centre of sphere1 (1x3) :param: radius0 radius of sphere0 (1) :param: radius1 radius of sphere1 (1)

Module contents

sksurgeryfred.utilities package

Submodules

sksurgeryfred.utilities.get_results module

Script to download and parse results from firestore

sksurgeryfred.utilities.get_results.**get_results**(*testing=False*)

Function to download results from our results database and parse them ready for analysis

sksurgeryfred.utilities.results_database module

Utilities to enable testing of FRED's firestore databases

class sksurgeryfred.utilities.results_database.**ResultsDatabase**(*teststring*)

Bases: object

Stands in as a fake database for testing purposes

class sksurgeryfred.utilities.results_database.**TestAddSet**

Bases: object

implements a get function for testing purposes

class sksurgeryfred.utilities.results_database.**TestCollection**(*teststring*)

Bases: object

A pretend collection for testing purposes

class sksurgeryfred.utilities.results_database.**TestDoc**

Bases: object

implements a document for testing purposes

class sksurgeryfred.utilities.results_database.**TestGet**(*teststring*)

Bases: object

implements a get function for testing purposes

class sksurgeryfred.utilities.results_database.**TestScore**(*name, score, ref*)

Bases: object

stores a name and a score plus implements to_dict for testing

to_dict()

returns a dictionary containing name and score

Module contents

Module contents

FiducialRegistrationEducationalDemonstration

- modindex
- genindex
- search

PYTHON MODULE INDEX

a

- `skssurgeryfred.algorithms`, 12
- `skssurgeryfred.algorithms.errors`, 10
- `skssurgeryfred.algorithms.fle`, 11
- `skssurgeryfred.algorithms.fred`, 11
- `skssurgeryfred.algorithms.point_based_reg`, 12
- `skssurgeryfred.algorithms.scores`, 12

m

- `main`, 9

s

- `skssurgeryfred`, 13

u

- `skssurgeryfred.utilities`, 13
- `skssurgeryfred.utilities.get_results`, 12
- `skssurgeryfred.utilities.results_database`, 13

INDEX

A

`addhighscore()` (in module *main*), 9

C

`calculate_score()` (in module *sksurgeryfred.algorithms.scores*), 12
`calculatescore()` (in module *main*), 9
`correlation()` (in module *main*), 9

D

`defaultcontour()` (in module *main*), 10

E

`expected_absolute_value()` (in module *sksurgeryfred.algorithms.errors*), 10

F

`favicon()` (in module *main*), 10
FLE (class in *sksurgeryfred.algorithms.fle*), 11

G

`get_results()` (in module *sksurgeryfred.utilities.get_results*), 12
`get_transformed_target()` (*sksurgeryfred.algorithms.point_based_reg.PointBasedRegistration* method), 12
`getfle()` (in module *main*), 10
`gethighscores()` (in module *main*), 10
`gettarget()` (in module *main*), 10

I

`index()` (in module *main*), 10
`initdatabase()` (in module *main*), 10
`is_valid_fiducial()` (in module *sksurgeryfred.algorithms.fred*), 11

M

main
 module, 9
`make_target_point()` (in module *sksurgeryfred.algorithms.fred*), 11

module

main, 9
sksurgeryfred, 13
sksurgeryfred.algorithms, 12
sksurgeryfred.algorithms.errors, 10
sksurgeryfred.algorithms.fle, 11
sksurgeryfred.algorithms.fred, 11
sksurgeryfred.algorithms.point_based_reg, 12
sksurgeryfred.algorithms.scores, 12
sksurgeryfred.utilities, 13
sksurgeryfred.utilities.get_results, 12
sksurgeryfred.utilities.results_database, 13

P

`perturb_fiducial()` (*sksurgeryfred.algorithms.fle.FLE* method), 11
`placefiducial()` (in module *main*), 10
PointBasedRegistration (class in *sksurgeryfred.algorithms.point_based_reg*), 12

R

`register()` (in module *main*), 10
`register()` (*sksurgeryfred.algorithms.point_based_reg.PointBasedRegistration* method), 12
`reinit()` (*sksurgeryfred.algorithms.point_based_reg.PointBasedRegistration* method), 12
ResultsDatabase (class in *sksurgeryfred.utilities.results_database*), 13

S

sksurgeryfred
 module, 13
sksurgeryfred.algorithms
 module, 12
sksurgeryfred.algorithms.errors
 module, 10
sksurgeryfred.algorithms.fle
 module, 11
sksurgeryfred.algorithms.fred

module, 11
sksurgeryfred.algorithms.point_based_reg
module, 12
sksurgeryfred.algorithms.scores
module, 12
sksurgeryfred.utilities
module, 13
sksurgeryfred.utilities.get_results
module, 12
sksurgeryfred.utilities.results_database
module, 13
sphere_volume() (in module
sksurgeryfred.algorithms.scores), 12
startfred() (in module main), 10

T

TestAddSet (class in sksurgeryfred.utilities.results_database),
13
TestCollection (class in
sksurgeryfred.utilities.results_database),
13
TestDoc (class in sksurgeryfred.utilities.results_database),
13
TestGet (class in sksurgeryfred.utilities.results_database),
13
TestScore (class in sksurgeryfred.utilities.results_database),
13
to_dict() (sksurgeryfred.utilities.results_database.TestScore
method), 13
two_sphere_overlap_volume() (in module
sksurgeryfred.algorithms.scores), 12

W

writegamerresults() (in module main), 10
writeresults() (in module main), 10