Release

1.1

UNIVERSITY OF KENT

Applied Optics Group

Fibre Bundle Simulator (Matlab)

Introduction

Fibre Bundle Simulator (Matlab) allows image acquisition through a fibre images to be simulated. The simulation does not model multimodal behaviour of the fibre cores, instead each core is modelled as a 2D intensity function such as a circle or Gaussian. The fibre bundle parameters are defined using **make_bundle** and the distal object is sampled using **sample_with_bundle.** An image of the proximal face of the bundle is then obtained using **project_cores**. Alternatively, to quickly create an image, **generate_bundle_image** performs both of these functions.

License

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details https://www.gnu.org/licenses

Functions

make_bundle

Generates a matlab fibre bundle struct containing all the information necessary to simulate image acquisition through a bundle.

Prototype:

bundle = make_bundle(coreSpacing, coreType, coreSize, bundleRadius, coreRandom, transferRandom)

Parameters:

coreSpacing: Centre-centre core spacing in mm
coreType: Intensity distribution of power in core, 'Gaussian' or 'Circular'

coreSize: Diameter of the core circle ('circular') or core FWHM ('Gaussian')
bundleRadius: Radius of fibre bundle in mm
coreRandom: Standard deviation of core positions relative to perfect hexagonal array (mm) (0 = perfect array)
transferRandom: standard deviation of throughput of fibres around a mean of 1 (0 = all fibres are identical)

get_core_sampling_function

Generates a 2D array containing a power distribution function for each core. Each core appears to have this 2D intensity in images of the bundle face, and the bundle will sample objects using the same intensity weighting.

Prototype:

samplingFunction = get_core_sampling_function (type, size, scale)

Parameters:

type: 'Circular' or 'Gaussian'
scale: Pixels per mm
size: The diameter of the circle ('circular') or FWHM ('Gaussian')

Returns:

samplingFunction: 2D array containing sampling function, normalize to have a sum of 1

generate_bundle_image

Simulates the acquisition of an image using a pre-defined fibre bundle. Samples the object using **sample_with_bundle** and then generates a proximal image using **project_cores**.

Prototype:

finalImage = generate_bundle_image(sampleImage, bundle, inScale, outScale)

Parameters:

sampleImage: The object to be sampled
bundle: Struct describing the fibre bundle with which to acquire the image, can be generated using make_bundle
inScale: Scale of sampleImage (pixels/mm)
outScale: Desired scale of returned image (pixels/mm)

Returns

finalImage: 2D double array containing image pixel values

generate_spot_centres

Generates a hexagonal array of spots for use in simulating a fibre bundle.

Prototype:

[xVec yVec] = generateSpotCentres(coreSpacing, radius, spotSD)

Parameters

coreSpacing: average distance between cores
radius: radius of fibre bundle
spotSD: standard deviation of Gaussian noise on spot position (0 for no error)

Returns

xVec: vector of x co-ordinates of spot centres
yVec: vector of y co-orindates of spot centres

sample_with_bundle

Samples an image using a simulated fibre bundle. Each fibre core performed a weighted sample of an area of the image given by the 2D normalised array sampling function.

Prototype:

sampledValues = sample_with_bundle(im, scale, bundle);

Parameters:

im: image to sample
scale: image scale (pixels per mm)
bundle: struct defining fibre bundle, created by make_bundle

Returns:

sampledValues: vector containing intensity for each core

project_cores

Generates an image of a fibre bundle end face.

Prototype:

bundleImage = project_cores(bundle, intensity, scale, imSize)

Parameters:

intensity: vector containing intensity of light transmitted by core

scale: pixels per mm for the output image

imSize: number of pixels of output image. Set -1 to make the image just large enough to enclose the bundle. Set equal to size of sampled image to maintain pixel size

Returns:

bundleImage: 2D double array containing image of end of bundle

Fibre Bundle Struct

The fibre bundle struct contains the following variables:

coreSpacing: centre-centre core spacing in mm

coreType: 'Gaussian' or 'Circular'

bundleRadius: radius of fibre bundle in mm

numCores: number of fibre cores in bundle

coreCentreX: vector containing centre position (x) of each core (mm). (0 = centre of bundle)

coreCentreY: vector containing centre position (y) of each core (mm). (0 = centre of bundle)

coreRandom: standard deviation of core centres from a perfect hexagonal lattice

transferRandom: standard deviation of core transfer function (i.e. percentage of power retained).

transfer: vector containing the transfer function for each core (i.e. the fraction of power transmitted by the core). The cores have a mean transfer of 1.