User manual

/\*

\* **Discrete Sine Transform and Inverse Discrete Sine Transform** (DST/IDST one to four)

\* DST/IDST I ---> IV

\* This CUDA code can handle/work with any type of the input mxArrays,

\* GPUarray or standard matlab CPU array as input {prhs [0]:= mxGPUArray or CPU Array}

\* GpuArray output, B=DSTI🡪IV\_Column/row (A) or B=DST\_GPU/ IDST\_GPU (A, type of DST, dimensions).

\* Developed at UCL, Institute of Neurology, 12 Queen Square, WC1N 3AR, London

\* Welcome Trust Centre for Neuroimaging

\* Part of the project SPM (http://www.fil.ion.ucl.ac.uk/spm)

\* Copyright 2018

\* Kevin Bronik

\*/

## **To compute DST user can choose either the following syntaxes:**

DST

[y](https://uk.mathworks.com/help/signal/ref/dct.html#outputarg_y) = DSTI🡪IV\_Column/row (x)

y = DST\_I\_Column(x)

y = DST\_I\_Row(x)

y = DST\_II\_Column(x)

y = DST\_II\_Row(x)

.

.

.

Inverse DST

[y](https://uk.mathworks.com/help/signal/ref/dct.html#outputarg_y) = DSTI🡪IV\_Column/row\_Inverse (x)

y = DST\_I\_Column\_Inverse(x)

y = DST\_I\_ Row \_Inverse(x)

y = DST\_II\_Column\_Inverse(x)

y = DST\_II\_ Row \_Inverse(x)

.

.

.

Which will returns the unitary discrete sine transform of input array x. The output y has the same size/type as x. Both the input and output arrays are gpuArray/cpuarray objects.

.

## **Or the following syntaxes:**

**[Note that the name “DST\_GPU/ IDST\_GPU” has been chosen to avoid any kind of confusion with the existing MATLAB’s dst/idst]**

DST

y=DST\_GPU (x, type of DST, dimensions)

Type of DST= one, two, three, four

Dimensions= row, column

Example:

a = ones (15, 6,'gpuArray');

c = DST\_GPU (a,'two','row')

Inverse DST

y=IDST\_GPU (x, type of inverse DST, dimensions)

Type of inverse DST= one, two, three, four

Dimensions= row, column

Example:

e = IDST\_GPU(c,'two','row')