This code aims at identifying sea-ice floes, sea-ice types, and presenting a color map of the ice floes and brash pieces based on sizes, and the corresponding ice floe size distribution histogram.

The GVF snake algorithm used in this code is an open source code, which is available on: <http://www.iacl.ece.jhu.edu/static/gvf/> .

This code is only available for academic non-commercial use. If you are using this code, please give a reference to:

Zhang, Q. and R. Skjetne, “Image processing for identification of sea-ice floes and the floe size distributions.” *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 53, No. 5, pp. 2913-2924, 2015.

Function: **seaice\_kmean\_GVF\_forenhancement**

Sea-ice image segmentation program. This program is suit for non-ridged ice floe in the marginal ice zone, and managed ice resulting from offshore operations in sea-ice.

Example:

[seg, bk] = **seaice\_kmean\_GVF\_forenhancement**( I, kms0, sigma, GradientOn, GVFOn, Num, mu, iter, alpha, beta, gamma, kappa, Dmin, Dmax, N, Ra\_min, Ra, Rc, Rl, se, timer);

Input:

**I**: the input ice image (the same as in **GVF\_distance**).

% parameters for image

**kms0**: the number of clusters for k-means clustering to detect all the ice.

**Ra\_min**: the minimum ice piece area, the ice piece area less than **Ra\_min** is considered as noise (the same as in **GVF\_distance**).

**Ra**: the maximum ice piece area, the ice piece area larger than **Ra** is considered as under-segmentation, another turn of segmentation is required (the same as in **GVF\_distance**).

**Rc**: convexity threshold, the ice piece convexity less than **Rc** is considered as under-segmentation, another turn of segmentation is required (the same as in **GVF\_distance**).

**Rl**: threshold ratio between length and width, ratio between length and width larger than **Rl** is considered as under-segmentation, another turn of segmentation is required (the same as in **GVF\_distance**).

**timer**: limit the maximum iterations of segmentation (the same as in **GVF\_distance**).

**se**: morphology structure element, for combining the nearby local maxima (the same as in **GVF\_distance**).

% parameters for GVF Snake, MOST TUNE!

**Num**: number of GVF iterations, the most important, tune parameter. For weak boundaries, large ice floe, high resolution image, **Num** should be increased (the same as in **GVF\_distance**).

**iter**: number of Snake iterations, adjust it according to the over- and under-segmentation result (the same as in **GVF\_distance**).

**N**: iter of deformation, adjust it according to the over- and under-segmentation result (the same as in **GVF\_distance**).

(The computation time of the algorithm will increase when increasing **Num**, **iter**, **N**.)

(Do not tune parameters which are not mentioned above.)

Output:

seg: the segmented sea-ice image.

bk: binary non-segmented ice detection image.

Function: **ice\_shape\_enhancement**

Enhance the ice floe (brash) shape; separate the sea-ice image into four layers: ice floe, brash ice, slush, and water; calculate the floe and brash size.

Example:

[out, index\_floe, ice\_floe, index\_brash, brash\_ice, index\_slush, index\_water, index\_residue, coverage] = ice\_shape\_enhancement(bk, seg, min\_floe, min\_brash, se\_th);

Input:

bk: binary non-segmented ice detection image (output of **seaice\_kmean\_GVF\_forenhancement**).

seg: the segmented sea-ice image (output of **seaice\_kmean\_GVF\_forenhancement**).

% parameters for shape enhancement

**min\_floe**: minimum floe area, to identify ice floe from brash.

**min\_brash**: minimum brash area.

**se\_th**: threshold for adaptive morphology structure element.

Output:

Out: morphology cleaned ice pieces (ice floe & brash ice).

index\_floe: layer of ice floe.

ice\_floe: information about each identified ice floe.

index\_brash: layer of brash ice.

brash\_ice: information about each identified brash ice.

index\_slush: layer of slush ice.

index\_water: layer of water.

index\_floe: layer of ice floe.

index\_residue: layer of residue.

coverage: coverage of each layer.

Function: sea\_ice\_model

Polygon-fit for each sea-ice floe, and disk-fit for each brash ice.

Example:

[floe, brash] = sea\_ice\_model( ice\_floe, brash\_ice, index\_floe );

Input:

ice\_floe: structure data of identified sea-ice floes (output of ice\_shape\_enhancement).

brash\_ice: structure data of identified brash ice (output of ice\_shape\_enhancement).

img: segmented (identified) binary image

Output:

floe: structure data, includes: Vertices, Center, Area, Perimeter, and overlapping floes.

brash: structure data, includes: Vertices, Center, Area, Perimeter, and overlapping brash.

An example of sea-ice image segmentation is given in the file: sea\_ice\_demo.m.