```
A quick way to process two input images is
        im1 = imread('your_first_frame_name');
        im2 = imread('your_second_frame_name');
        uv = estimate_flow_interface(im1, im2, 'classic+nl-fast');
The output uv is an M x N x 2 matrix.

You can also run the demo program estimate_flow_demo.m and follow it to try different
methods and parameter setting. Have fun!

Any scientific work that makes use of our code should appropriately mention this in the
text and cite our CVPR 2010 paper (see below).

Note that the program was written in a Linux environment with MATLAB R2008b. It has only
been tested on one Windows machine (Windows7 with MATLAB R2007b) and one Mac machine
(Matlab 2009b). Please report problems you encounter to us (dqsun at cs.brown.edu).

Output:
uv = estimate_flow_demo; The estimated flow field on "RubberWhale" has an average
angular error (AAE) of 2.401 degrees and an average end-point error (EPE) of 0.076.

Note that the default method is a fast version of the reported "Classic+NL" method on
the Middlebury benchmark. Please read the comments in the beginning of the demo program
for more methods.

Data:
Please download training and test data
other-color-twoframes.zip, other-gt-flow.zip, and eval-color-twoframes.zip
from http://vision.middlebury.edu/flow/data/
and extract them to data/  (You can change the filepath in the beginning of
utils/local/read_flow_file.m "filePath = 'data/';" to the folder that you save the data)

There might be a tiny difference for some methods ("classic-c" and "ba") and sequences
from those reported in the technical report, if you run estimate_flow_deom.m directly.
Results reported in the technical report were obtained using the gray images from the
Middlebury website. estimate_flow_demo.m takes the color images and converts them to the
gray images with the MATLAB built-in function rgb2gray, but the converted gray images
are slightly different from those directly downloaded from the Middlebury website.


You may need to compile utils/mex/sor.pp file to use the sor solver or download some
MATLAB sor solver.

Possible errors
a)
Error using ==> \
Out of memory. Type HELP MEMORY for your option.

Uncomment this line in estimate_flow_interface.m
%ope.solver    = 'pcg';

Acknowledgment:
Thanks to T. A. Davis, Y. Rubner, and M. A. Ruzon for their public source codes.
Thanks to J. Gai, F. Li, and J. Wang for useful feedback on the previous code for HS and
BA.

References:
Sun, D.; Roth, S. & Black, M. J. "Secrets of Optical Flow Estimation and Their
Principles" IEEE Int. Conf. on Comp. Vision & Pattern Recognition, 2010
Sun, D.; Roth, S. & Black, M. J. "A Quantitative Analysis of Current Practices in
```

Optical Flow Estimation and The Principles Behind Them" Technical Report Brown-CS-10-03, 2010