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-fastp');
```

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 IJCV or CVPR paper (see below).

Typical input and output

```
uv = estimate_flow_demo;
```

output “AAE 2.371 average EPE 0.075”

```
im1 = imread('data/sintel/ambush_5/frame_0036.png');
im2 = imread('data/sintel/ambush_5/frame_0037.png');
uv = estimate_flow_interface(im1, im2, 'classic+nl-fastp');
tuv = readFlowFile('data/sintel/ambush_5/frame_0036.flo');
[aae stdae aepe] = flowAngErrUV(tuv, uv, 0);
fprintf('\nAAE %3.3f average EPE %3.3f \n', aae, aepe);
```

output “AAE 19.738 average EPE 19.074”

Possible errors

```
Error using ==> \
Out of memory. Type HELP MEMORY for your option.
```

Uncomment this line in estimate_flow_interface.m

```
%ope.solver = 'pcg';
```

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

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.
Thanks to Cristina Garcia Cifuentes for reporting and fixing bugs on weighted_median.m and scale_image.m.

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" International Journal of Computer Vision, 106(2):115-137, 2014.