Continuous Max-Flow (CMF) Algorithm to 2D/3D Multi-Region Image Segmentation v1.0

Jing Yuan[1]

University of Western Ontario
London, Ontario, Canada N6A 5B7
cn.yuanjing@gmail.com

1 What does this software package solve?

This software package includes the programs which are designed to efficiently solve the 2D/3D multi-region image segmentation problem (Potts Model), based on the fast continuous max-flow method (CMF) proposed by J. Yuan, E. Bae, X-C Tai and Y. Boykov [7].

Multi-region image segmentation, through *Potts model*, can be mathematically formulated as:

$$\min_{\{\Omega_i\}_{i=1}^n} \quad \sum_{i=1}^n \int_{\Omega_i} C_i(x)\,dx + \alpha \sum_{i=1}^n |\partial \Omega_i| \tag{1}$$

$$\text{s.t.} \quad \cup_{i=1}^n \Omega_i \,=\, \Omega\,; \quad \Omega_k \cap \Omega_l \,=\, \emptyset\,, \quad \forall k \neq l \tag{2}$$

where $|\partial\Omega_i|$ measures the perimeter of each disjoint subdomain Ω_i, $i = 1 \ldots n$, which may be weighted; the function $C_i(x)$, $i = 1 \ldots n$, evaluates the cost of assigning the specified position $x \in \Omega$ to the region Ω_i.

As proposed in [8, 4, 1], the *Potts model* based image segmentation problem (1) can be efficiently solved by its convex relaxation:

$$\min_{u \in S} \sum_{i=1}^n \int_\Omega u_i(x)\, C_i(x)\,dx \,+\, \sum_{i=1}^n \int_\Omega \omega(x)\,|\nabla u_i|\;dx \tag{3}$$

where $u_i(x)$, $i = 1 \ldots n$, gives the indicator function of the respective segmented region Ω_i; S is the convex constrained set of $u(x) := (u_1(x), \ldots, u_n(x))$:

$$S \,=\, \{u(x)\,|\,(u_1(x), \ldots, u_n(x)) \,\in\, \triangle_+\,,\; \forall x \in \Omega\,\}\,, \tag{4}$$

$\triangle_+$ is the simplex set, i.e.

$$\text{for } \forall x \in \Omega\,, \quad \sum_{i=1}^n u_i(x) \,=\, 1\,; \quad u_i(x) \in [0,1]\,, \quad i = 1 \ldots n\,.$$

In this document, (3) is called the *Convex Relaxed Potts Model*.

J. Yuan et al [7] proposed the continuous max-flow approach to (3), for which the continuous max-flow model reads:

$$\max_{p_s, p_t, p} \quad \int_\Omega p_s \, dx \tag{5}$$

$$\text{s.t.} \quad (\operatorname{div} p_i + p_t^i - p_s)(x) = 0 \, ; \quad i = 1 \dots n \tag{6}$$

$$p_t^i(x) \leq C_i(x) \, , \quad |p_i(x)| \leq \omega(x) \, ; \quad i = 1 \dots n \, . \tag{7}$$

Remark 1 *In mathematics, J. Yuan et al [7] proved* (5) *is is dual to* (3)*, i.e. the duality btw. continuous max-flow* (5) *and the multi-region cut* (3)*; the authors showed that* (5) *gives a variational flow-maximization explanation to* (3)*. This is similar to the classical duality between max-flow and min-cut.*

A similar work of the continuous max-flow approach to image segmentation with two regions: foreground and background, was proposed in [6, 5]. Its related software package was published at:

$$http://www.mathworks.com/matlabcentral/fileexchange/34126$$

and

$$https://sites.google.com/site/wwwjingyuan \, .$$

Remark 2 *In numerics,* (5) *leads to the* fast continuous max-flow algorithm *[7], which solves* (3) *efficiently and can be easily speeded-up by GPU. Essentially, its CPU version runs as fast as graph-cuts, e.g. α-expansion, [2, 3] and its GPU version runs more than* 6 *times faster than graph-cuts in practice, especially in 3D!*

Remark 3 *This software package implements the fast continuous max-flow algorithms to both 2D and 3D multi-region image segmentation* (3)*. The programs were developed with three programming tools: matlab, C and GPU (CUDA based) respectively.* **They are ready for the matlab usage and can also be easily embeded to other applications.**

2 What does this software package include?

The software package includes the programs solving the 2D/3D multi-region image segmentation problem (3), through the fast continuous max-flow algorithm [7]. All the programs were developed to be ready for matlab usages, with three implementations: matlab, C and CUDA C (GPU). *For the GPU programs, an Nvidia CUDA based GPU card is required.*

We list all the files as follows:

– Matlab programs:

  `CMF_ML_Cut.m, CMF3D_ML_Cut.m`

  are the matlab implementation of the fast continuous max-flow algorithm, for 2D and 3D respectively.

– C programs:

  `CMF_ML_mex.c, CMF3D_ML_mex.c`

  are the implementation of the fast continuous max-flow algorithm by C, for 2D and 3D respectively.

– GPU(CUDA) programs:

  `CMF_ML_GPU.cu, CMF_ML_kernels.cu, CMF3D_ML_GPU.cu, CMF3D_ML_kernels.cu`

  are the implementation of the fast continuous max-flow algorithm by CUDA C, for 2D and 3D respectively. The files *_kernesl.cu give the GPU kernel functions.

– Matlab examples:

  `test_CMF_ML.m, test_CMF3D_ML.m`

  show how to use the associated C and GPU(CUDA) programs in matlab.

– The GNU Licence file and this file:

  `LICENCE.TXT, CMFML_README.pdf`

3 How to use this software package?

To use the matlab programs:

`CMF_ML_Cut.m, CMF3D_ML_Cut.m`

you can directly modify the data and parameters in the corresponding matlab files.

To use the C programs:

`CMF_ML_mex.c, CMF3D_ML_mex.c`

you should first compile them by matlab mex.

To use the GPU programs:

`CMF_ML_GPU.cu, CMF3D_ML_GPU.cu`

you should first compile them by nvmex. The detailed command and configurations can be found at: http://developer.nvidia.com/matlab-cuda .

Once you finish compiling C and GPU programs, they can be used by any matlab program in a very similar way. **For Linux users, you may need the permission to access the GPU card!**

For example, you can use the following steps in matlab to apply the function CMF3D_ML_GPU() (also see the two included matlab examples, for more details):

```
>> define the volumes: w and Ct.
>> define the parameter vector: para
>> [u, erriter, i, timet] = CMF3D_ML_GPU(single(w), single(Ct), single(para));
```

The output u is a 4D volume, which records n 3D segmentation surfaces and each one defines a subset of the given 3D volume. The matlab example:

```
test_CMF3D_ML.m
```

shows how to extract the n 3D segmentation volumes (Fig. 1 (a, b, c, d) demonstrate the 4 computed 3D segmentation surfaces).

The output *erriter* gives the error estimation at each iteration and it is used to show the convergence as follows:

```
>> loglog(erriter,'DisplayName','erriter');figure(gcf)
```

The output i and *timet* show at which step the algorithm converges and the total computing time respectively.

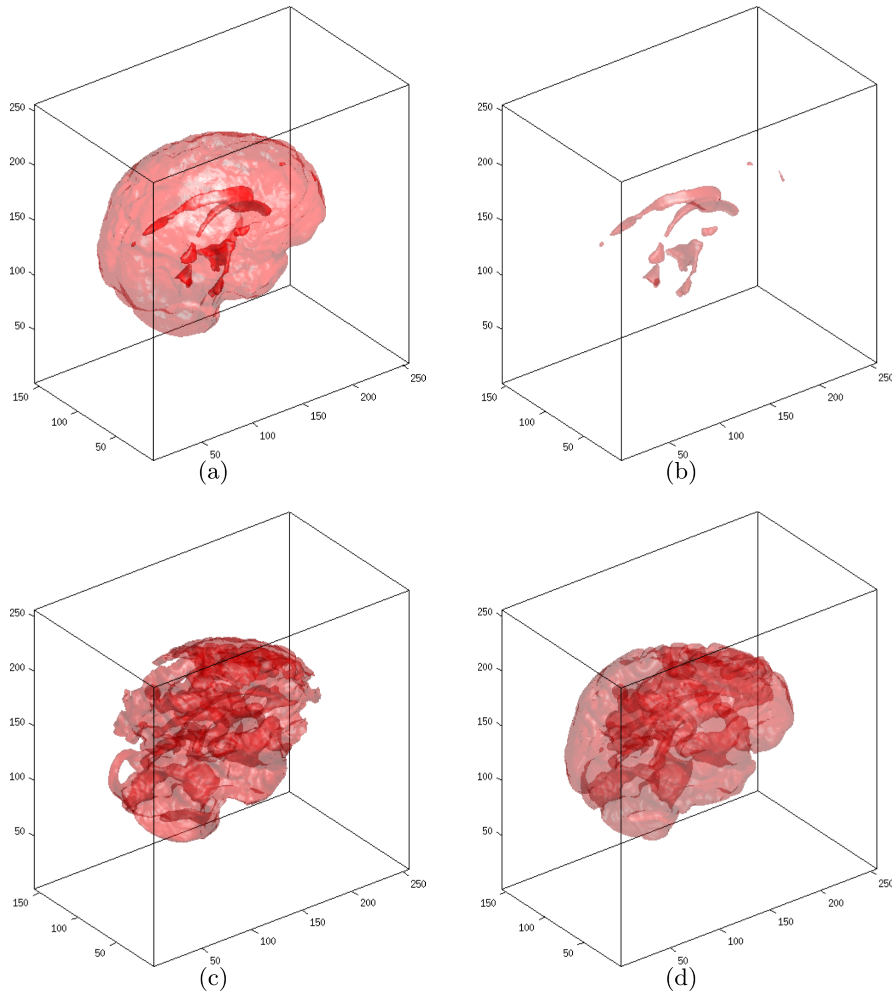

Fig. 1. Example of 3D Multi-Region Segmentation ($256 \times 156 \times 256$ voxels) 4 3D segmentation volumes are computed, which are represented by 4 3D surfaces: (a) (b) (c) (d).

4 Licence

Please email

Jing Yuan (cn.yuanjing@gmail.com)

or

Egil Bae (Egil.Bae@math.uib.no)

for any questions, suggestions and bug reports. If you use this software, you have to reference the paper [7].

5 Contributions

The programs are not partcularly optimized. You are very welcome to provide your suggestions for the improvement of these programs. I will incorporate you in the following list of contributors for your kind contributions:

Jing Yuan	Univ. of Western Ontario, Canada	cn.yuanjing@gmail.com
Egil Bae	Bergen Univ., Norway	Egil.Bae@math.uib.no
Xue-Cheng Tai	Bergen Univ., Norway	tai@cma.uio.no
Yuri Boykov	Univ. of Western Ontario, Canada	yuri@csd.uwo.ca

References

1. Egil Bae, Jing Yuan, and Xue-Cheng Tai. Global minimization for continuous multiphase partitioning problems using a dual approach. *International Journal of Computer Vision*, 92(1):112–129, 2011.
2. Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on PAMI*, 23:1222 – 1239, 2001.
3. Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts. *IEEE Transactions on PAMI*, 26:65–81, 2004.
4. Jan Lellmann, Jörg Kappes, Jing Yuan, Florian Becker, and Christoph Schnörr. Convex multi-class image labeling by simplex-constrained total variation. In *SSVM '09*, pages 150–162, 2009.
5. J. Yuan, E. Bae, X.-C. Tai, and Y. Boycov. A study on continuous max-flow and min-cut approaches. Technical report CAM-10-61, UCLA, 2010.
6. J. Yuan, E. Bae, and X.C. Tai. A study on continuous max-flow and min-cut approaches. In *CVPR*, USA, San Francisco, 2010.
7. Jing Yuan, Egil Bae, Xue-Cheng Tai, and Yuri Boykov. A continuous max-flow approach to potts model. In *ECCV*, 2010.
8. C. Zach, D. Gallup, J.-M. Frahm, and M. Niethammer. Fast global labeling for real-time stereo using multiple plane sweeps. In *VMV 2008*, 2008.