GRASS GSoC2016 Proposal

-Additional segmentation algorithms for i.segment

1. Contact details

- Name and surname: Bo Yang
- Country: United States
- Email: yangb2@mail.uc.edu
- Phone: +1 513-828-4757
- Address: 477 Riddle Rd Apt 3, Cincinnati, OH 45220

- Organization: OSGeo – Open Source Geospatial Foundation
- Mentor Name: Markus Metz and Moritz Lennert
- Title: Additional segmentation algorithms for i.segment

2. Studies

- What is your School and degree?
  Ph. D. student, Geography – University of Cincinnati, USA
- Would your application contribute to your ongoing studies/degree? If so, how?
  Yes, Adding segmentation algorithms for i.segment would greatly contribute to my ongoing studies since I've been processing different types of remote sensing data during my research project. For years I have worked with raster processing algorithms and computation efficiency, for example, coding to process large volume of remote sensing data. Systematically learning and coding the satellite image processing algorithm through the GSoC2016 would be helpful for pursuing my ongoing degree as well as contribute to my dissertation work.

3. Programming and GIS

- Computing experience: operating systems you use on a daily basis, known programming languages, hardware, ecc.
  I am proficient in Python, C/C++ and R and have utilized QGIS, GRASS, ArcGIS a lot in my study and research project. I use windows OS as my daily basis and am proficiency in

software packages such as GRASS, ArcGIS, ENVI, Visual Studio, Microsoft Office, Eclipse, SPSS, R Studio and Photoshop.

4. GSoC participation

- Have you participated to GSoC before?
  No
- Have you submitted/will you submit another proposal for GSoC 2016 to a different org?
  No

Abstract

GRASS GIS has the i.segment which provides the possibility to segment an image into objects. This is a basic step in object-based image analysis (OBIA). Currently, the module only provides one segmentation algorithm: region-growing. The code of i.segment was structured in a way that allows addition of other algorithms. The core of proposed GSoC project would thus be to add a series of these algorithms. It would be more useful and comprehensive to add at least one or two top-down methods to the i.segment module because the current region growing approach only allows bottom-up hierarchical segmentation. New segment methods, such as mean-shift, split-window and watershed, would allow top-down hierarchical segmentation, which could be used in more types of satellite image processing. Special care should be taken for the whole project to code as efficiently as possible, i.e. to make the code run in reasonable time, even for very large images.

Background

Image segmentation or object recognition is the process of grouping similar pixels into unique objects. Segmentation of remote sensing images is a challenging task. A myriad of different methods have been proposed and implemented in recent years. In spite of the huge effort invested in this problem, there is no single approach that can generally solve the problem of segmentation for the large variety of image modalities existing today. The most effective segmentation algorithms are obtained by carefully customizing combinations of components. The parameters of these components are tuned for the characteristics of the image modality used as input and the features of the objects to be segmented [1].

In the GRASS i.segment module currently only region growing and merging algorithm is implemented. Each object found during the segmentation process is given a unique ID and is a collection of contiguous pixels meeting some criteria. Note the contrast with image classification where all pixels similar to each other are assigned to the same class and do not need to be contiguous. The image segmentation results can be useful on their own, or used as a preprocessing step for image classification. The segmentation preprocessing step can reduce noise and speed up the classification [2].

Segmentation Methods

1. Region growing and merging (available in i.segment module)

This segmentation algorithm sequentially examines all current segments in the raster map. The similarity between the current segment and each of its neighbors is calculated according to the given distance formula. The basic approach of a region growing algorithm is to start from a seed region (typically one or more pixels) that are considered to be inside the object to be segmented. The pixels neighboring this region are evaluated to determine if they should also be considered part of the object. If so, they are merge to the region and the process continues as long as new pixels are added to the region.

The similarity between segments and unmerged objects is used to determine which objects are merged. Smaller distance values indicate a closer match, with a similarity score of zero for identical pixels. During normal processing, merges are only allowed when the similarity between two segments is lower than the given threshold value. During the final pass, if a minimum segment size of 2 or larger is given with the minimum size parameter, segments with a smaller pixel count will be merged with their most similar neighbor even if the similarity is greater than the threshold. The threshold must be larger than 0.0 and smaller than 1.0. A threshold of 0 would allow only identical valued pixels to be merged, while a threshold of 1 would allow everything to be merged. Initial empirical tests indicate threshold values of 0.01 to 0.05 are reasonable values to start. It is recommended to start with a low value, e.g. 0.01, and then perform hierarchical segmentation by using the output of the last run as seeds for the next run. Segments will be merged if they meet a number of criteria, including: The pair is mutually most similar to each other (the similarity distance will be smaller than to any other neighbor), and the similarity must be lower than the input threshold. The process is repeated until no merges are made during a complete pass [2].

2. Mean-shift (plan to be implemented during GSoC 2016)

Comaniciu and Meer proposed the mean-shift image segmentation algorithm for the analysis of a complex multimodal feature space and to delineate arbitrarily shaped clusters in it [3]. The basic computational module of the mean-shift algorithm is an old pattern recognition procedure. For discrete data the convergence of a recursive mean shift procedure to the nearest stationary point of the underlying density function and thus its utility in detecting the modes of the density. The mean shift segmentation is a local homogenization technique that is very useful for damping shading or tonality differences in localized objects.

For the algorithm implementation of this case, basically the algorithm replaces each pixel with the mean of the pixels in a range-r neighborhood and whose value is within a distance d. The Mean Shift takes usually 3 inputs: 1) A distance function for measuring distances between pixels. Usually the Euclidean distance, but any other well defined distance function could be used. The Manhattan Distance is another useful choice sometimes. 2) A radius. All pixels within this radius (measured according the above distance) will be accounted for the calculation. 3) A value difference. From all pixels inside radius r, we will take only those whose values are within this difference for calculating the mean [4].

3. Watershed (plan to be implemented during GSoC 2016)

Watershed segmentation classifies pixels into regions using gradient descent on image features and analysis of weak points along region boundaries. Imagine water raining onto a landscape topology and flowing with gravity to collect in low basins. The size of those basins will grow with increasing amounts of precipitation until they spill into one another, causing small basins to merge together into larger basins. Catchment basins are formed by using local geometric structure to associate points in the image domain with local extrema in some feature measurement such as curvature or gradient magnitude. This technique is less sensitive to user-defined thresholds than classic region growing methods, and may be better suited for fusing different types of features from different data sets. The watersheds technique is also more flexible in that it does not produce a single image segmentation, but rather a hierarchy of segmentations from which a single region or set of regions can be extracted a-priori, using a threshold, or interactively, with the help of a graphical user interface.

The strategy of watershed segmentation is to treat an image f as a height function, i.e., the surface formed by graphing f as a function of its independent parameters, $x \in U$. The image f is often not the original input data, but is derived from that data through some filtering, graded (or fuzzy) feature extraction, or fusion of feature maps from different sources. The assumption is that higher values of f (or − f) indicate the presence of boundaries in the original data. Watersheds may therefore be considered as a final or intermediate step in a hybrid segmentation method, where the initial segmentation is the generation of the edge feature map [1].

Main Goal

Implement more image segmentation methods to extend the available i.segment for image processing in GRASS. As the general logistics of the i.segment module is in place, adding mean-shift and watershed segmentation algorithms should be possible. The core of the GSoC project thus is to add a series of these algorithms. In addition, the current implementation only uses distance within the multidimensional space of all input bands as the criteria whether to merge segments or not. Adding shape as an additional merge criteria will be helpful. If time permits, this feature of additional shape criteria will be implemented.

Timeline

Preparation: Discuss with Mentors. Gather ideas from the community. Feature requests, image segmentation literature, and any other ideas and suggestions.

16 – 21 May week 0: Setup coding environmental, get familiar with programming manual, test through existing code.

23 -- 28 May week 1: Start coding, develop pseudo code to outline the work

30 May -- 4 June week 2: implement mean-shift image segmentation algorithm

6 -- 11 June week 3: Validation and debugging mean-shift algorithm

13 -- 18 June week 4: implement watershed image segmentation algorithm

20 -- 26 June Week 5: Validation and debugging watershed algorithm

27 June Mid-term evaluation: Evaluate the existing program, determine the plan for the remaining 3-4 weeks.

28 June -- 2 July Week 6: based on the evaluation of the mid-term, test and ensure a solid existing program.

4 -- 9 July Week 7: Implement split-window image segmentation algorithm

11 -- 16 July Week 8: Validation and debugging split-window algorithm

18 -- 23 July: Further refine tests and documentation for the whole project.

25 July – 13 August Week 9-11: Improving the main algorithm, if time permits, adding shape as an additional merge criteria,

15 August - 23 August Final week: Tidy code, write tests, improve documentation and submit code sample.

Reference

[1] OTB software guide: https://www.orfeo-toolbox.org/packages/OTBSoftwareGuide.pdf

[2] GRASS Manuals: https://grass.osgeo.org/grass71/manuals/i.segment.html

[3] Comaniciu, D., & Meer, P. (2002). Mean shift: a robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(5), 1–37.

[4] http://stackoverflow.com/questions/4831813/image-segmentation-using-mean-shift-explained