# gPb Contour Detection for UAV Data

Author: Sophie Crommelinck, PhD Candidate University of Twente, ITC (s.crommelinck@utwente.nl)
Supervisors: George Vosselman, Mila Koeva, Michael Ying Yang

## Summary

This document describes how to use the pipeline of gPb-owt-ucm on UAV data as described in the paper: [Crommelinck, S.; Bennett, R.; Gerke, M.; Yang, M.; Vosselman, G. Contour detection for UAV-based cadastral mapping. *Remote Sensing* **2017**, *9*, 1-13.]

### 1.1. UAV Data

#### *1.1.1. In Theory*

The following example implementation of the delineation tool is based on data from Amtsvenn, Germany (latitude/longitude: 52.17335/6.92865). The data was captured with indirect georeferencing, i.e., Ground Control Points (GCPs) were distributed within the field and measured with a Global Navigation Satellite System (GNSS). The orthoimage captures an extent of 1000 x 1000 m and has a GSD of 0.05 m. It was captured with a fixed-wing UAV (model: GerMAP G180, camera: Ricoh GR) flying with a forward overlap of 80% and a sideward overlap of 65%. The orthoimages were generated with Pix4DMapper.

#### *1.1.2. In Practice*

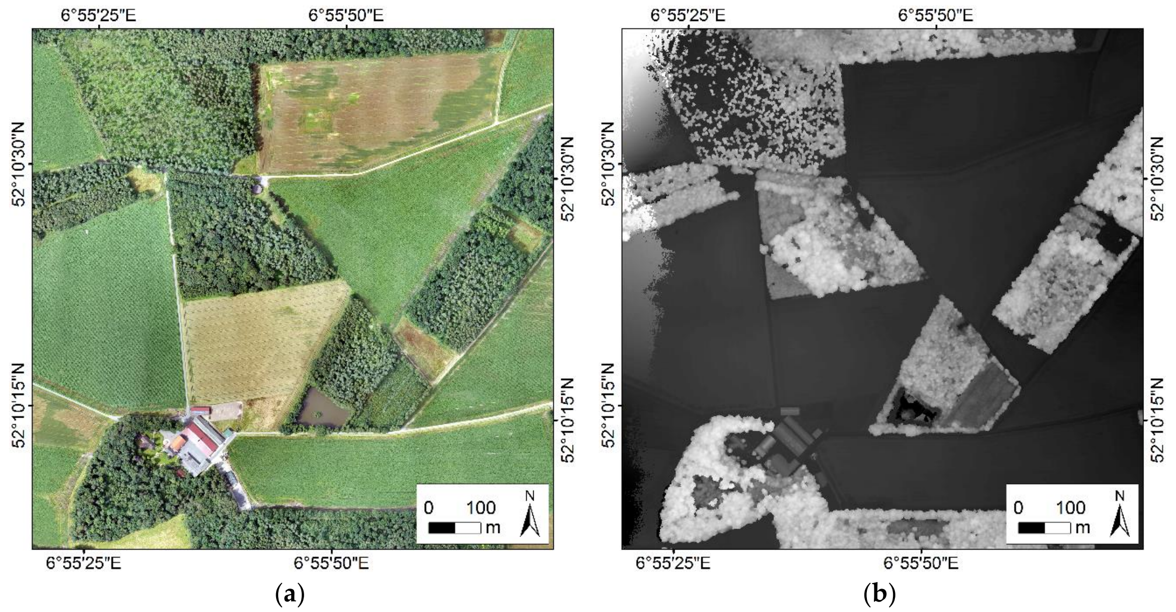
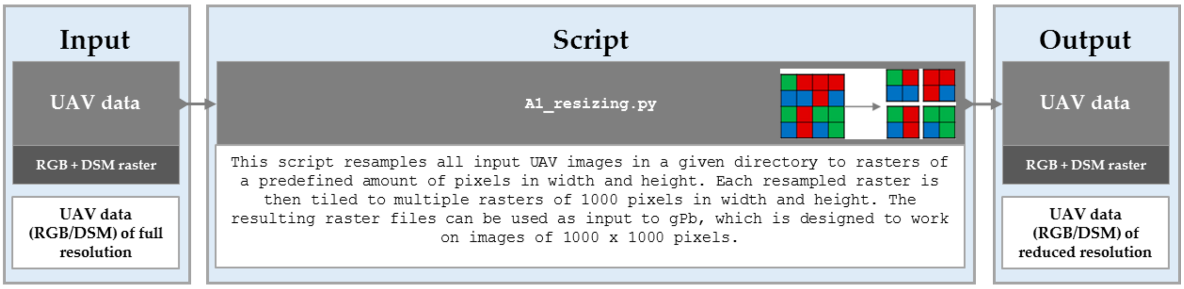



**Figure 2.** UAV data of Amtsvenn, Germany showing an extent of 1000 x 1000 m with a GSD of 0.05 m as RGB (a) and DSM (b) orthoimages.

As input for the subsequent workflow step (gPb contour detection) both the RGB and the DSM orthoimages are downsampled to <=1000 x 1000 pixels. This is done with a PyQGIS script.




## 1.2. gPb Contour Detection

### 1.2.1. *In Theory*

Contour detection refers to finding closed boundaries between objects or segments. Globalized probability of boundary (gPb) contour detection refers to the processing pipeline of gPb-owt-ucm, which is visualized in Figure 3, explained in the following and based on [1]. The pipeline originates from computer vision and aims to find closed boundaries between objects or segments within an image. This is achieved through combining edge detection and hierarchical image segmentation, while integrating image information on texture, color and brightness on both a local and a global scale.

In a first step, oriented gradient operators for brightness, color and texture are calculated on two halves of differently scaled discs. The cues are merged based on a logistic regression classifier resulting in a posterior probability of a boundary, i.e., an edge strength per pixel. This local image information is combined trough learning techniques with global image information that is obtained through spectral clustering. The learning steps are trained on natural images from the 'Berkeley Segmentation Dataset and Benchmark' [2]. By considering image information on different scales, relevant boundaries are verified, while irrelevant ones, e.g., in textured regions, are eliminated.

In the second step, initial regions are formed from the oriented contour signal provided by a contour detector through oriented watershed transformation (owt). Subsequently, a hierarchical segmentation is performed through weighting each boundary and their agglomerative clustering to create an ultrametric contour map (ucm) that defines the hierarchical segmentation.

The overall results consists of (i) a contour map, in which each pixel is assigned a probability of being a boundary pixel, and (ii) a binary boundary map containing closed contours, in which each pixel is labelled as 'boundary' or 'no boundary'. The approach has shown to be applicable to UAV orthoimages for an initial localization of candidate object boundaries. Due to the global optimization, UAV orthoimages of extents larger than 1000 x 1000 pixels need to be reduced in resolution, in order to be processed with the original implementation. These performance issues, leading to a reduced localization quality [3] are addressed in [4]. To improve the localization quality of initially detected candidate boundaries is the aim of the proceeding workflow steps.

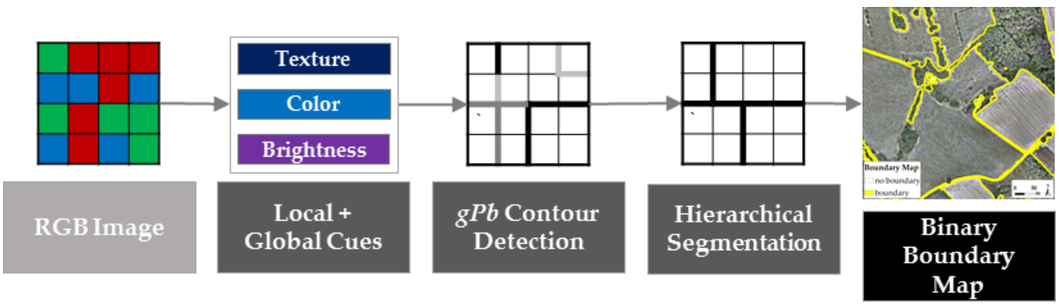
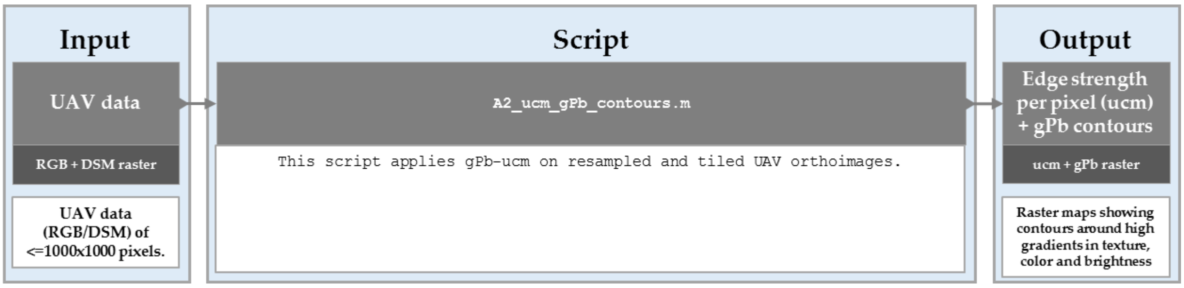
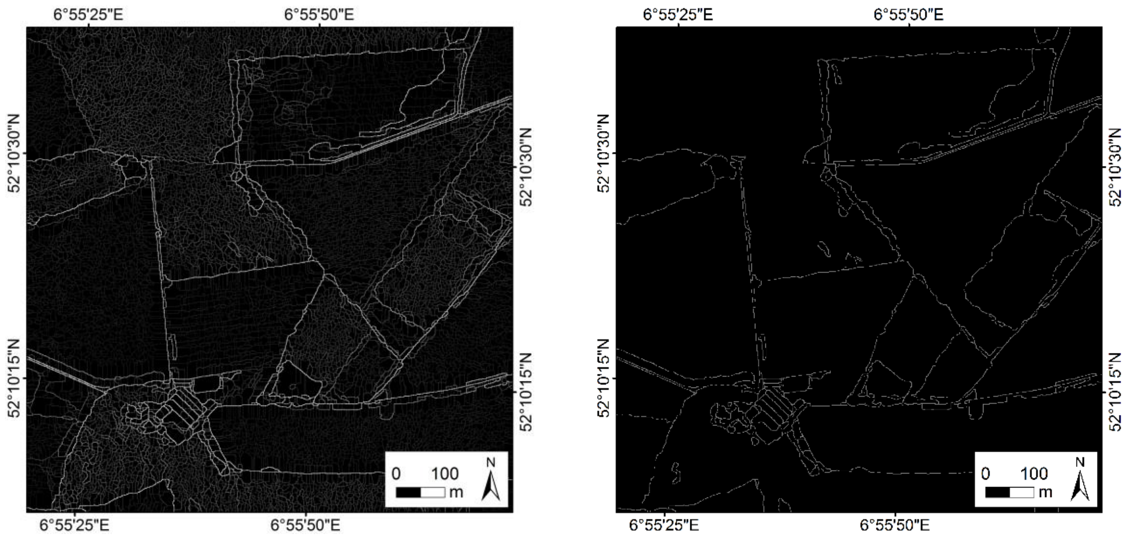



**Figure 3.** Processing pipeline of globalized probability of boundary (gPb) contour detection and hierarchical image segmentation resulting in a binary boundary map containing closed segment boundaries. The approach is described in [1].

### *1.2.2. In Practice*

The pipeline of gPb-owt-ucm is implemented in Matlab [5]. The source code is publicly available in [6]. The script requires a list of precompiled `.mex` files in a `lib` directory and runs on Linux. That `lib` directory contains a `geotiffwrapper.m` script, in which the variable `A.ProjectedCSTypeGeoKey` needs to be set to the EPSG code of the input image. This ensures that the output `.tif` files have the same georeferencing as the input images. All files in the `lib` directory are precompiled `.mex` files, which are called via the main script in Matlab (`gPb_ucm_final.m`). The images to be processed (`.tif` and their `.tfw` files) need to be copied to the `data` directory. An example output of the gPb-owt-ucm pipeline is shown in Figure 4.

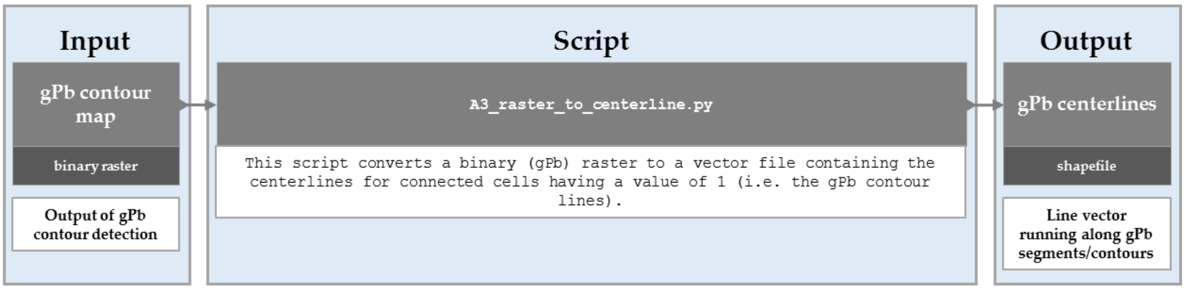



(**a**) (**b**)

**Figure 4.** The result of the gPb-owt-ucm pipeline applied to the Amtsvenn UAV data (**Figure 2**). (**a**) shows the ucm map, in which each pixel is assigned a specific probability of being a boundary. (**b**) shows the gPb map, in which all edges having a probability above a certain threshold are kept for an hierarchical image segmentation, resulting in a binary raster map of closed contours.

The gPb raster map is transferred to a vector shapefile by keeping the centerline for each gPb contour. This is done with a PyQGIS script.



## References


1. Arbelaez, P.; Maire, M.; Fowlkes, C.; Malik, J. Contour detection and hierarchical image segmentation. *Pattern Analysis and Machine Intelligence* **2011**, *33*, 898-916.
2. Arbeláez, P.; Fowlkes, C.; Martin, D. Berkeley segmentation dataset and benchmark. Available online: https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/ (accessed on 10 November 2016).
3. Crommelinck, S.; Bennett, R.; Gerke, M.; Yang, M.; Vosselman, G. Contour detection for UAV-based cadastral mapping. *Remote Sensing* **2017**, *9*, 1-13.
4. Donoser, M.; Schmalstieg, D. In *Discrete-continuous gradient orientation estimation for faster image segmentation*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014; pp 3158-3165.
5. MathWorks. Available online: www.mathworks.com (accessed on 10 June 2016).
6. Arbeláez, P.; Maire, M.; Fowlkes, C.; Malik, J. Contour detection and image segmentation resources. Available online: https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html (accessed on 10 November 2016).