From Satellite Images to Vector Representations
Onur Tasar, Yuliya Tarabalka, Pierre Alliez, Alain Giros, Sébastien Clerc

To cite this version:
Onur Tasar, Yuliya Tarabalka, Pierre Alliez, Alain Giros, Sébastien Clerc. From Satellite Images to Vector Representations. JC2 - 14èmes Journées CNES Jeunes Chercheurs, Nov 2019, Toulouse, France. hal-02367545v2

HAL Id: hal-02367545
https://inria.hal.science/hal-02367545v2
Submitted on 4 Dec 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
From Satellite Images to Vector Representations

Omar Tasar1 (omur.tasar@inria.fr)
Yuliya Tarabalka1, Pierre Alliez2, Alain Giros2, Sébastien Clerc3
INRIA Sophia Antipolis TITANE Team4, CNES2, ACRI-ST3

Introduction
One of the most popular challenges in the field of remote sensing is to generate vector representations from satellite images to be included into various Geographic Information System (GIS) applications. One way to generate the aforementioned vector representations is to split this task into two consecutive sub-tasks, where the first one consists in generating pixel-wise maps in raster format using advanced machine learning techniques, and the second one aims at vectorizing the obtained raster map by applying computational geometry methods. We propose novel methods for both sub-tasks.

Overview

Incremental Learning: Learning new classes continually without forgetting the already learned classes.

Domain Adaptation: Generating high quality maps even when there exists a large difference between spectral bands of training and test images.

Mesh Approximation: Vectorizing the raster map using a triangle mesh with edge flip, half edge collapse, and vertex relocation operators.

Conclusions

Incremental Learning: To the best of our knowledge, we propose the first work on incremental learning for semantic segmentation. Domain Adaptation: The proposed approach is architecturally simple but powerful. We achieve a better performance than the existing approaches in a much shorter period of time. Mesh Approximation: The approach yields a better accuracy than the simplification algorithms in common GIS softwares, although our method generates significantly less number of vertices.

References

Domain Adaptation

Bad Ischl Fake Bad Ischl Villach Fake Villach

• When there is a large shift between the spectral bands of training and test images, the existing machine learning models fail to segment the test data.
• We generate fake training images that are semantically exactly the same as the original training images, but whose spectral distribution as close as possible to test images by ColorMapGAN [2].
• We then use the fake training data to fine-tune the model trained on the original training data.

Mesh Approximation

• We transform the initial mesh to the optimal mesh by minimizing an objective function with certain operators [4].
• Our objective function balances between fidelity to the classification map in ℓ1 norm sense, right angle regularity for polygonized buildings, and the final mesh complexity.
• The operators: edge flip, half edge collapse, vertex relocation.

Training data 1
Training data 2
Training data 3
Test Image
Predicted Map
Initial network
Updated network1
Updated network2
Updated network2

• When new classes are added to the already trained model, the current approaches suffer from catastrophic forgetting (significant performance drop for the previous classes).
• The training phase of our framework [1, 3] has two stages: adaptation and remembering.
• During the adaptation, we use new ground-truth to learn new classes on the current data, and we use the output of the previously trained network to learn old classes from the current data.
• In the remembering phase, we store only a small portion of the previous training data, and we use it to remind the network the learned information for old classes from the previous data.