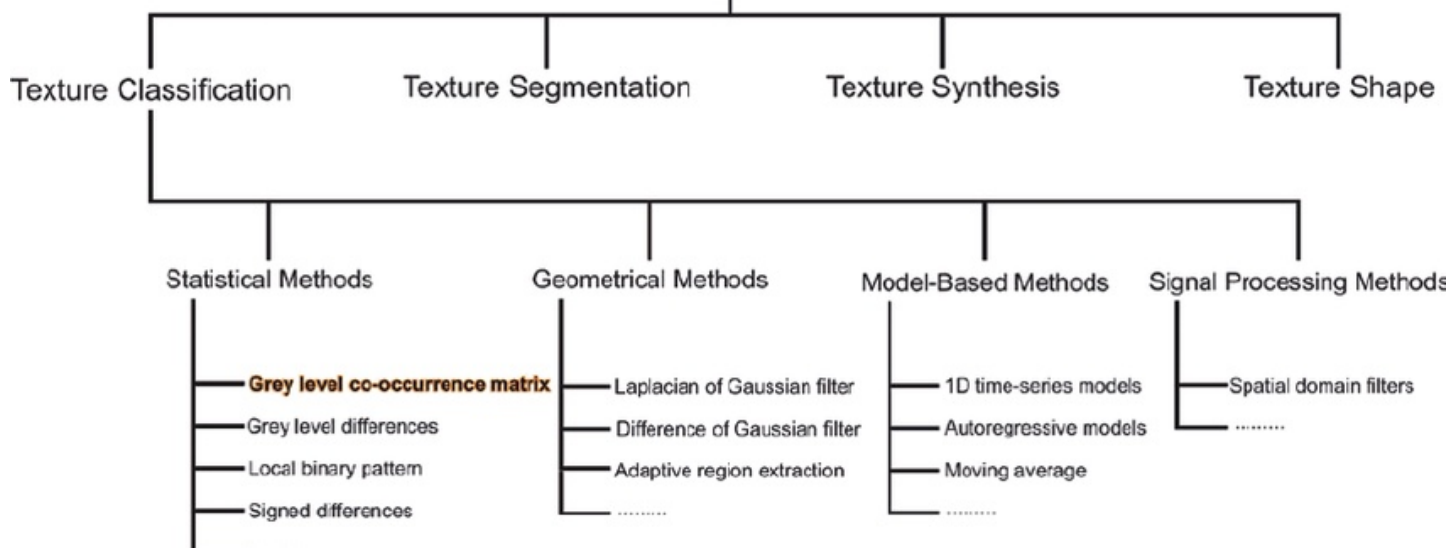


Texture Analysis



使用灰度共生矩阵 GLCM 提取影像纹理信息



最傻的熊...

Talk is cheap

11 人赞了该文章

撰文/熊氏阿回

灰度共生矩阵

Google Earth Engine 不仅解决一些常见的地学遥感图像处理问题，也可用于特定的图像分析，比如纹理特征的提取和识别。

纹理特征是一种不依赖于颜色或亮度而反映图像中同质现象的视觉特征，包含了物体表面结构组织排列的重要信息以及它们与周围环境的联系。纹理识别不仅可以应用在遥感图像上，对医学、天文、显微图像等都具有应用价值。

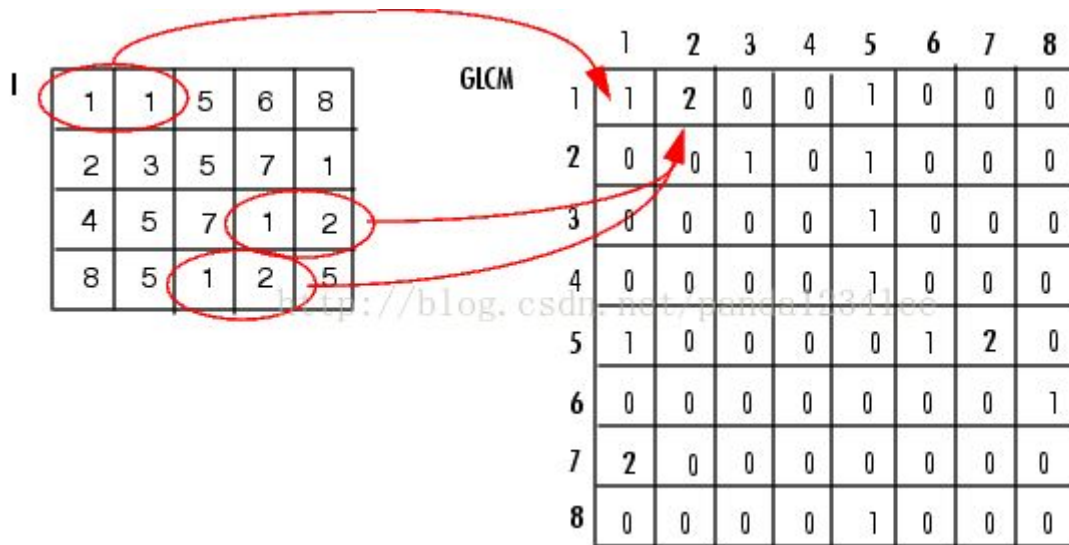
由于纹理是由灰度分布在空间位置上反复出现而形成的，因而在图像空间中相隔某距离的两像素之间会存在一定的灰度关系，即图像中灰度的空间相关特性。

灰度共生矩阵就是一种通过研究灰度的空间相关特性来描述纹理的常用方法。1973 年由 Haralick 提出，灰度共生矩阵是涉及像素距离和角度的矩阵函数，它通过计算图像中一定距离和一定方向的两点灰度之间的相关性，来反映图像在方向、间隔、变化幅度及快慢上的综合信息。

▲ 赞同 11 ▼

● 添加

水平相邻。



灰度共生矩阵的特征量

1) 角二阶矩 (Angular Second Moment, ASM)

公式: $ASM = \sum(p(i,j))^2$, 其中 $p(i,j)$ 表示归一后的灰度共生矩阵

意义: 角二阶矩是图像灰度分布均匀程度和纹理粗细的一个度量, 当图像纹理较细致、灰度分布均匀时, 能量值较大, 反之, 较小。

2) 熵 (Entropy, ENT)

公式: $ENT = -\sum(p(i,j) \cdot \log(p(i,j)))$

意义: 描述图像具有的信息量的度量, 表明图像的复杂程度, 当复杂程度高时, 熵值较大, 反之则较小。

3) 反差分矩阵 (Inverse Differential Moment, IDM)

公式: $IDM = \sum(p(i,j) / (1 + (i-j)^2))$

意义: 反映了纹理的清晰程度和规则程度, 纹理清晰、规律性较强、易于描述的, 值较大; 杂乱无章的, 难于描述的, 值较小。

知乎

首发于
地球云计算

GEE 中已经提供了 GLCM 的函数接口，可以一次生成所有波段的 GLCM 纹理，只需要按照需要把想要的部分增强显示出来：

```
var region = ee.Geometry.Polygon([[[-97.7237319946289, 25.425601933964312],
                                   [-97.46932983398438, 25.428702591345065],
                                   [-97.46143341064453, 25.578369229079655],
                                   [-97.72956848144531, 25.57589179857278]]]);

Map.addLayer(region);
Map.setCenter(-97.46143341064453, 25.578369229079655, 14)
var cloudMask = function(image) {
  var bqa = image.select('BQA')
  .eq([61440,59424,57344,56320,53248,39936,36896,36864,61440, 59424, 57344, 31744, 286
  return image.updateMask(bqa.not());
};

var NDVI = function(img) {
  var ndvi = img.normalizedDifference(['B5', 'B4']);
  return ndvi;};

var l8 = ee.ImageCollection('LANDSAT/LC8_L1T_TOA').filterDate('2016-01-01', '2016-12-31')
  .select('B5', 'B4', 'BQA').map(cloudMask);

var ndvi = l8.map(NDVI).median().clip(region);
Map.addLayer(ndvi,{min:-1,max:1});

var input = ndvi.add(127.5).multiply(127.5).toUint16()
Map.addLayer(input, {min:16291,max:16300}, 'GLCM input');

print(input.reduceRegion(ee.Reducer.stdDev(), region, 30).get('nd'))

var glcm = input.glcmTexture();
Map.addLayer(glcm.select('nd_avg',{min:32582,max:32591}, 'GLCM output (SAVG)'));
Map.addLayer(glcm.select('nd_var',{min:0,max:20}, 'GLCM output (VAR)'));
```

▲ 赞同 11 ▼

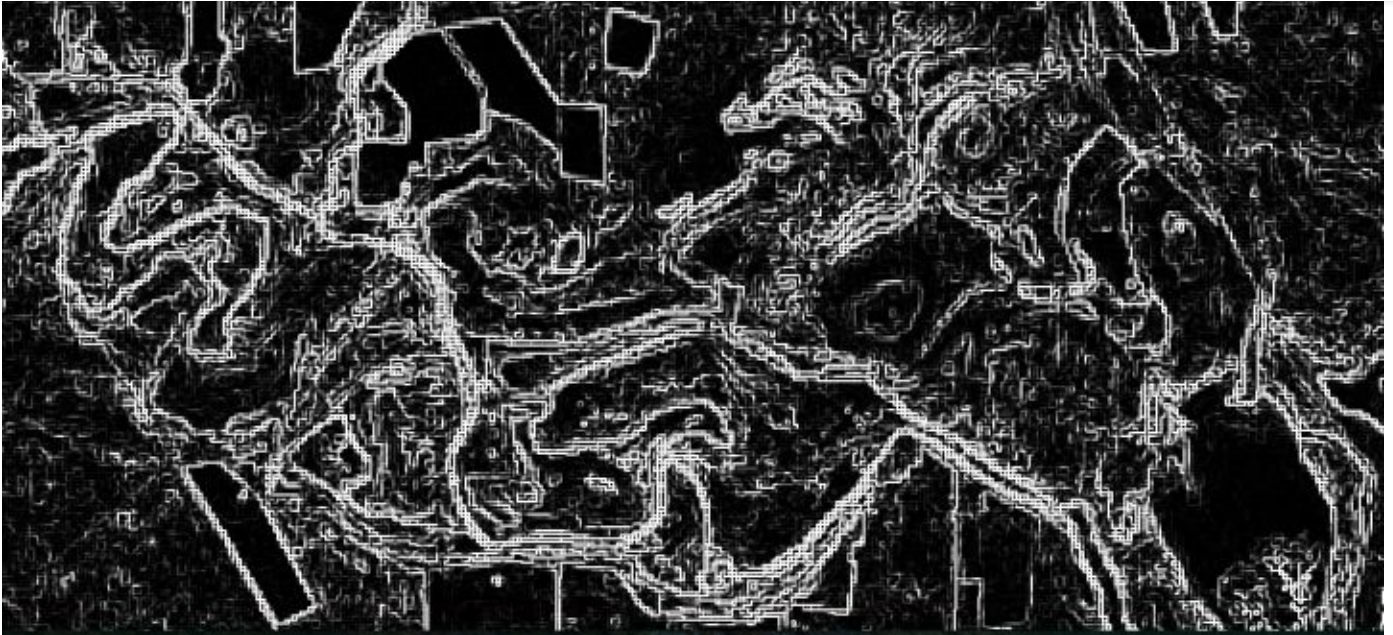
● 添加



待分析影像



纹理信息一：区分有/无植被



纹理信息二：有/无植被地块的边界信息

参考资料

1. Mathworks: [Texture Analysis Using the Gray-Level Co-Occurrence Matrix \(GLCM\)](#)
2. Wikipedia: [Co-occurrence matrix](#)
3. [图像的灰度共生矩阵 - CSDN博客](#)
4. [灰度共生矩阵及相关特征值的计算--opencv - yanxiaopan的博客 - CSDN博客](#)
5. [OpenCV22（灰度共生矩阵/灰度共生矩阵） - CSDN博客](#)

发布于 2017-10-01

云计算

图像处理

纹理

文章被以下专栏收录



地球云计算

进入专栏

▲ 赞同 11 ▼

● 添加

数字图像处理

第1、2章 绪论、图象处理的基本概念。第3、4章 空域和频域变换第5、8章 图象增强和图象复原。第6章 图象编码第7、9章 图象分割和形态学。第1章 绪论 1.1 从图象到图象工程图象和数字图象图...

大师兄

图像视觉领
续更新)

本文转自开源
Residual Net
Learning for
KaimingHe/c
networksIde
Deep Residu
极视角

还没有评论

评论由作者筛选后显示

