Towards a Method for Automatic Identification of Trees in Aerial Images

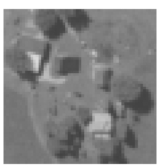
BING WANG[1], ITALO O. MATIAS[2], AND LUIZ M. G. GONÇALVES[1 2]

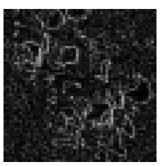
[1]Department of Computer Science
University of Massachusetts (UMASS), Amherst MA 01003 USA
(bing,lmarcos)@cs.umass.edu
[2]Laboratório de Computação Gráfica - COPPE
Universidade Federal do Rio de Janeiro (UFRJ), CP 68511 Rio de Janeiro RJ 21945-970
(itom)@lcg.ufrj.br

Abstract. This work describes ongoing research towards finding a methodology to extract information about trees from aerial images. Examples of output are the approximate number of trees, the diameter and area of a tree, and the total green area. For regions with sparse trees, a method based on segmentation and border extraction is described. It works reasonable well segregating trees from the other components (buildings, etc). For dense tree areas, two feature based methods, one using simple $template\text{-}matching$ and the other one using a Backpropagation $neural\text{-}network$ classifier, are described. Both methods work reasonable well if a good sample of trees is provided.

Method for images with sparse trees

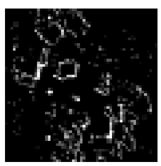
Aerial image with some sparse trees and houses.





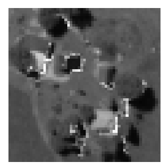
Gray scale image obtained from edge detection.

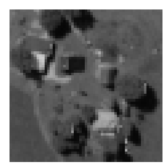
Binary image after applying a ”close” operation.



Band thresholding over the original image.

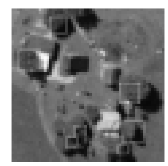
Resulting image by not using color information.





Resulting image by using the color information.

Detected tree areas are marked with white squares.



Method for images with dense trees

Original “mosaic” image.

↓

Samples of trees ($l = 1, ..., 7$).

↓

$$G^{(0)} = ce^{ar^2}$$

$$G^{(1)} = 2ace^{ar^2}(x+y)$$

$$G^{(2)} = 4ace^{ar^2}(a(r^2+x+y)+1) \rightarrow$$

where $r^2 = x^2 + y^2$, $a = \frac{-1}{2s^2}$, $c = \frac{1}{s\sqrt{2\pi}}$, and $s = 1.7$.
Gaussian masks for feature extraction.

$$\mu^{(k)}_{l,i,j} = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\frac{G^{(k)}_{l,i+m,j+n}}{G^{(k)}{}_{Max}}$$

$${\sigma^{(k)}_{l,i,j}}^2 =$$

$$\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\left(\frac{G^{(k)}_{l,i+m,j+n}}{G^{(k)}{}_{Max}} - \mu^{(k)}_{l,i,j}\right)^2$$

Invariant moments from Gaussian.

↑

$$o_j = (1 + e^{-\sum_{i=0}^{A}\omega_{i,j}x_i})^{-1}$$

$$\Delta\omega_{i,j}(t+1) = \epsilon\delta_j o_i + \alpha\Delta\omega_{i,j}(t),$$

$$\delta_j = \begin{cases} o_j(1-o_j)(y_j - o_j) \\ o_j(1-o_j)\sum_{k=1}^{B}\delta_k\omega_{jk} \end{cases}$$

Backprop $Neural\text{-}Network$ (BP).

$\rightarrow$

Resulting images. Top uses BP (140 trees found). Botton uses $template\text{-}matching$ (130 trees found).

↑

$\rightarrow$

$$Min_l\left\{\sum_{i=1}^{I}\sum_{j=1}^{J}\left(\mu^{(k)}_{i,j} - \mu^{(k)}_{l,i,j}\right)^2 + \sum_{i=1}^{I}\sum_{j=1}^{J}\left(\sigma^{(k)}_{i,j} - \sigma^{(k)}_{l,i,j}\right)^2\right\} < \epsilon$$

Minimum $template\text{-}matching$.