

## Image matching using Position Trees

Pixels in an image are sequentially presented and aligned, like letters in a text. In addition, each pixel of an image can be digitized as r/g/b values and we can label each edge of Position Tree with a non-negative integer instead of a letter.

1. Convert two images to 3 r/g/b value planes.
  - a) Load two images with Matlab.  
Let us call one of them as text and the other one as pattern.
  - b) Convert those two images into 3-dimensional double rgb arrays.  
Those arrays are composed of 3 double planes, r-value plane, g-value plane, and b-value plane.
  - c) Convert those double r/g/b planes of text and pattern into non-negative integer planes by multiplying 100 to those r/g/b planes and dropping decimal part of each value in those planes.
2. Build 3 Position Trees based on r/g/b planes of text.
  - a) Expand each of r/g/b planes of text (big enough for corresponding r/g/b planes of pattern to be merged, even without actual overlapping letters) with cells filled with the letter which can be matched with any regular letter including itself.
  - b) Convert each plane of text into a one dimensional non-negative integer array by combining all rows of each plane of text into one dimensional non-negative integer array.
  - c) Build three position trees of three planes (r/g/b planes) of text from corresponding one dimensional non-negative integer arrays made in b).
3. Find matching positions in the text where pattern can be properly merged using Position Trees.
  - a) Search down each of r/g/b position trees and figure out all positions where each row of the corresponding r/g/b planes of pattern appears and vote on the following position:  
**Voted** position =  $p - (P\_i \times T\_c)$ , where  
P: Position where each row of corresponding plane of pattern appears  
P\_i: Index of row of corresponding plane of pattern  
T\_c: The # of columns of expanded planes of text
  - b) Among all positions with full votes (3 x the s# of rows in planes of pattern),  
find a position which will give us (1)minimized size or (2)maximized number of overlapping letters.
4. Merge the text and pattern.
  - a) Merge 3-dimensional double arrays of text and pattern at the position found in 3-b).
  - b) Plot the merged 3-dimensional double array using Matlab.

Please notice that

- 1) For improved performance, we can easily substitute Compressed Position Trees or Suffix Tree for Original Position Trees
- 2) This image matching algorithm is not quite practice since it is comparing pixels without any object detection.