**CSC 545/645 Computer Speech, Music and Images**

**Exercise No. 12, April 19, 2015**

**Component labelling—the recursive grass fire algorithm**

**Goals**

Learn to label blobs in a binary image

**Background**

It’s one thing to create a binary image in which the foreground objects are one color and the background objects another. While the objects may be easy to see with the human eye, a computer vision system has to systematically differentiate the foreground objects from the background. This is called *component labeling.* A common algorithm for component labeling is the recursive grassfire algorithm (also less colorfully known as the floodfill algorithm). The idea is simple: for each pixel in the image, if that pixel is a foreground image that has not been labelled, label it and assign the same label to all connected foreground pixels. The idea is that labels spread like a grass fire from one pixel to its neighbors. An object is defined as a set of connected foreground pixels and each pixel will receive a unique label that will apply to all its pixels. In practice, each object is often assigned its own color.

**Procedure**

Write a Processing program to carry out the recursive grass fire algorithm. Start with Ex12\_ComponentLabelling, stored in the Exercises folder on the class server. The algorithm is formally defined as follows:

void grassFire(img) {

for each pixel x, y in img

create a unique color c

label(img, x, y, )

}

void label(img, x, y, c) {

If pixel at img(x, y) is foreground and is unlabelled

Set pixel’s color to c

label(img, x-1, y, c) //Check boundaries

label(img, x+1, y, c)

label(img, x, y-1, c)

label(img, x, y+1, c)

}