

# A Pixel Matching Problem

---

*Source:* <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2005-11/msg02258.html>

---

- *From:* Roedy Green <[my\\_email\\_is\\_posted\\_on\\_my\\_website@xxxxxxxxxxxxxxxxx](mailto:my_email_is_posted_on_my_website@xxxxxxxxxxxxxxxxx)>
  - *Date:* Sat, 19 Nov 2005 21:24:43 GMT
- 

this problem is like a miniature OCR problem.

Let's say you have some images that contain text, or screen snapshots obtained by the Robot class. How can you extract the text from them?

You have the advantage that you probably have access to the fonts used to create the text.

How might you go about creating an OCR for images? I was thinking of writing this up as a student project.

The advantage you have over true OCR is all matches will be exact.

The complications are a variety of colours for foreground/background, antialiasing and painting over multicoloured backgrounds. For JPGs you may not have the fonts used. The edges of the text may be tweaked in various ways, e.g. 3D, blurred, warped.

I thought you might proceed like this:

1. find rectangular regions containing only two colours.
2. draw an e in 10 font sizes for each font as your search templates.
3. look for a hole the right shape. When you find one, check all the e's you have with that size/shape hole to see if you have a match on the entire letter, check the whole rectangle.
4. if you have a match, calculate the baseline and starting point. now draw a template ea in that same font and compare. If it fails try eb etc. Work your way both left and right pulling that line.

You might construct a hashMap indexed by a digest of the glyph so you can more rapidly check for matches. Your digest algorithm might trim the glyph top/bottom/left/right so you don't need the stringWidth information by actually drawing the character pair.

5. You carve the rectangle out of the bigger one, and break the remaining into rectangles.

## A Pixel Matching Problem

6. repeat until there are no more rectangles.
7. export the text each labelled with x.y where it was found.
8. In another program allow the user to highlight text, e.g. a column or box to determine the linear order of the text desired.

Potential uses for such software include:

1. capturing filenames, error messages, crash locations that were displayed in a non-cut/pasteable way.
2. by people trying to defeat my email munger. See <http://mindprod.com/applets/masker.html>
3. by blind people extracting textual information from images.
4. To allow you to copy from any Swing Component.
5. to extract information from a screen snapshot without having to retype it.

—  
Canadian Mind Products, Roedy Green.  
<http://mindprod.com> Java custom programming, consulting and coaching.

- 
- Prev by Date: ***Re: Best Java ever***
  - Next by Date: ***Re: Best Java ever***
  - Previous by thread: ***hi all***
  - Next by thread: ***Re: Urgent Requirement***
  - Index(es):
    - ◆ ***Date***
    - ◆ ***Thread***