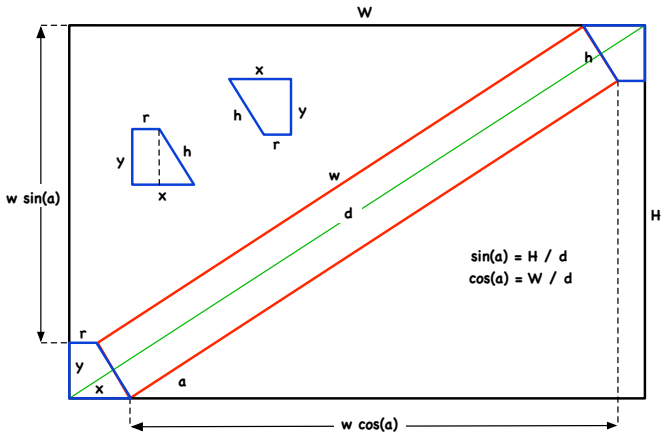
Drawing a string along the diagonal of a given rectangle: the 'centered' method

Wagner L. Truppel

August 26, 2006

Say that your view rectangle is W pixels wide and H pixels tall and you want to draw a slanted string that's 'centered' on the view's diagonal. You need to find the rectangle inside which to draw the string. In the figure below, that rectangle is drawn in red, with width w and height h, and the view's diagonal is drawn in green (with length d). Note the two identical trapezoids at the corners (drawn in blue, and reproduced separately for clarity), with the given dimensions. Also, a is the angle by which the view's diagonal is inclined, which is also the angle by which the string rectangle is inclined.



What is known? We know W, H and, presumably, the actual string to be drawn.

What's needed? We want to determine w, h, x, a, and an appropriate font size with which to draw the given string. w and h determine the string rectangle, and x determines the anchor point around which the string rectangle must be rotated (counterclockwise, by the angle a). The two other variables appearing on the picture (y and r) are auxiliary variables.

The general strategy to solve the problem outlined above is to choose a font size and use it to determine w, from which we'll obtain all other variables. We then compare the computed value of h with the height of the tallest glyph of the string, for the chosen font size: if h is smaller, the tallest glyph is too tall, and we must decrease the font size. If, instead, h is larger than the height of the tallest glyph, we can attempt to increase the font size. Either way, w is changed (along with the values of all the other variables) and we must repeat the procedure just described until we converge on the optimal font size: the largest font size for which the string rectangle's width w is larger than, but as close as possible to, the actual string width and the string rectangle's height h is larger than, but as close as possible to, the height of the tallest glyph. This increasing and decreasing of the font size can be done most efficiently through a simple binary search.

So, in the following, we'll assume that w is also a known quantity, from which all other variables must be computed. What equations can we use to determine them? Note the following equalities, which result from considering various similar triangles:

$$\begin{aligned}
\sin(a) &= \frac{H-y}{w} = \frac{x-r}{h} = \frac{H}{d}\,, \\
\cos(a) &= \frac{W-x-r}{w} = \frac{y}{h} = \frac{W}{d}\,, \\
\tan(a) &= \frac{H-y}{W-x-r} = \frac{x-r}{y} = \frac{H}{W}\,,
\end{aligned}$$

where $d = \sqrt{W^2+H^2}$. From the last equality of the second line, we can determine y in terms of h and known quantities:

$$y = \frac{h}{d}\,W\,.$$

Then, combining this result with

$$\frac{H-y}{w} = \frac{H}{d}$$

from the first line, it follows:

$$h = \frac{H}{W}\,(d-w)\,,$$

which allows us to rewrite y as

$$y = \frac{H}{d}\,(d-w)\,.$$

Next, we use

$$\frac{W-x-r}{w} = \frac{W}{d} \quad \text{and} \quad \frac{x-r}{h} = \frac{H}{d}$$

to obtain

$$W - x - r = \frac{W}{d}\, w \quad \text{and} \quad x - r = \frac{H}{d}\, h\,.$$

By adding and subtracting these two equations, we can isolate r and x, respectively:

$$W - 2r = \frac{W}{d}\, w + \frac{H}{d}\, h \quad \text{and} \quad W - 2x = \frac{W}{d}\, w - \frac{H}{d}\, h\,,$$

resulting in

$$2r = W - \frac{W}{d}\, w - \frac{H}{d}\, h \quad \text{and} \quad 2x = W - \frac{W}{d}\, w + \frac{H}{d}\, h\,.$$

Inserting the result derived for h, it then follows:

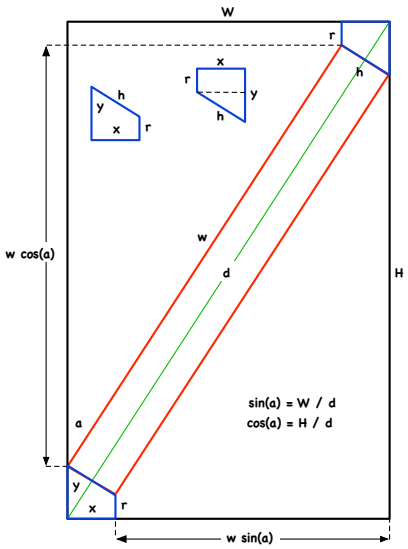
$$r = \frac{(W^2 - H^2)}{2dW}\,(d - w) \quad \text{and} \quad x = \frac{d}{2W}\,(d - w)\,.$$

As promised, we've determined the values of all variables in terms of w, W, and H:

$$h = \frac{H}{W}\,(d - w)\,, \qquad x = \frac{d}{2W}\,(d - w)\,,$$

$$y = \frac{H}{d}\,(d - w)\,, \qquad r = \frac{(W^2 - H^2)}{2dW}\,(d - w)\,.$$

The derivation above assumes $W \geq H$, that is, that the view rectangle is longer than taller, or a square. If that's not the case, the equations need to be changed, as follows.



From the picture above, we find:

$$\begin{aligned}
\sin(a) &= \frac{W-x}{w} = \frac{y-r}{h} = \frac{W}{d}\,, \\
\cos(a) &= \frac{H-y-r}{w} = \frac{x}{h} = \frac{H}{d}\,, \\
\tan(a) &= \frac{W-x}{H-y-r} = \frac{y-r}{x} = \frac{W}{H}\,.
\end{aligned}$$

Comparing to the previous case, we see that the new equalities are obtained from the old ones by simultaneously exchanging x with y and W with H. Thus, when $W \le H$, we find:

$$\begin{aligned}
h &= \frac{W}{H}\,(d-w)\,, \qquad & y &= \frac{d}{2H}\,(d-w)\,, \\
x &= \frac{W}{d}\,(d-w)\,, \qquad & r &= \frac{(H^2-W^2)}{2dH}\,(d-w)\,.
\end{aligned}$$

Collecting all the results in one place, we then have:

- $H \le W$:

$$\begin{aligned}
h &= \frac{H}{W}\,(d-w)\,, \qquad & x &= \frac{d}{2W}\,(d-w)\,, \\
y &= \frac{H}{d}\,(d-w)\,, \qquad & r &= \frac{(W^2-H^2)}{2dW}\,(d-w)\,.
\end{aligned}$$

- $W \le H$:

$$\begin{aligned}
h &= \frac{W}{H}\,(d-w)\,, \qquad & x &= \frac{W}{d}\,(d-w)\,, \\
y &= \frac{d}{2H}\,(d-w)\,, \qquad & r &= \frac{(H^2-W^2)}{2dH}\,(d-w)\,.
\end{aligned}$$

We now have all we need to draw a slanted string centered along the view rectangle's diagonal. Here's some pseudo-code:

```
d = sqrt(W*W + H*H);     // length of view rectangle's diagonal
f = min(W,H) / max(W,H); // view rectangle's aspect ratio
a = atan(f);             // angle to rotate the string rectangle by

TOL = 0.05;              // some appropriately small positive number

minfs = MINFS;           // minimum font size
maxfs = MAXFS;           // maximum font size
fs = maxfs;              // initial font size
```

```
while (true)
{
    oldfs = fs;
    fs = (minfs + maxfs) / 2.0;

    if (|fs - oldfs| < TOL)
    { break; } // we've converged on the optimal font size, so we're done

    // htg = height of tallest glyph for the current font and font size
    htg = appropriate Cocoa call here

    // w = string length in pixels for the current font and font size
    w = appropriate Cocoa call here

    // compute the desired string rectangle's height
    h = f * (d - w);

    if (w > d || htg > h) // string too long or tallest glyph too tall
    { maxfs = fs; } // decrease font size
    else
    { minfs = fs; } // increase font size
}
```

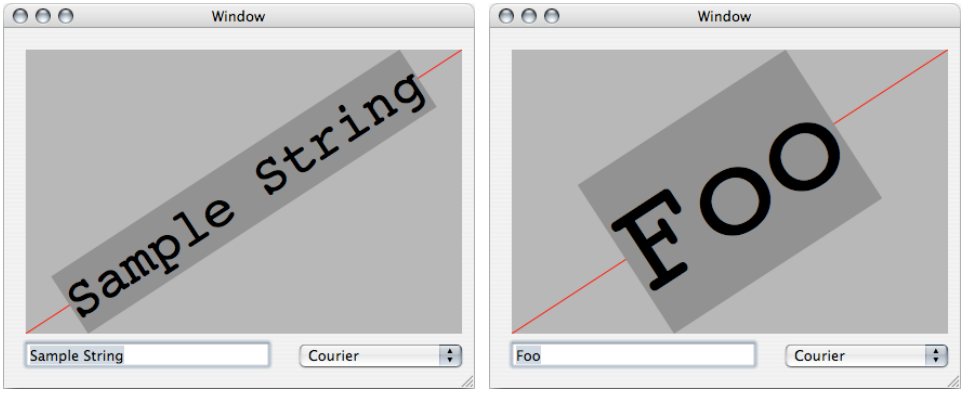
At this point, we've determined the width (w) and height (h) of the rectangle inside of which the string will be drawn, as well as the optimal font size for the string in question. We can then use these values to center the string both vertically and horizontally inside this rectangle, using the appropriate Cocoa calls.

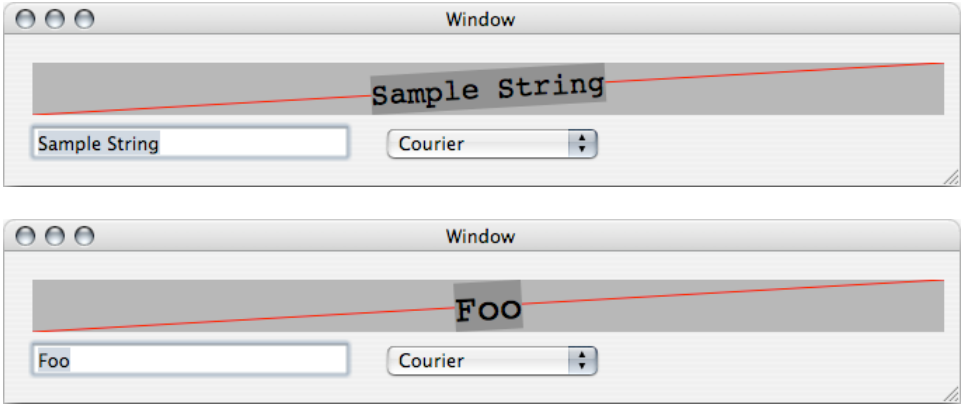
Next, we need to compute (x_a, y_a), the position of the anchor point around which the string rectangle should be rotated:

$$x_a = x = \frac{d}{2W}\,(d - w)\,, \qquad y_a = 0\,, \qquad \text{if } H \leq W\,,$$

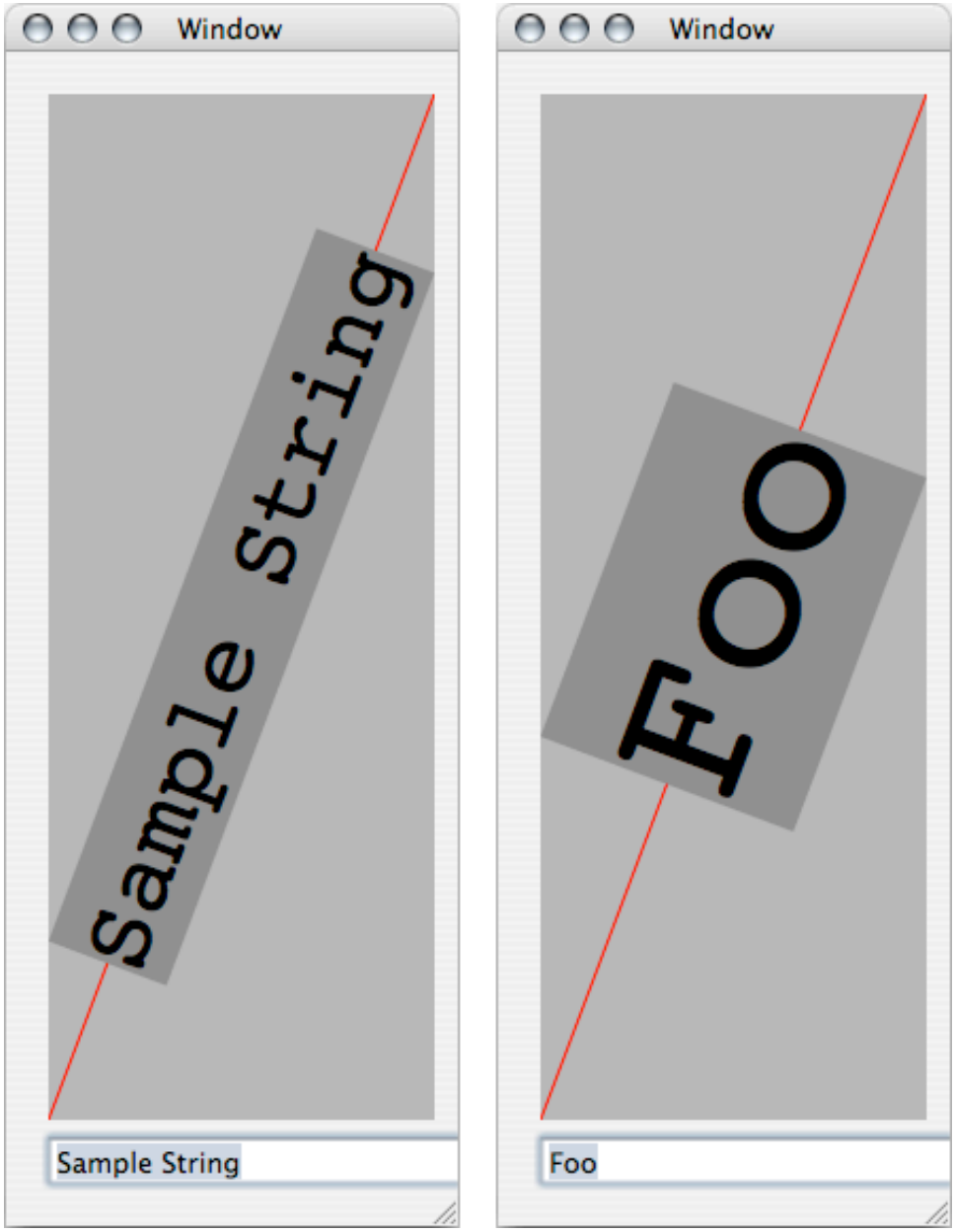
$$x_a = x = \frac{W}{d}\,(d - w)\,, \qquad y_a = r = \frac{(H^2 - W^2)}{2dH}\,(d - w)\,, \qquad \text{if } W \leq H\,.$$

Finally, position the string rectangle such that its lower left vertex lies at the position determined by (x_a, y_a), then rotate it counterclockwise by a degrees, if $H \leq W$, or by $(90 - a)$ degrees, if $W \leq H$. Here are some sample results, from an actual application implementing the algorithm above:


Window
Sample String
Sample String
Courier
Window
Foo
Foo
Courier


Window
Sample String
Sample String
Courier
Window
Foo
Foo
Courier


Window
Sample String
Sample String
Window
Foo
Foo


Window
Sample String
Sample String
Window
FOO
Foo