Subject: Re: Optimized storage of 1BPP images
Posted by mdelfede on Thu, 07 Feb 2008 13:32:03 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Thu, 07 February 2008 13:54There's third way too.

With zooms in range 0..100% the new scaled image is always smaller/equal to original picture, so the memory consumptions is under O(2). I don't think this case needs further improvements, and I would keep it as it is. (actually this method is probably optimal even for zooms like 100-200%)

When the zoom is > 100%, you can try different approach, not to scale whole image, but only the current viewport of it.
In case even this is too slow, you may during panning use the previous viewport from cache, and rescale only the parts which are newly on screen.

I'm not sure how large your viewport is, but anything under 1000x1000 pixels should be well manageable by GHz CPUs, if you have the original picture in RAM.

Images are 1728 x 2210 pixels per page, with often tenths of pages... in RGBA mode they're really too big to buffer.
Quote:

In this way your rescaled image for >100% zooms will have fixed size, and will not grow further with bigger zooms, also with bigger zooms you will use smaller and smaller part of original picture to create the rescaled one, so the performance will grow.

I guess it's not so simple. When I do an on-the-fly rescale, it works like you say, so the performance is limited only by control's width, at whorse. But if I do buffering, I need to scroll/pan over all image, so I must buffer the whole. Even whorse, having continuous scroll between pages, I must buffer whole file.

Quote:
..... and unless your view has like 2000x1000 pixels, the modern CPU + RAM should easily rescale such amount of data on the fly.

that's the problem

I guess I'll take out the buffering stuff and follow as before, keeping the slight feeling of bumpy scroll. All other ways seems to me whorse or too time expensive to code....

Thanx to all for answers !

Ciao

Max