Subject: Slow image drawing on big images
Posted by mdelfede on Tue, 29 Jan 2008 13:31:30 GMT
View Forum Message <> Reply to Message

I'm making a fax viewer application, that must retrieve a .tiff multipage file from a fax server and display it on screen.
So, I found following problems :

1 - lack (IMHO...) of support of multi-image tiffs
2 - *very* slow speed of image drawing using ImageCtrl

I guess point 1 is easy to solve, just adding few lines of code both in tif library and in Image class.
Point 2... I don't know how to handle it, so I have some questions.
I've seen that drawing is done in

void Image::Data::Paint(Draw& w, int x, int y, const Rect& src, Color c)

in ImageX11.cpp. As I can understand, the image is converted to an X11 one on the fly just before drawing... It would maybe make sense to keep (optionally) the X11 converted image buffered so on next paints it goes faster.
As is it now when I change control size (dragging main window) the pic takes some thenths of second to redraw, so real time pan/zoom is impossible.

Any smart solution ?

Ciao

Max

Subject: Re: Slow image drawing on big images
Posted by mirek on Tue, 29 Jan 2008 13:35:26 GMT
View Forum Message <> Reply to Message

Interface for multipaged images is in fact already in "Raster"... so only plugin/tiff has to be fixed.

As for speed, many issue can be involved. Image IS cached. Anyway, you have not told us whether you are rescaling it or reloading it.

Maybe posting at least your paint routine would help.

Mirek

Subject: Re: Slow image drawing on big images
Posted by mdelfede on Tue, 29 Jan 2008 14:03:30 GMT

simple app to test stuff, with companion .tif file attached here.
app should be put in gatofax assembly, but it's easily changed.

Just run the app, then try to resize/shrink main window.

Ciao

Max

File Attachments

1) GatoFax.zip, downloaded 637 times

Subject: Re: Slow image drawing on big images
Posted by mirek on Tue, 29 Jan 2008 18:38:59 GMT

Well, the problem simply is that the image is too big

If cx or cy is bigger than 2000, U++ heurestics decides that you are trying to print it. In that case, "RLE banding and rect compression" algorithm kicks in to reduce the data send to printer and also the amount of memory needed to print -> slow.

The fix is simple - specify the source rectangle. E.g.:

```
struct App : TopWindow {
 Image img;

 virtual void Paint(Draw& w) {
  w.DrawImage(0, 0, img, (Rect)GetSize());
 }

 App() {
  Sizeable().Zoomable();
     img = StreamRaster::LoadFileAny( "e:/fax000000007.tif" );
 }
};


GUI_APP_MAIN
{
 App().Run();
}
```

Mirek

Subject: Re: Slow image drawing on big images
Posted by mirek on Tue, 29 Jan 2008 19:04:16 GMT
View Forum Message <> Reply to Message

On the second thought, I have also improved heurestics to avoid banding when the output is screen or ImageDraw.

Mirek

Subject: Re: Slow image drawing on big images
Posted by mdelfede on Tue, 29 Jan 2008 21:27:48 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 29 January 2008 20:04On the second thought, I have also improved heurestics to avoid banding when the output is screen or ImageDraw.

Mirek

Much better
Already on UVS ?

Max

Subject: Re: Slow image drawing on big images
Posted by mirek on Tue, 29 Jan 2008 21:30:13 GMT
View Forum Message <> Reply to Message

Yes.

Subject: Re: Slow image drawing on big images
Posted by mdelfede on Tue, 29 Jan 2008 22:56:22 GMT
View Forum Message <> Reply to Message

WOW, that was a great speed improvement !
Thanx !

Max

EDIT : I need also to implement ZOOM, PAN and a multipage interface. For PAN, no problem, I can derive from ImageCtrl.
Regarding ZOOM I wonder if there's not a fastest way than rescaling the image on the fly.... Does the main image paint code provide some stuff for resizing image on the fly, maybe with help of native code ?
For the multipage stuff I'll look tomorrow on Raster code.

Ciao

Max

Subject: Re: Slow image drawing on big images
Posted by mirek on Wed, 30 Jan 2008 08:01:25 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Tue, 29 January 2008 17:56WOW, that was a great speed improvement !
Thanx !

Max

EDIT : I need also to implement ZOOM, PAN and a multipage interface. For PAN, no problem, I can derive from ImageCtrl.
Regarding ZOOM I wonder if there's not a fastest way than rescaling the image on the fly.... Does the main image paint code provide some stuff for resizing image on the fly, maybe with help of native code ?
For the multipage stuff I'll look tomorrow on Raster code.

Ciao

Max

IMO, forget about ImageCtrl. It was not designed for this purpose and it is about 20 lines anyway. Roll your own widget.

For zooming, just use Rescale, it is close to perfect. Also, you might prefer cached variant, use

Image CachedRescale(const Image& m, Size sz, const Rect& src);

or

Image CachedRescalePaintOnly(const Image& m, Size sz, const Rect& src);

PaintOnly means that you do not intent to access the pixel matrix anymore; Draw is then allowed to free the memory. "Cached" means you can call this function in Paint routine; if source

parameters do not change, Image is restored from the cache.

Mirek

Subject: Re: Slow image drawing on big images
Posted by mdelfede on Wed, 30 Jan 2008 09:33:48 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 30 January 2008 09:01

IMO, forget about ImageCtrl. It was not designed for this purpose and it is about 20 lines anyway. Roll your own widget.

For zooming, just use Rescale, it is close to perfect. Also, you might prefer cached variant, use

Image CachedRescale(const Image& m, Size sz, const Rect& src);

or

Image CachedRescalePaintOnly(const Image& m, Size sz, const Rect& src);

PaintOnly means that you do not intent to access the pixel matrix anymore; Draw is then allowed to free the memory. "Cached" means you can call this function in Paint routine; if source parameters do not change, Image is restored from the cache.

What I need is :

1-load images from a multipage tiff file
2-make some small thumbnails of it to show on left side (easy)
3-show the pages on the center, with zoom/rotate/pan ability

For the point 3, I'd rescale all the images at once on zoom request (zooms are seldom than pans), but I'd keep the original images in order to avoid multiple file loading and to not loose quality on successive rescalings. So, original images + scaled images to display them quickly + thumbnails. The best would be a rescale that keeps the cached image up to a new rescale, but it should allow panning, so multiple partial displays of it... I don't know if CachedRescale does the job.

Ciao

Max