
Subject: Rescale and copy area of source image into an area of dest image

Posted by [mdelfede](#) on Sun, 02 Mar 2008 19:32:48 GMT

[View Forum Message](#) <> [Reply to Message](#)

I could not manage to do an apparently simple stuff : copy a rectangular area on source image, rescale it and put in another rectangular area on a destination imagebuffer.

This is my piece of code :

```
// extract and rescale image area
ImageEncoder t;
Rescale(t, destSize, page, sourceRect);
// copies rescaled image inside cache rectangle
const RGBA *rgba = t;
for(int iLine = 0 ; iLine < t.GetHeight(); iLine++)
{
    Copy(imageCache[i+rect.top]+rect.left, rgba, t.GetWidth());
    rgba += t.GetWidth();
}
```

or :

```
// extract and rescale image area
ImageEncoder t;
Rescale(t, destSize, page, sourceRect);
Image img = Image(t);
// copies rescaled image inside cache rectangle
const RGBA *rgba = img;
for(int iLine = 0 ; iLine < img.GetHeight(); iLine++)
{
    memmove(imageCache[i+rect.top]+rect.left, rgba, img.GetWidth()* sizeof(RGBA));
    rgba += img.GetWidth();
}
```

And many other options, but no way, destination stuff doesn't contain the required copy. I know that 't' contains a valid image because painting it directly is ok.

Any hint ?

Ah, last stuff.... ImageDraw class (which could be used for this purpose....) is documented in help with many member functions that it doesn't have in reality, first of them the constructor

ImageDraw(Image &)

Ciao

Max

Subject: Re: Rescale and copy area of source image into an area of dest image
Posted by [mdelfede](#) on Mon, 03 Mar 2008 16:52:05 GMT

[View Forum Message](#) <> [Reply to Message](#)

well, partly solved... the bug was not here... for others interested in using Image and Imagebuffer, beware :

```
ImageBuffer buf;  
Draw d = .....  
.....  
.....  
d.DrawImage(buf, .....)
```

will **not** work, or better, it'll work ONCE.

DrawImage expects an Image& parameter, so buf is converted on the fly onto Image object, loosing its content due to destroy-on-copy behaviour.

The correct code is :

```
ImageBuffer buf;  
Draw d = .....  
.....  
.....  
Image img = buf;  
d.DrawImage(img, .....)  
buf = img;
```

quite un-intuitive, indeed. It would be much better to have a different version of DrawImage with an ImageBuffer & as first parameter. Also, allowing such automatic type conversion on objects that have the destroy-on-copy semantics can lead to subtle bugs...

Ciao

Max

Subject: Re: Rescale and copy area of source image into an area of dest image
Posted by [masu](#) on Mon, 03 Mar 2008 17:37:26 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Max,

can't you just use it like this:

ImageBuffer buf;

Draw d =

.....

.....

d.DrawImage(Image(buf),)?

Matthias

Subject: Re: Rescale and copy area of source image into an area of dest image

Posted by [mdelfede](#) on Mon, 03 Mar 2008 20:32:58 GMT

[View Forum Message](#) <> [Reply to Message](#)

masu wrote on Mon, 03 March 2008 18:37Hi Max,

can't you just use it like this:

ImageBuffer buf;

Draw d =

.....

.....

d.DrawImage(Image(buf),)?

Matthias

mhhhh.. I think it'll not work, as imagebuf get still converted on a temporary image, so it loses its contents.

Max

Subject: Re: Rescale and copy area of source image into an area of dest image

Posted by [masu](#) on Mon, 03 Mar 2008 22:11:19 GMT

[View Forum Message](#) <> [Reply to Message](#)

Max, I think you are right.

The documentation says, conversion clears buffer when converted to image and vice versa.

Matthias
