Subject: Re: Inverse palette conversion algorithm...
Posted by mirek on Fri, 07 Apr 2006 11:33:19 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Fri, 07 April 2006 07:161) why to use indexed color? (ok, sometimes it's still really needed, but usually 32b ARGB is the way to go)
2) so you speeded up 20 times code which is run single time during init and takes under 100ms even with brute-force algorithm, and you needed 3 days to do that.
2 - related) in case you generate the conversion table more than one time, there's something fishy about that. (and looks like you can speed up the thing much more just by omiting multiple initialization of the same table)

Add 1) Yes, of course, that is BTW the main motivation why I completely refactor U++ Images now. HOWEVER, for certain applications, palette images are still needed. E.g. webserver could need to generate .gifs based on some RGBA data and this algorithm has to be part of it (and would limit the speed of webserver significantly; "brute force" algorithm spend 100ms on 2.4Ghz AMD64 machine, that simply was unacceptable).

Also, believe it or not, another motivation is that at least two of our existing applications deal with images that simply do not fit into memory even when indexed - e.g. nice 256 color, 32768x32768 pixels - >1GB of data. If you want to deal with such beasts, you need to have some nice tools to do so (of course speed of inverse transformation does not matter here).

2) You have to generate such table for each used palette. And each time you are about to save RGBA image in indexed format, you need to generate such inverse transformation cube.

(And sure, there is always "because I can" factor

Mirek