Forget most of what I said in my previous post I dug deeper and found out that the reason why
XGetImage fails is most probably that it is called on an image with one dimension equal to zero.
The other calls are guarded against such thing, but this one for alpha layer was left unprotected.

I believe that if the image is empty (that is zero pixels in any direction, or in code
GetSize().IsEmpty()==true), we don't have to bother drawing it at all, right? Or are there some side
effects I didn't notice? I'd suggest something like: ImageDraw::operator Image() const
```
{
 if(size.IsEmpty()){            // this block is added
  ImageBuffer ib(size);
  return ib;
 }
 GuiLock __;
 XImage *xim = XGetImage(Xdisplay, dw, 0, 0, max(size.cx, 1), max(size.cy, 1), AllPlanes,
ZPixmap);
 Visual *v = DefaultVisual(Xdisplay, Xscreenno);
 RasterFormat fmt;

 RGBA   palette[256];
// ...
}
```

For me it works fine and eliminates the crashes. The roll-out animation of color selector even
looks a bit smoother now (but that might be just my imagination).

Oh, and I almost forgot to mention that the troublesome call to this operator Image() is located in
Image WheelRampCtrl::PaintWheel(Size size) in DlgColor.cpp, in the "return iw;" statement (iw is
the empty ImageDraw) where implicit conversion to Image happens.

Honza