How to blend two pf24bit images using ScanLine

> ***QUESTION***
>
> Does anyone know how to implement basic alphablending without using Win2000 *AlphaBlend* function? I am trying to write a simple graphics program that will perform some alphablending on Win95 and Win98, but I don't know the necessary steps.

Here's a method of blending two bitmaps. It uses 24-bit bitmaps and scanlines, so only works with D3 or higher. It assumes that variable *b*, a *TBitmap*, has pixelformat *pf24bit*.

```
type  {for scanline access to 24-bit bitmaps}
  TRGBArray = Array[0..32767] of TRGBTriple;
  pRGBArray = ^TRGBArray;
```

Gradient is one bitmap, *b* is the other. *Amount* is the percentage of the *Gradient* image to blend with *b*. *tBufr* is an existing *TBitmap* that's used internally for operations, *b* is the original bitmap. *tBufr is sized to match b, and Gradient is stretched into it for this operation. All bitmaps are created and freed elsewhere!*

```
procedure TForm1.MergeGradient(Amount: integer);
var
  pb, pc: pRGBArray;
  x, y: integer;
  GrdPct: Single;
  ImgPct: Single;
begin
  Screen.Cursor := crHourGlass;
  GrdPct := Amount / 100;
  ImgPct := 1 - GrdPct;
  tBufr.Width := b.Width;
  tBufr.Height := b.Height;
  tBufr.PixelFormat := pf24bit;
  tBufr.Canvas.StretchDraw(
    Rect(0, 0, tBufr.Width, tBufr.Height), Gradient
  );
  for x := 0 to tBufr.Height-1 do
  begin
    pb := tBufr.ScanLine[x];
    pc := b.ScanLine[x];
  for y := 0 to tBufr.Width - 1 do
  begin
    pb[y].rgbtRed := Round(
      pb[y].rgbtRed * GrdPct + pc[y].rgbtRed * ImgPct
    );
    pb[y].rgbtBlue := Round(
      pb[y].rgbtBlue * GrdPct + pc[y].rgbtBlue * ImgPct
    );
    pb[y].rgbtGreen := Round(
      pb[y].rgbtGreen * GrdPct + pc[y].rgbtGreen * ImgPct
    );
  end;
  end;
  Image1.Picture.Assign(tBufr);
  Screen.Cursor := crDefault;
end;
```

Original resource:	The Delphi Pool
Author:	Harm
Added:	2009-11-06
Last updated:	2009-11-06