

Courses IPCx, C2: Histo, FAQs

Copyright © by V. Miszalok, last update: 30-01-2001

Umwandlung von 8-Bit-DIBs in 24-Bit-DIBs und zurück

8-Bit ohne Palette in 24 Bit

1. Gegeben sei ein 8-Bit-Bitmap der Zeilenlänge $xSize8$ durch BYTE $B8[xSize8*ySize]$.
2. Suche eine Zeilenlänge $xSize24 \geq xSize8$ so, dass $3*xSize24$ ohne Rest durch 4 teilbar ist.
3. Reserviere Speicherplatz für ein 24-Bit-Bitmap RGBTRIPLE $B24[xSize24*ySize]$.

```
for ( y=0; y < ySize; y++)
  for (x=0; x < xSize8; x++ )
    B24[y*xSize24+x].rgbtBlue =
    B24[y*xSize24+x].rgbtGreen =
    B24[y*xSize24+x].rgbtRed = B8[y*xSize8+x];
```

8-Bit mit Palette in 24 Bit

1. Gegeben sei ein 8-Bit-Bitmap der Zeilenlänge $xSize8$ durch BYTE $B8[xSize8*ySize]$.
2. Gegeben sei eine Palette in der Form RGBQUAD $palette[256]$;
3. Suche eine Zeilenlänge $xSize24 \geq xSize8$ so, dass $3*xSize24$ ohne Rest durch 4 teilbar ist.
4. Reserviere Speicherplatz für ein 24-Bit-Bitmap RGBTRIPLE $B24[xSize24*ySize]$.

```
for ( y=0; y < ySize; y++)
  for (x=0; x < xSize8; x++ )
    { B24[y*xSize24+x].rgbtBlue = palette[ B8[y*xSize8+x] ].rgbBlue;
      B24[y*xSize24+x].rgbtGreen = palette[ B8[y*xSize8+x] ].rgbGreen;
      B24[y*xSize24+x].rgbtRed = palette[ B8[y*xSize8+x] ].rgbRed;
    }
```

24-Bit in 8-Bit ohne Palette

1. Gegeben sei ein 24-Bit-Bitmap der Zeilenlänge $xSize24$ durch RGBTRIPLE $B24[xSize24*ySize]$.
2. Suche eine Zeilenlänge $xSize8 \geq xSize24$ so, dass $xSize8$ ohne Rest durch 4 teilbar ist.
3. Reserviere Speicherplatz für ein 8-Bit-Bitmap BYTE $B8[xSize8*ySize]$.

```
for ( y=0; y < ySize; y++)
  for (x=0; x < xSize24; x++ )
    B8[y*xSize24+x] =
    ( B24[y*xSize24+x] ].rgbtBlue +
      B24[y*xSize24+x] ].rgbtGreen +
      B24[y*xSize24+x] ].rgbtRed ) / 3;
```

24-Bit in 8-Bit mit Palette

Diese Aufgabe ist exakt nicht lösbar, weil man den RGB-Farbraum unmöglich auf 256 einzelne Farben abbilden kann.

Diese Aufgabe kann man nur durch brutale Näherungen lösen, die hier nicht behandelt werden.