

*Beispiel:*

*Bei einem Format von 720x576 Bildpunkten (Pixel) pro Bild, (was einem gewöhnlichen Fernsehformat (PAL) entspricht) hat man bei einer Farbtiefe von 24 Bit und einer Bildfrequenz von 25 Bildern pro Sekunde (fps) eine Datenrate von  $720 \times 576 \text{ Pixel} \times 24 \text{ Bit} \times 25 \text{ fps} = 259,2 \text{ Mbit/sek}$ , also ca. 32,4 MByte/sek. Damit wäre eine Festplatte von 100 GByte in ca. 51 min voll! Hier wird also schon deutlich, wie wichtig Datenreduktionsmethoden in der Bildverarbeitung und -übertragung sind.*

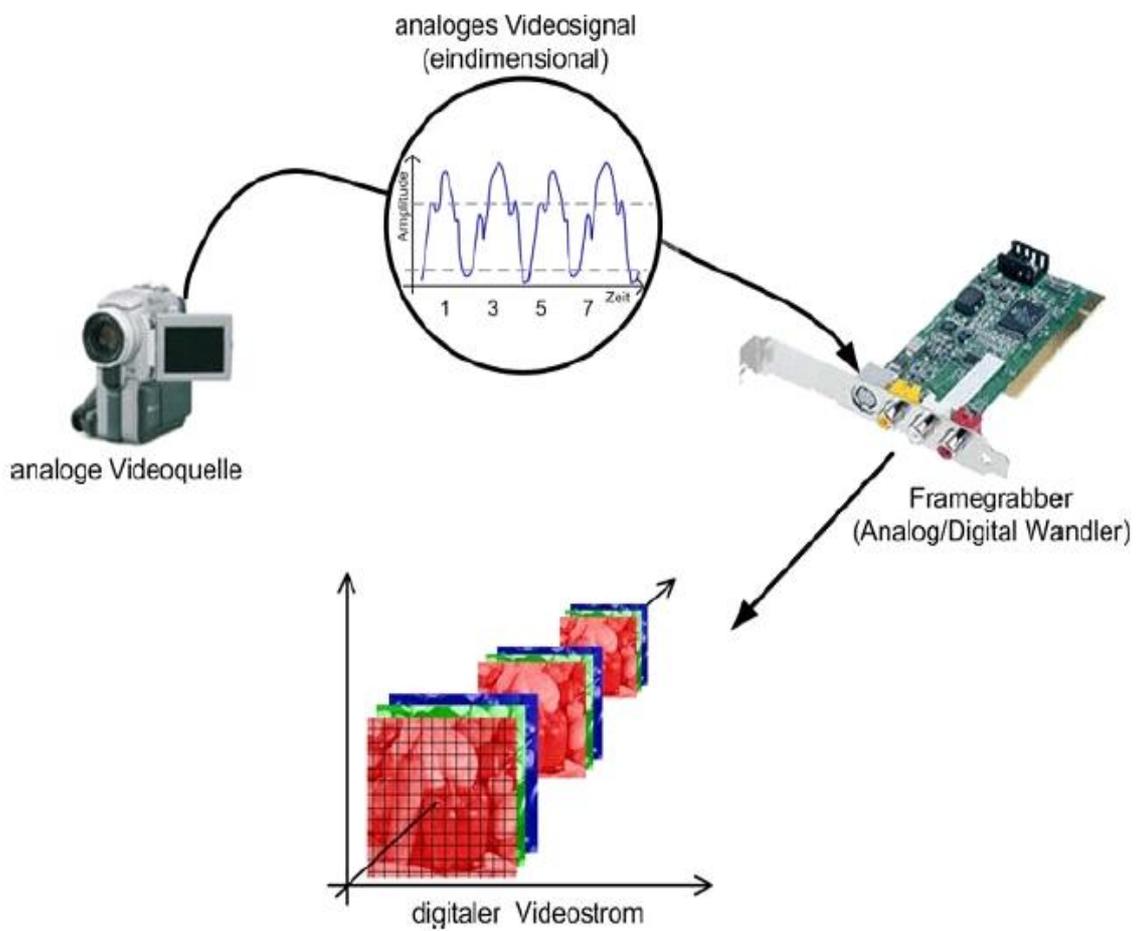


Abb. 1.13 analoges Videosignal - Framegrabber

PCI-USB2.0 Karte von  
Asus



Audio- Videotechnik

**Datenerfassung**

Hochschule Zittau/Görlitz

Prof. Dr. S.Bischoff

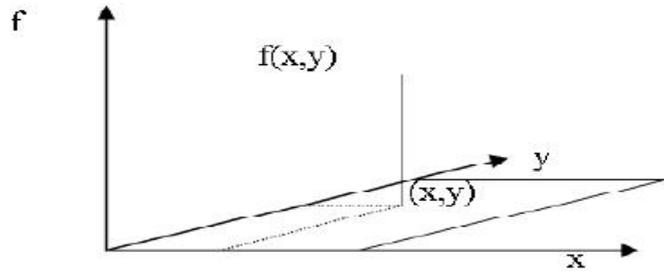


Abb. 1.3 3D Darstellung

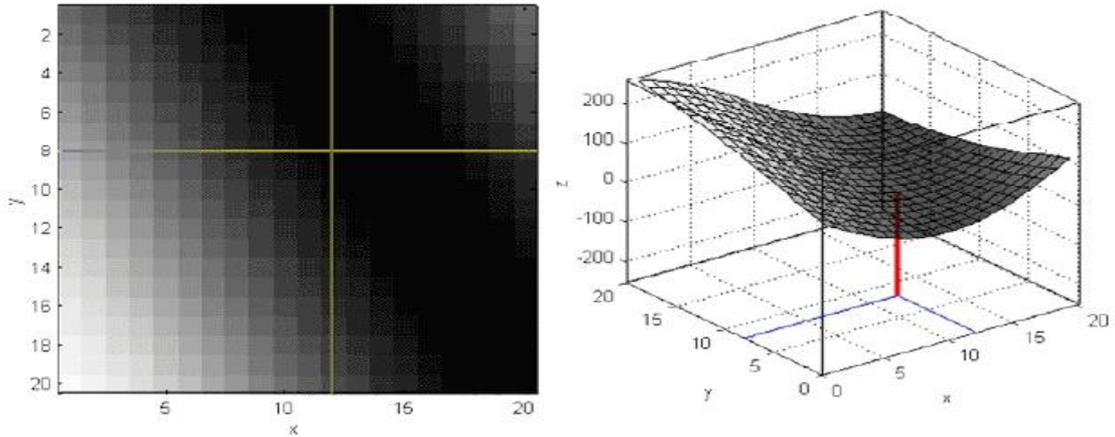


Abb. 1.4 Beispiel für Grauwert in Digitalen Bildern

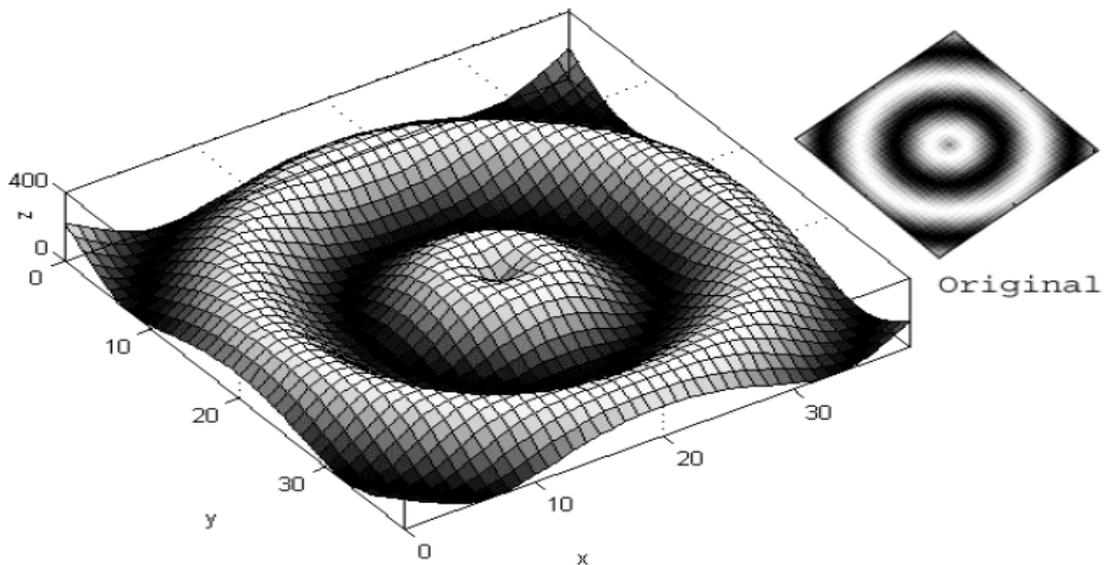


Abb. 1.5 Beispiel zur Visualisierung von Grauwerten

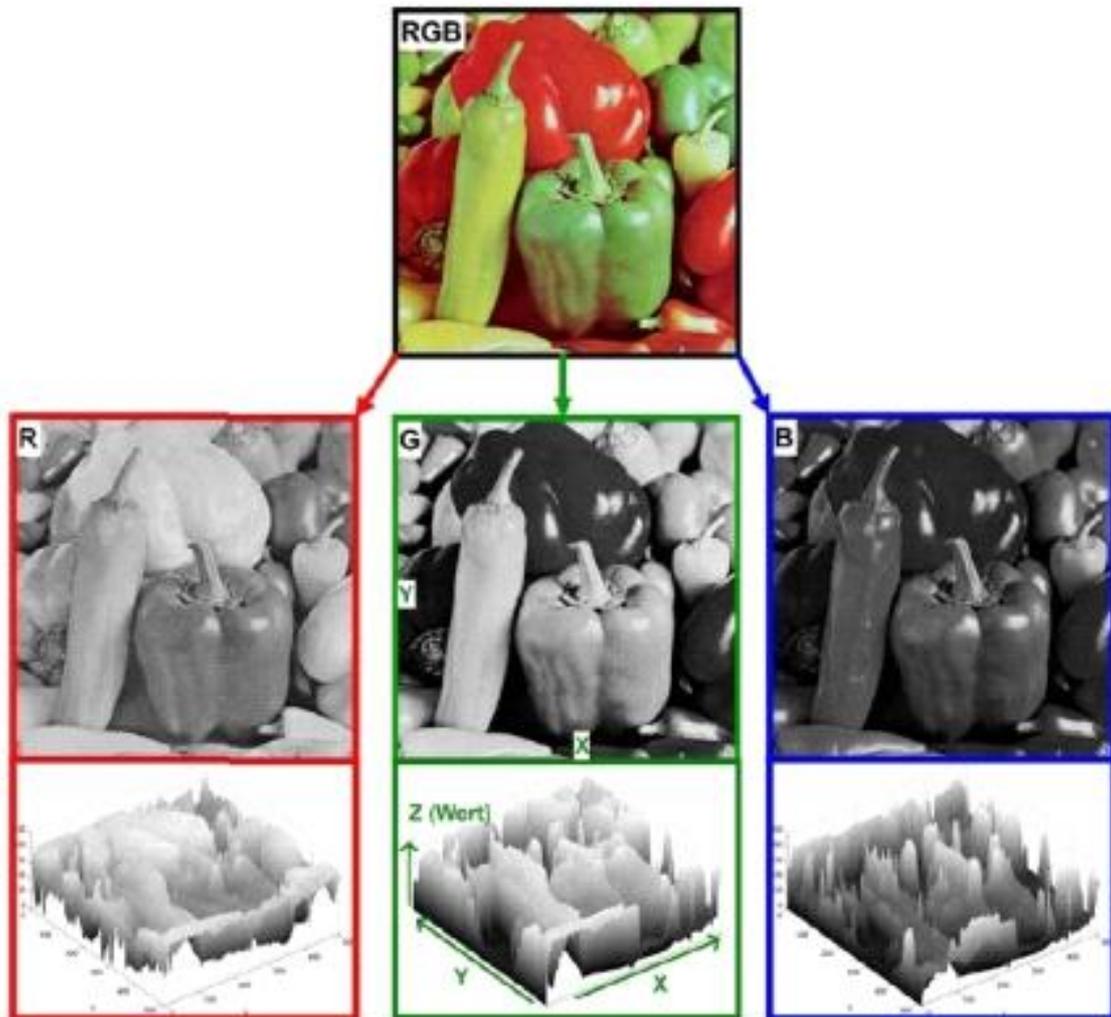
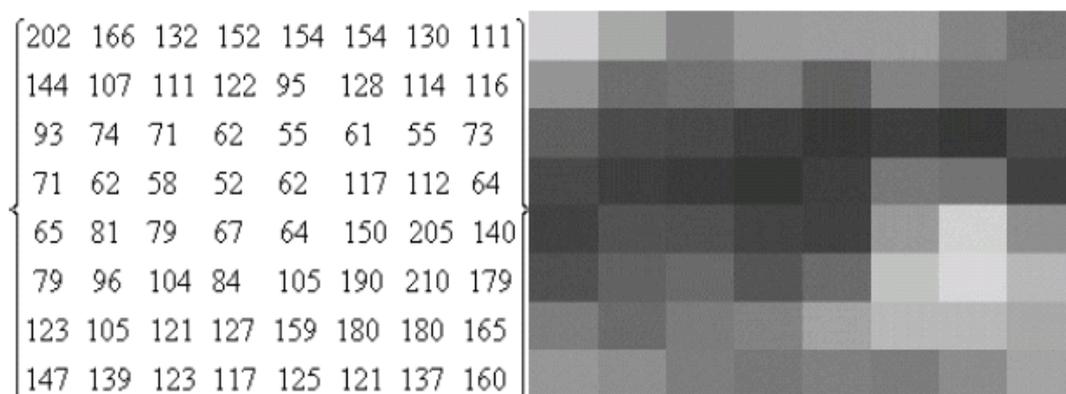


Abb. 1.16 RGB Farbkanäle eines digitalen Bildes  
 (helle Grauwerte = große Farbamplituden, Dunkle Grauwerte = kleine Farbamplituden)

Audio- Videotechnik	Farbbilder	Hochschule Zittau/Görlitz Prof. Dr. S.Bischoff
---------------------	------------	---

$$f(n,m) = \begin{pmatrix} f(0,0) & f(0,1) & \cdot & f(0,N-1) \\ f(1,0) & \cdot & & \cdot \\ f(2,0) & & \cdot & \cdot \\ & & & \cdot \\ f(M-1,0) & & & f(M-1,N-1) \end{pmatrix}$$

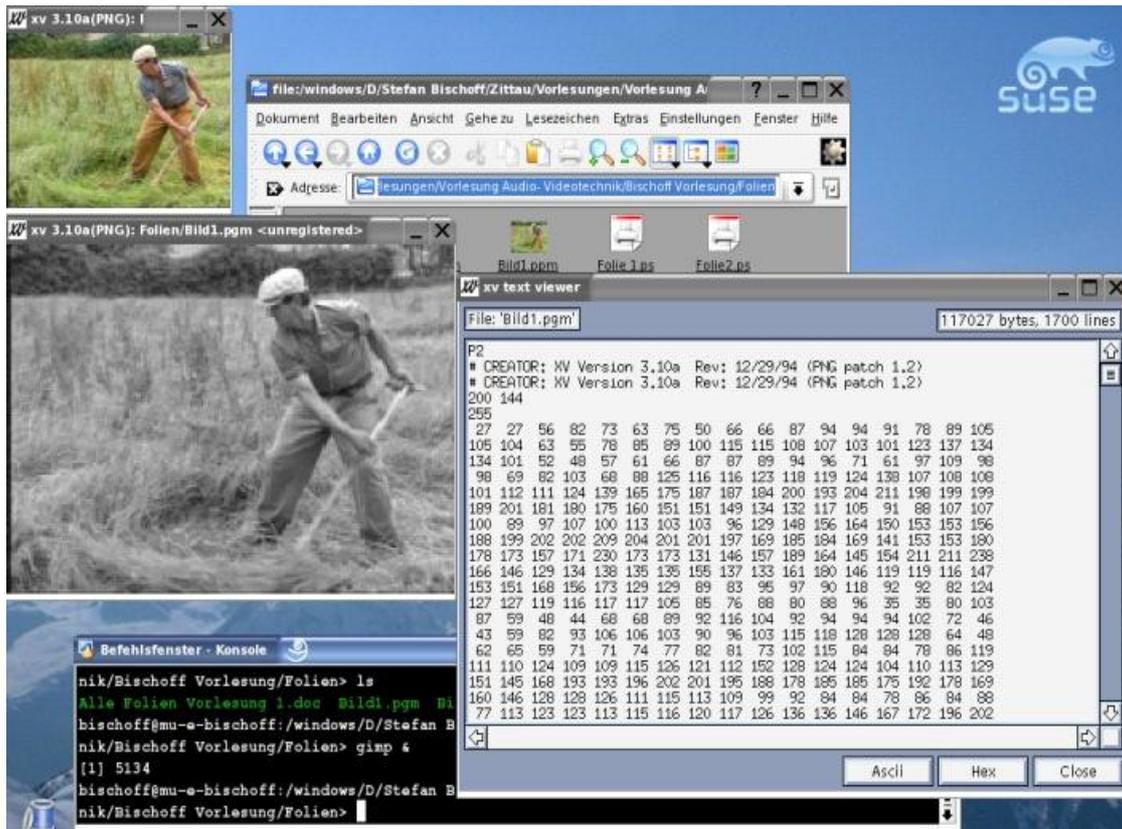
Beispiel für einen (8x8 Grauwertblock):



## Unkomprimierte Bildformate

Portable Gray Map     \*.pgm     Grauwertbilddatei

Portable Pix Map     \*.ppm     Farbbilddatei



### Dateiinhalt:

```
P2
# CREATOR: ...
n m
255
27 27 56
```

### Erklärung:

PGM-Format  
Kommentarzeilen im Header n  
Spalten (Breite) m Zeilen (Höhe)  
0 .. 255 mögliche Grauwerte  
Grauwerte der Pixel von links  
oben nach rechts unten

```

/* Einfacher C-Sourcecode zum Laden und Speichern von pgm-
Files */
int          d1, d2, ibuf, i, j, n, m;
char         text1[250], text2[250], sbuf[20];
unsigned char *buf;
long int     size;
File         *din, *dout;

main (void)
{
printf( "Name des Pgm-Files, Eingabe \n");
scanf( "%s", text1);
din := fopen(text1,"r"); // Lesedatei öffnen
printf( "Name des Pgm-Files, Ausgabe \n");
scanf( " %s ", text2);
dout := fopen( text2, "w"); // Zieldatei öffnen
fgets( sbuf, sizeof(sbuf), din); // erste Zeile einlesen
while( sbuf[0] == "#" )
    { fgets( sbuf, sizeof(sbuf), din); // Kommentare einlesen}
sscanf( sbuf, "%d %d", &n, &m); // Dimension einlesen
size := d1 * d2; // Grösse des Speichers festlegen
if ( (buf := (unsigned char*) malloc(size)) != NULL )
    {printf(stderr, "Fehler beim Speicher allocieren !");}
fread (buf, 1, size, din); // alle Pixel im Speicher buf
fclose (din); // Lesedatei schliessen
fprintf(dout, "P2 \n %d %d", n ,m); // Header in Zieldatei schreiben
for (j:=0; j<m; j++)
    for ( i:=0; i<n; i++) {
        ibuf := (int) buf[ j*n + i]; // Hier kann man auf die Pixel
zugreifen
        fprintf(dout, "%d", ibuf); // Grauwert eines Pixels in Zieldatei
    }
fclose(dout); // Zieldatei schliessen
}

```