Universität Mannheim
Lehrstuhl für Praktische Informatik IV
Professor Dr. W. Effelsberg
Stephan Fischer, Raum 415

Multimedia-Technik
Wintersemester 1996/97
Übungsaufgaben

Multimedia-Technik: Übungsblatt 13

Aufgabe 1 *Inhaltsanalyse*

Entwickeln Sie einen Algorithmus, der auf Basis von Histogrammen Schnitte in digitalem Video automatisch erkennt. Die Bilder seien dabei als char image[height]-[width] vereinbart. Wie kann man erreichen, daß die Schnitterkennung nicht durch starke Bewegung von Objekten in Bildsequenzen gestört wird? **Lösung:** int histogram[2][256];

```
int cutDetect(char *image, int w, int h, int histoNumber)
/* w und h sind die Bildmaße, histoNumber gibt das zu verwendende Histogr. an
*/
{
int i,j;
   j=0;
   for (i=0;i<256;i++) histogram[histoNumber][i]=0;
   for (i=0;i<w*h;i++) histogram[histoNumber][image[i]]++;
   for (i=0;i<256;i++) j+=ABS(histogram[0][i]-histogram[1][i];
   if (((double)j/(w*h))>0.4) return 1; /* Cut detected */
   return 0;

}
```

Starke Objektbewegung stört das Verfahren deshalb nicht, da sie im Histogramm keinerlei Einfluß hat. Daher muß man keine Maßnahmen treffen, um dies zu verhindern.

Aufgabe 2 *Inhaltsanalyse*

Entwerfen Sie einen rekursiven Region-Growing-Algorithmus, der die in einem Bild enthaltenen Objekte findet. Welche Probleme treten dabei auf? Wie lassen sie sich lösen?

Im hier angegebenen Algorithmus wird die Funktion *inImage* verwendet. Diese prüft zum einen, ob der zu prüfende Pixel noch im Bild ist und ob dieser das Homogenitätskriterium des Region Growing erfüllt.

```
char image[width*height];
char visited[width*height];
void regionGrow(int j,int k, int l)
{
   int i;
   if !(visited[j*width+k])
      {
         visited[j*width+k]=1;
         if inImage(j-1,k-1)
            if !(visited[(j-1)*width+k-1]) regionGrow(j-1,k-1,l);
         if inImage(j-1,k)
            if !(visited[(j-1)*width+k]) regionGrow(j-1,k,l);
         if inImage(j-1,k+1)
            if !(visited[(j-1)*width+k+1]) regionGrow(j-1,k+1,l);
         if inImage(j,k-1)
            if !(visited[j*width+k-1]) regionGrow(j,k-1,l);
         if inImage(j+1,k+1)
            if !(visited[j*width+k+1]) regionGrow(j,k+1,l);
         if inImage(j+1,k-1)
            if !(visited[(j+1)*width+k-1]) regionGrow(j+1,k-1,l);
         if inImage(j+1,k)
            if !(visited[(j+1)*width+k]) regionGrow(j+1,k,l);
         if inImage(j+1,k+1)
            if !(visited[(j+1)*width+k+1]) regionGrow(j+1,k+1,l);


      }
}
int main();
{
   int i,k,l;     l=1; /* Object number */
```

```
    for (i=0;i<width*height;i++)
        visited[i]=(char)0;
    for (i=0;i<height;i++)
        for (k=0;k<width;k++)
            if !(visited[i*width+k]) regionGrow(i,k,l++);
}
```

Aufgabe 3 *Inhaltsanalyse*

Betrachten Sie eine Live-Übertragung über eine Telefonleitung. Wie viele Werte können Sie pro Sekunde abtasten, wenn die Bandbreite von 64 kbit/s nicht überschritten werden soll und ein Wert mit 8 Bit gespeichert wird?

Lösung:

Pro Sekunde können 8000 Werte abgetastet werden. Dies bedeutet nach dem Shannon Theorem (die Abtastrate ist doppelt so hoch wie die maximal abtastbare Frequenz), daß 4000 Hz abgetastet werden können.

Aufgabe 4 *Inhaltsanalyse*

Der Grundton ist die tiefste Hauptfrequenz eines Audiospektrums. Entwickeln Sie eine Methode, die diesen Grundton automatisch bestimmt.

Lösung: Zuerst muß ein bestimmter Audiobereich (typischerweise 30 ms) über die Fouriertransformation in den Frequenzbereich überführt werden. Anschließend stellt sich die Aufgabe, das Vorkommen der Maxima dieses Frequenzspektrums pro Sekunde zu zählen. Dies entspricht genau dem Grundton. Dazu berechnet man die Autokorrelation des Frequenzspektrums mit dem folgender Frequenzspektra. Diese Korrelation wird dann hoch sein, wenn ein ähnliches Frequenzspektrum gefunden wird. Zählt man nun die Anzahl der Korrelationsmaxima pro Sekunde, so hat man genau den Grundton.