

# Praktikum Multimedia-Technik

## Blatt 6 – Analyse der Kamerabewegung

### Aufgabe 9 – Ermittlung der Bewegung in Bildern

Implementieren Sie für die Klasse *MotionDetection* folgende Methoden:

- `void drawLine (int x1, int y1, int x2, int y2, unsigned int color, Image &img);`
- `double blockDifference (const Image &img1, int x1, int y1, const Image &img2, int x2, int y2);`
- `void getMotionVectors (const Image &img, const Image &img2, vector<MotionVector> &mv);`
- `drawMotionVectors (Image &img1, vector<MotionVector> &mv);`
- `void getAverageMotionVector (const vector<MotionVector> &mv, MotionVector &avgVector);`

- 1.) Die Methode *drawLine* soll in ein Bild eine Gerade von (x1/y1) nach (x2/y2) zeichnen. Überlegen Sie sich ein geeignetes Verfahren, um die Gerade zu zeichnen. Betrachten Sie zunächst den einfachen Fall, für den gilt:  $x1=x2$  bzw.  $y1=y2$ . Entwickeln Sie anschließend ein Verfahren, um für beliebige Positionen eine Gerade in ein Bild einzuzichnen.

*Aufgabe:* Laden Sie das Bild *HistTest1.ppm*, zeichnen Sie folgende Linien und speichern Sie das Bild ab:

- `drawLine ( 80,100,230,100, 255, img);`
- `drawLine (195,145,115,145, 255, img);`
- `drawLine ( 80,100,115,145, 255, img);`
- `drawLine (195,145,230,100, 255, img);`
- `drawLine (155,120,155, 80, 0, img);`
- `drawLine (155, 40,155, 80, 0, img);`
- `drawLine (190, 80,155, 40, 0, img);`
- `drawLine (190, 80,155, 85, 0, img);`

- 2.) Implementieren Sie die Methode *blockDifference*, in der zwei Bildbereiche (Blöcke) mit einer Größe von  $16 \times 16$  Pixeln verglichen und die Summe der absoluten Pixeldifferenzen über alle Kanäle der beiden Blöcke berechnet wird. Anschließend normieren Sie die Differenz auf die Anzahl der betrachteten Pixel und Kanäle. (x1/y1) definiert die linke obere Ecke des Blockes im ersten Bild, (x2/y2) die linke obere Ecke im zweiten Bild.

(Fortsetzung Seite 2)

- 3.) Die Methode *getMotionVectors* soll mehrere Bewegungsvektoren für zwei Bilder berechnen. Die Vektoren werden in einem Vektor vom Typ *MotionVector* gespeichert, der folgende Struktur besitzt:

```
struct MotionVector {
    int px;          // horizontal start position of motion vector
    int py;          // vertical start position of motion vector
    double mx;       // horizontal length of motion vector
    double my;       // vertical length of motion vector
    double diff;     // difference of the blocks
};
```

Für alle Bildpositionen (px/py) mit  $px=\{0,16,32,\dots\}$  und  $py=\{0,16,32,\dots\}$  soll ein Bewegungsvektor ermittelt werden, der in *mv* gespeichert wird. Der Block im ersten Bild an der Position (px/py) wird mit allen Blöcken innerhalb einer Umgebung um den Punkt (px+mx/py+my) im zweiten Bild verglichen. Eine Umgebung von 16 Pixeln soll betrachtet werden, d.h. mx und my können Werte im Intervall [-16;16] annehmen.

Verwenden Sie die Funktion *blockDifference (img1,px,py,img2,px+mx,py+my)*, um die Differenz zweier Blöcke zu berechnen. Der Block mit der minimalen Differenz definiert den Bewegungsvektor (mx/my).

*Aufgabe:* Wie viele Additionen und Multiplikationen werden zur Berechnung der Bewegungsvektoren zwischen zwei Bildern ungefähr benötigt? Laden Sie die Videosequenz *sequenz3.ppm* und berechnen Sie für jeweils zwei benachbarte Bilder die Bewegungsvektoren.

- 4.) Die Methode *drawMotionVectors* soll für jeden Bewegungsvektor eine weiße Linie in das erste Bild von der Startposition (px/py) zur Endposition (px+mx/ py+my) zeichnen. Um Start- und Endpunkt unterscheiden zu können, soll auf dem Startpunkt ein schwarzes Pixel gesetzt werden.

*Aufgabe:* Laden Sie die Videosequenz *sequenz3.ppm*, berechnen Sie für jeweils zwei benachbarte Bilder die Bewegungsvektoren, zeichnen Sie die Vektoren in das erste der beiden Bilder ein und speichern Sie die geänderten Bilder ab.

- 5.) In der Methode *getAverageMotionVector* soll ein durchschnittlicher Bewegungsvektor für zwei aufeinander folgende Bilder eines Videos berechnet werden. Für die zuvor durch *getMotionVectors* ermittelten Bewegungsvektoren wird der durchschnittliche Vektor berechnet.

*Aufgabe:* Welche Information liefert der durchschnittliche Bewegungsvektor im optimalen Fall? Geben Sie die durchschnittliche horizontale und vertikale Bewegung für die einzelnen Bilder der Sequenz *sequenz3.ppm* aus und zeigen Sie die Werte graphisch an.