

**Computer-Grafik+Animation CS19-2 (Informatik), März 2022 (90 Min)**

Comp-Grafik, Animate Color, Motion, Path, Transform, Effects, 3D-2D, Timing

**Frage 1)**

Erläutern Sie kurz die folgenden sechs Begriffe sowie deren Funktionsweise oder Verfahren.

(12 Punkte)

**Animate Motion :**



**Computer Graphics :**



**ITU.int, BT.2020 :**



**Kumulatives Histogramm :**



**Point Cloud :**



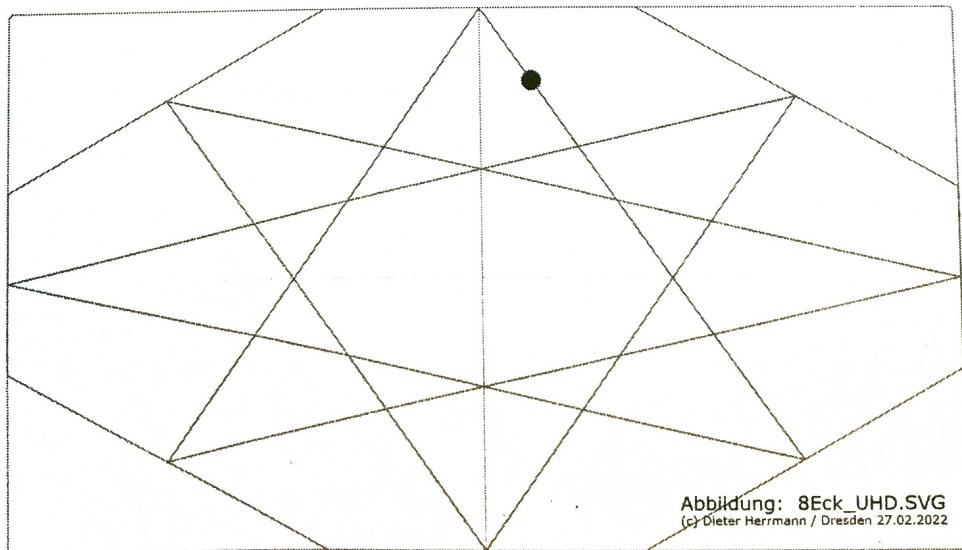
**YCbCr :**



## Frage 2)

Ein Motion Path soll im UHD Bildschirm 3840x2160 Pixel @ 16:9 ein Stern Polygon mit den Punkten P1-4-7-2-5-8-3-6-1 periodisch durchlaufen. Der UHD Nullpunkt ("canvas") ist links oben, die x Achse nach rechts und die y Achsen nach unten. Auf jeder UHD Seite wird der Bereich von 1/3 bis 2/3 rot markiert. Mt 4 weiteren Schrägen werden diese 4 Abschnitte zu einem "8-Eck\_rot" verbunden und begrenzen den Bereich der Animation. Die Mittelpunkte der 8 Polygon Seiten bilden dann einen 8-Eck Stern beginnend bei P1(1920, 0) im Uhrzeigensinn. P3 ist in der Mitte rechts, P5 in der Mitte unten, und P7 in der Mitte links. Tragen Sie im Polygon "8-Stern\_gruen" die 8 points x,y in der Reihenfolge der Animation P1-4-7-2-5-8-3-6-1 ein. Der Kreis "Flying\_Spot" erfolgt mit animateTransform als relative Translation beginnend bei P1(1920, 0) mit den values 0,0. Die Animation endet ebenso mit values 0,0 nach einer vollen Periode. Tragen Sie die anderen 7 aufeinander erfolgenden Verschiebungen x,y dazwischen ein. (12 Punkte)

```
<svg id="8-Eck_UHD" width="100%" height="100%"  
      viewBox="-20 -20 3880 2200" . . . >  
<title> Motion Path 8-Eck Stern in UHD 3840x2160px </title>  
<rect id="canvas" x="0" y="0" width="3840" height="2160" . . . />  
<line id="horiz_x" x1="840" y1="1080" x2="3000" y2="1080" . . . />  
<line id="verti_y" x1="1920" y1="0" x2="1920" y2="2160" . . . />  
  
<polygon id="8-Eck_rot"  
      points=" 2560,0    3840,720    3840,1440    2560,2160  
              1280,2160    0,1440      0,720      1280,0    " . . . />  
<polygon id="8-Stern_gruen" style="stroke:rgb(0,200,0);"  
      points=" 1920,0      _____'_____      _____'_____      _____'_____  
              _____'_____      _____'_____      _____'_____      _____'_____ " . . . />  
  
<circle id="Flying_Spot" cx="1920" cy="0" r="40" . . . />  
<animateTransform id="Clockwise_P1-4-7-2-5-8-3-6-1"  
type="translate" begin="1s" dur="9s" repeatCount="indefinite"  
values=" 0,0 ; _____,_____ ; _____,_____ ; _____,_____ ;  
        _____,_____ ; _____,_____ ; _____,_____ ; _____,_____ ;  
        0,0 " /> </circle> </svg>
```



### Frage 3)

Bei Bildstabilisatoren, Objekt-Suche und Verfolgung, Gesichts-Erkennung oder Personen-Identifikation, in VR/AR bis zum Predictive Modeling wird Pattern-Matching (Block-Matching) eingesetzt. In der MPEG Video-Kompression mit Motion-Compensation MC oder MPEG-7 zur visuellen Beschreibung (Visual Description) von Merkmalen wie Farben, Texturen, Kanten, Formen, Bewegungen und dergleichen wird ebenso Block-Matching genutzt.  
 Ein Pixel-Raster aus 4 Bildpunkten wie ein Symbol Y, Linker und Rechter Arm oben, Sternpunkt mittig und darunter Mittelfuß, mit Objekt [1 1 1 1] bewegt sich relativ vom I-Frame 1 zu P-Frame 2. Im Block-Matching Y im Bereich [3\*3] Pixel sind jeweils die 4 Einzel-Differenzen  $di=f(Frame\ 2 - MC\ Frame\ 1)$  in allen 9 Richtungen um die 9 Fadenkreuze einzutragen. Aus den jeweils 4 Einzel-Differenzen ist unten die Summe der Differenz-Quadrate zu bilden. Abschließend sind die Abweichungen in den 9 Richtungen nach einem Ranking zu sortieren von 1 für Minimal bis 9 für Maximal. Eventuelle Gleichrangigkeit wird entsprechend Abarbeitung ausgeschlossen. Die minimale Abweichung 1 entspricht damit dem gefundenen und wahrscheinlichsten Bewegungs-Vektor.

I-Frame 1 Motion Frame 2 Motion Estimation - d[2x2]  
 $di=f(Frame\ 2 - MC\ Frame\ 1)$

1   2   3   4   5	2   2   2   4   6	1   *   1	*	*
2   <u>1</u>   4   <u>1</u>   6	2   2   4   6   6	4		
3   4   <u>1</u>   6   7	3   5   0   6   0	*	*	*
4   5   <u>1</u>   7   8	5   5   5   2   7	-+ + -	-+ + -	-+ + -
5   6   7   8   9	5   7   6   1   9	*	*	*

Block-Summe (Diff) <sup>2</sup> Ranking Min-Max 1...9

#### **Frage 4)**

In visuellen Systemen (Grafik, Foto, Video, HDTV, UHD) werden Testbilder, Farb-Differenzen YCbCr(RGB) oder Farb-Grauwert Transformationen Y(RGB) eingesetzt. Gegeben sind die 8 Farben additiv RGBW und subtraktiv CMYK in einer tri-chromatischen Test-Impulsfolge mit 3 bit (blue, red, green) zur additiven Erzeugung von 8 Farb-Balken (RGB) und 8 Luminanz-Stufen Y. Berechnen Sie die positiven wie negativen Farb-Differenzen B-Y und R-Y für UHD TV (Fast und vereinfacht). Skalieren Sie die Werte wie in der HD Fernsehtechnik auf Cb und Cr jeweils im Bereich von -0,5 bis +0,5. Tragen Sie die 6 Farb-Differenz-Koordinaten Cb und Cr von (B R M G C Y) in das CbCr-Diagramm als Farb-Sechseck ein. (12 Punkte)

(12 Punkte)

R+1 |  
R-2 |

G+1 |  
G-2 |

Luminanz Y(Fast) = f(R, G, B) = 0,25 \* R + 0,50 \* G + 0,25 \* B

$x = 1.00001$ ,  $y = 1.00001$

**D. Y.** 12.00001 2. 1. 2. 1. 2. 1. 2. 1. 2. 12.00001

B-1 | 0.0000 | 0. | 0. | 0. | 0. | 0. | 0.0000 |

R-Y | 0.0000 | 0. | 0. | 0. | 0. | 0. | 0. | 0.0000 |

---

Normieren auf -0,5 bis 0,5: Cb=0, \_\_\_\_\_ (B-Y) ; Cr=0, \_\_\_\_\_ (R-Y)

+-----+-----+-----+-----+-----+-----+-----+-----+

Cb | 0.0000 | 0. | 0. | 0. | 0. | 0. | 0. | 0. | 0.0000 |  
 +---K---+---B---+---R---+---M---+---G---+---C---+---Y---+---W---+

```
Cr | 0.0000 | 0.    | 0.    | 0.    | 0.    | 0.    | 0.    | 0.0000
```

---

**Cx 0.5** +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4

A horizontal number line representing the interval [-1, 1]. The line is marked with vertical tick marks at intervals of 0.1. The origin, where the line begins, is explicitly labeled with the value '0'. The tick marks are labeled with their corresponding decimal values: -1, -0.9, -0.8, -0.7, -0.6, -0.5, -0.4, -0.3, -0.2, -0.1, 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1.

-0.1 + -- + -- + -- + -- + -- + -- + -- + -- + -- + -- + -- + ..

-0.2 + -- + -- + -- + -- + -- + -- + -- + -- + -- + -- + -- + -- +

-0.3 + -- + -- + -- + -- + -- + -- + -- + -- + -- + -- + -- + -- + -- +

-0.4 + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - +

A horizontal number line starting at -9 and ending at 11. There are 11 tick marks on the line, including the endpoints. The points are labeled with their corresponding integer values: -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11. A dashed vertical line extends from the point labeled 5 upwards, indicating the value of  $t$ .

Berufssakademie Sachsen; Studienbereich Informatik; Klausur Teil I  
Fragestellung und Fragen; Dozent Dieter Herrmann, Dresden

14.03.2022

4.03.2022;

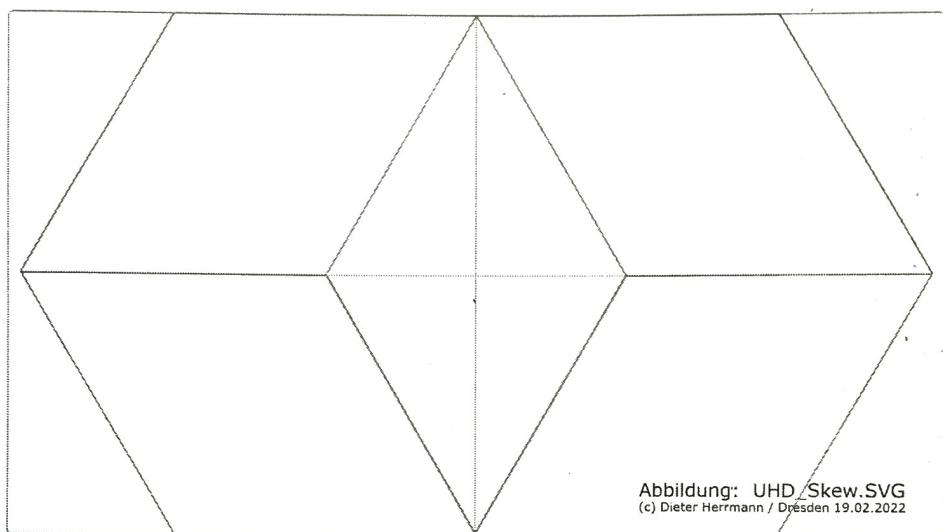
### **Frage 5)**

Gegeben ist ein quadratisches Test-Bild in 4K mit 4000x4000 Pixeln zum Composing auf einem 8K UHD Monitor (Super High Vision - SHV) mit 7680x4320 Pixel in 16 : 9. Mittels Affiner Transformationen sollen 5 identische Rhomben entsprechend Layout erzeugt und positioniert werden. Im Underscan füllen die 5 Rhomben den Monitor von oben bis unten, berühren aber nicht den linken und rechten Bildschirmrand. Die Verankerung der Rhomben ist jeweils oben links mit aufrechter Orientierung des Bildes nach oben entsprechend Abbildung.

Für den in der Mittelsenkrechten stehenden Rhombus sind die einzelnen 4 notwendigen Transformationen zu bestimmen.

1. Die verzerrte Skalierung im Null-Punkt ; 2. SkewX zum Rhombus ; 3. Die Translation der rechten Rhombus-Spitze zum oberen Mittelpunkt (3840, 0) ; und 4. die Rotation um diesen Punkt  $cx="3840"$   $cy="0"$  in die Mittelsenkrechte auf dem 8K Monitor.

Bestimmen Sie unten die ersten 3 einzelnen Affinen Transformationen entsprechend SVG+XML in 2D mit  $transform="matrix( a \ b \ c \ d \ e \ f )"$  und die letzte 4. mit Rotate Winkel. (12 Punkte)

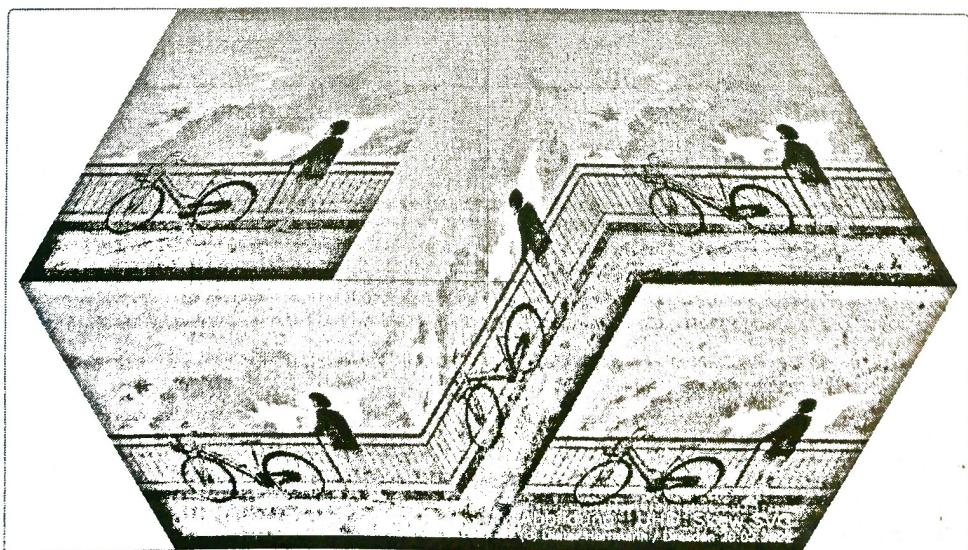


[1] Scale :  $S_x = \underline{\hspace{2cm}}$   $S_y = \underline{\hspace{2cm}}$   $transform="matrix(\underline{\hspace{2cm}} \ 0 \ 0 \ \underline{\hspace{2cm}} \ 0 \ 0 )"$

[2] SkewX :  $X = \underline{\hspace{2cm}}^\circ$   $transform="matrix(1 \ 0 \ \underline{\hspace{2cm}} \ 1 \ 0 \ 0 )"$

[3] Translate :  $T_x = \underline{\hspace{2cm}}$   $transform="matrix(1 \ 0 \ 0 \ 1 \ \underline{\hspace{2cm}} \ 0 )"$

[4] Rotate  $cx="3840"$   $cy="0"$  :  $transform="rotate(\underline{\hspace{2cm}}, 3840 \ 0 )"$



### Frage 6)

Eine alltägliche Anwendung in allen visuellen Systemen (CRT, TFT, DSC, DVC, TV, Video, Disc, Print, UHD, HDR-TV, VR, AR) ist die Anpassung der Signal-Übertragungskennlinie durch Gamma-Korrektur oder eine adaptive Transfer-Funktion.

( Applying gamma correction means that each of the three R, G and B must be converted to  $R' = R^{\text{gamma}}$ ,  $G' = G^{\text{gamma}}$ ,  $B' = B^{\text{gamma}}$ , before handing to the Operating System. This may rapidly be done by building a 10 bit HDR 1024-element Look-Up Table - LUT. )

Gamma=1,0 entspricht einer linearen Übertragung ohne Korrektur, Signal-Ausgang gleich Signal-Eingang. Die Look-Up Table LUT in 10 bit HDR mit 1024 Elementen soll adaptiv vom kumulativen Histogramm abgeleitet werden. Die diskrete Histogramm-Funktion ist mit 11 absoluten Häufigkeiten von 0 bis 10 mit  $H_i$  gegeben. Berechnen Sie das kumulative Histogramm  $cH$  durch fortlaufendes Aufsummieren von  $H_i$ . Mit dem Maximal-Wert von 100 % werden alle anderen kumulierten Summen normiert zu relativen Histogramm-Werten  $rH$  zwischen 0.0 und 1.0. Schließlich erfolgt dann die Rücknormierung auf 10 bit Ausgabe mit  $Y_2$  von 0 bis 1023. Berechnen Sie die absoluten und relativen Koordinaten der Transfer-Curve und skizzieren Sie diese im Diagramm mit 10 bit Skala. (12 Punkte)

Histogramm -  $H_i$ , kumulierte Histogramm -  $cH$ :

$H_i$	0	80	30	10	20	50	60	50	40	70	90
$cH$	0										
---	---	---	---	---	---	---	---	---	---	---	---

Gamma = 1.0 : [  $Y_1 = f(X_1)$ , Kurve 1 ]

$X_1$	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$G_1$	0.0	.10	.20	.30	.40	.50	.60	.70	.80	.90	1.0
$Y_1$	000	102	205	307	409	512	614	716	818	921	1023

HDR adaptive Transfer-Curve, LUT - Look-Up Table [ 10 bit von dezimal 000 bis 1023 ]

$X_1$	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$rH$	0.0	.	.	.	.	.	.	.	.	.	1.0
$Y_2$	000										1023
---	---	---	---	---	---	---	---	---	---	---	---

Output  $y=f(x)$

