1. **Info**

* Wirkliche Empfehlung : TAFELWERK Wissen (7. bis 12. Klasse) einsatzbereit !
* Funktionen: linear, exponentiell, Wurzel, log, sin, cos, tan, und arc sin cos tan
* Einheits-Kreis, Koordinaten, kartesisch, polar, Winkel, Degrees DEG und RAD; Pythagoras, Kreis-Gleichung r2 = x2 + y2, Kugel-Gleichung r2 = x2 + y2 + z2; Berechnungen und Umrechnungen (Transformationen)
* Ebene 2D Figuren, Dreieck, Viereck, Parallelogram, Polygon, Kreis, Ellipse, 2D Grafik, mit englischen Bezeichnungen, Elementen, (SVG) ;
* Körper 3D, Würfel, Quader, Pyramide, Prisma, Polyeder, Zylinder, Kegel, Doppel-Kegel, Kugel, 3D Grafik, mit englischen Bezeichnungen, Elementen, (X3D) ;
* Zahlen-Systeme, dezimal, dual, hexadezimal, 8 bit, 10 bit, 12 bit, 16 bit, je Kanal, Dimension, Koordinate, Position, Farbe, absolute Koordinaten (Pixel, Voxel, Farben, Graustufen) und relative Werte oder Koordinaten (0.0 bis 1.0 ; 0% bis 100%), Berechnungen und Umrechnungen, (Transformationen)
* Beispielsweise 6 Aufgabe aus folgenden Komplexen :
  1. Begriffe ;
  2. 1D Transfer Function, y=f(x) mit Parametern (m, n, Gamma) ;
  3. 2D Affine Transformationen, svg+xml, (rotate, scale, translate, skew, matrix)
  4. 3D Farbraum Transformationen, rgba, hsla, ycbcr (JPEG, MPEG) ;
  5. Animationen, Time-Sheets
  6. 3D, Virtual Reality, x3d+xml, x3dom
  7. Pattern Matching

1. **Begriffe**

* HDR, LDR
  + High Dynamic Range vs. Low Dynamic Range bzw. vs. Standard Dynamic Range
  + Dynamikumfang wird aus dem Verhältnis zwischen maximalem und minimalem Signal im Messbereich bestimmt
  + Analoge Werte oder digitale Repräsentationen (Bit);
  + Anwendung auf bspw. Lautstärke, Helligkeit, Kontrast, Farbtiefe, …
  + HDR = große Helligkeitsunterschiede
    - 10.000 nits / 0,001 nits = 10.000.000 : 1
  + SDR < 1000 : 1
  + Nits = Candela pro m^2 = cd/(m^2)
* Motion Path, Time Sheet
  + Motion Path (Animation): Bewegung eines Objektes über einen vordefinierten Pfad über eine bestimmte Zeit hinweg
  + Time Sheet ?
* RGB Color Cube:
  + Dreidimensionale Repräsentation des RGB-Farbraums
* Scalability:
  + Deutsch: Skalierbarkeit
  + Berechnung eines Bildes bei Vergrößerung anhand der Bildinformationen
  + Bei SVG verlustfrei, bei PNG, JPEG, o.ä. nicht verlustfrei
* Transfer Function, Lin, Log
  + Übertragungsfunktion: Mathematische Funktion, welche das Ausgabesignal einer Komponente (elektronisches System, Steuerungselement, …) für jedes mögliche Eingabesignal modelliert.
  + Ausgangssignal kann als lineare Differentialgleichung dargestellt werden: lineares System
  + Log ?
* X3D+XML, X3DOM
  + X3D: eine auf XML basierende Beschreibungssprache für 3D-Modelle, die in einem Webbrowser angezeigt werden können
  + X3DOM: a framework for integrating and manipulating (X)3D scenes as HTML5 DOM elements, which are rendered via WebGL without additional plugins

1. **1D Transfer Function**

Zur Transfer Function 1D - Übertragungs-Funktion.

BA01210L \_pdf \_Seite 8-8 \_Frage 6 \_Transfer Function HDR \_Lösung \_0675x1080px \_04bpp.png ;

Da ich es weitgehend auf wiederverwendbare Mathematik zurückführe, geht es schlicht um eine Funktion wie y = f(x) = x' = m \* x (hoch Gamma) + n.

Ist der Exponent Gamma = 1 wird daraus eine lineare Gleichung y = mx + n.

n ist bekanntermaßen die Nullpunkts-Verschiebung auf der y Achse, auch Versatz, Offset, Bias oder Verschiebung und Translation je nach Einsatzgebiet genannt. Die meisten Systeme werden Nullpunkt korrigiert, Schwarz Pegel.

Ist dann auch n = 0, bleibt nur noch y = m \* x.

In m (Maßstab, Magnification) steckt gleichzeitig die Skalierung und die Rotation, also Anstieg, Anstiegswinkel, Zoom-In/-Out, Scale, Gain, Verstärkung. Nach dem Schwarz-Pegel mit n (Helligkeit) wird mit m der Kontrast auf weiß geregelt.

In einem Digitalen Signal Prozessor (DSP) läuft das Ganze meist ohne Zwischenspeicherung in Echtzeit, beziehungsweise weitgehend ohne störende Verzögerung. AGC - Automatic Gain Control - Automatische Verstärkungs-Regelung.

Die Transfer Function können vorzugsweise relativ zwischen 0.0 und 1.0 oder rückskaliert in jeweils absoluten Werten gerechnet werden.

Eine neutrale normierte lineare Funktion ergibt dann die Diagonale y=f(x) zwischen 0 und 1, mit n=0.0, m=1.0 und Gamma=1.0 .

Und y(x=0.5) = 0.5 .

Bei Nicht-linearen Funktionen mit Gamma ungleich 1 liegt die Kurve bei einer Wurzel-Funktion über der Diagonalen, und bei einer Exponential-Funktion unter der Diagonalen. y (x=0.5) = 0.25 | Gamma = 2.0 .

Mit 10 Stützstellen im 10 % Abstand lässt sich der Verlauf der Krümmung schon gut verdeutlichen.

Neben den relativen Achsen kann man nun auf absolute Werte zurück skalieren je nach bit Tiefe, und egal in welcher physikalischen Größe, Position, Pixel, Voxel, Farbe, HDR, Lautstärke, Zeit, usw.

Werte Tabelle und entsprechender Graph der Transfer Function sollten dann übereinstimmen und plausibel sein.

1. **2D Affine Transformationen**

Zur Affinen Transformation 2D - Selbstähnliche Abbildung.

BA01210L \_pdf \_Seite 8-7 \_Frage 5 \_Affine Transformation 2D (a b c d + e f) \_Lösung \_0669x1080px \_08bpp.png ;

Vereinfacht ist das analog zur Funktion oben, nur Y = M \* X + T .

Nur haben wir jetzt 2 Eingangs-Größen x und y, sowie 2 Ausgangs-Größen x' und y' .

Sind Eingang und Matrix gegeben, können wir den Ausgang berechnen. Umgekehrt gilt auch, haben wir Eingangs- und Ausgangs-System gegeben, wird eine passende Matrix zur selbstähnlichen Abbildung gesucht, wie Digital Foto auf Display.

Wenn keine Scherung (Skew) oder Drehung (Rotation) vorliegt, können wir jede Dimension einzeln zuordnen.

x' = Sx \* x + Tx ;

y' = Sy \* y + Ty .

Wenn wir die Eingangs-Fläche Nullpunkt korrigieren, bleiben folgende Funktionen :

x' = Sx \* x | Tx=0 ;

y' = Sy \* y | Ty=0 .

S = Ausgang / Eingang.

Sx = x' / x ;

Sy = y' / y ;

Wenn nicht verzerrt werden soll, muss Sx = Sy sein, um das Seitenverhältnis zu erhalten.

Es gibt von vielen Möglichkeiten des Zoomens 2 Vorzugsvarianten, Einpassen (fit in, oder Underscan) mit S = Sin = Min (Sx | Sy), oder vollständig Ausfüllen und Überstehendes wird abgeschnitten oder ist gar nicht mehr sichtbar (fill out, oder Overscan), S = Sout = Max (Sx | Sy).

In der Aufgabe war Einpassen gefordert, also die kleinere Skalierung.

Sx = x' / x = 7680 / 13500 = 0.56889

Sy = y' / y = 4320 / 9000 = 0.48000 .

S = Sin = Min (Sx | Sy) = Min ( 0.56889 | 0.48000 ) = 0.48000 .

Rück-Probe :

x' = S \* x = 0.48000 \* 13500 = 6480 ; Tx = ( width - x' ) / 2 = ( 7680 - 6480 ) / 2 = 1200 / 2 = 600 ;

y' = S \* y = 0.48000 \* 9000 = 4320 ; Ty = ( heigth - y' ) / 2 = ( 4320 - 4320 ) / 2 = 0 / 2 = 0 ;

transform="matrix(a b c d e f)" = transform="matrix(Sx 0 0 Sy Tx Ty)" = transform="matrix(0.48000 0 0 0.48000 600 0)"

1. **3D Farbraum-Transformationen**

Noch ein Hinweis - Color Look Up Table für Farbraum Transformation (3D), am Beispiel RGB(A) - HSL(A).

Um Rechenoperationen zu beschleunigen, nutzt man Look Up Table, pre-calculated Values aus einem Array oder Register in Echtzeit abrufbar.

Bei RGBA zu HSLA Farbraum Transformation sucht man nun Stützstellen, die ohne Rundungsfehler zueinander passen.

Kleinstes gemeinsame Vielfache durch Primfaktoren Zerlegung :

Der RGB Cube hat 24 bit Farben, auf den 6 Farb-Kanten liegen bei 8 bit nur 1530 Farben, und auf einer linearen Kante nur noch 256 Werte von 0 bis 255.

Beim HSL Farbkreis sind es 360 Faben auf 360°, und 60 Hue Töne von 0° bis 60°, und in jedem anderen Sektor von 60°.

RGB : 255 = 17 \* 5 \* 3 = 5 \* 51 ;

( Hue : 360 = 6 \* 6 \* 5 \* 2 ) ;

Hue : 60 = 6 \* 5 \* 2 = 5 \* 12 ;

Die Primzahl 5 steckt in beiden und ist gut nutzbar.

Delta RGB = 51 : 0, 51, 102, 153, 204, 255 ;

Delta Hue = 12° : 0°, 12°, 24°, 36°, 48°, 60° ;

Da wir 6 Kanten und 6 Sektoren und jeweils 5 Werte darin haben, kann man für die insgesamt 30 Werte eine einfache vorausberechnete Color Look Up Table RGB - HLS erstellen.

Mit Null als Ausgangspunkt haben wir je 6 Werte, die im RGB Color Cube mit 6 Rot \* 6 Grün \* 6 Blau = 216 auch die Web Save Color Farb Palette ergeben.

Mit einer kleinen Zuordnungs-Tabelle der 30 Werte kann man sich Eingabe Fehler ersparen, bei Zwischenwerten einfach interpolieren, oder berechnete Ergebnisse vergleichen und überprüfen.

1. **Animationen, Time-Sheets**

* Die Farbkanten des RGB Würfels sowie der Umfang des HSL Farbkreises sind ein geschlossener Linienzug von Rot zu Rot, R-Y-G-C-B-M-R.
* In HSL sind das 6 Sektoren je 60 Grad linear von 0 Grad bis 360 Grad. Und in RGB sind das 6 Kanten je 255 weitere Farben von insgesamt linear von 0 bis 1530. Beide 1D Linien von 0% bis 100% lassen sich nun über einfachste Verhältnisgleichungen mappen :
* hue(circle) / 360 ° = rgb(cube) / 1530. Da Hue nur 360 Integer hat ist es einfacher aber gleichzeitig völlig verlustreich, weil von RGB 1530-360 = 1170 Farben im Farbkreis verloren gehen und gerade 360 Hue übrig bleiben.
* Man kann das Ganze auch Segmentweise berechnen, mit Modulo 60 ° beziehungsweise Modulo 255. Das lineare Verhältnis wird dann eine einfache Interpolation hue(sector) / 60 ° = rgb(edge) / 255. Beispielsweise bleiben 20 % in beiden Welten zugeordnet, Drehung 12° entsprechen Verschiebung 51 auf der Würfelkante.
* Die Schrittweite bei 7 Stützstellen ist einfach delta(hue) = 360°/7 = 51,429 Grad oder delta(rgb) = 1530 / 7 = 218,571 RGB Farben. Obwohl im photometrischen Hue Winkel nur Integer gelten, ist es aber bei hohen Auflösungen mit mehreren 1000 Pixeln und geometrischen Rotationen empfehlenswert, auch auf 1/10.000 genau zu drehen, damit grobe Rundungen nicht zu Fehlersprungstellen führen.
* Für y = m \* x + n brauchen wir jetzt nur noch einen Startwert oder Offset n als Nullpunkts-Verschiebung.
* Die Folge 1 bis 7 kann man in SVG im Uhrzeiger Sinn (CW - Clockwise) zählen, von Rot auf der x-Achse über Magenta zu Blau, und so fort.
* Ebenso geht es in Hue Drehrichtung von Rot auf der x-Achse entgegen Uhrzeiger (CCW - Counter Clockwise) über Yellow zu Grün, und so fort. Da 1 Punkt genau auf -y liegt, kann man auch von Nord in CW nach rechts, oder in CCW nach links zählen.
* Hier soll Hue in CW genutzt werden, 1 Orange bis 7 Rot. Der Offset beträgt 90 Grad - delta hue.
  + Hue(0) = 360 °/4 - 51,429 ° = 90 ° - 51,429 ° = 38,571 ° ;
  + Hue(i) = i \* 51,429 ° + 38,571 ° ;
  + Hue(0) = 0 \* 51,429 ° + 38,571 ° = 38,571 ° ~~ 39°;
  + Hue(1) = 1 \* 51,429 ° + 38,571 ° = 90,000 ° ;
  + Hue(2) = 2 \* 51,429 ° + 38,571 ° ;
  + Hue(3) = 3 \* 51,429 ° + 38,571 ° ;
  + Hue(4) = 4 \* 51,429 ° + 38,571 ° ;
  + Hue(5) = 5 \* 51,429 ° + 38,571 ° ;
  + Hue(6) = 6 \* 51,429 ° + 38,571 ° = 347,145 ° ~~ 347 ° ;
  + Hue(7) = 7 \* 51,429 ° + 38,571 ° = 398,574 °; 398,574 ° - 360 ° =~ 38,574 ° ~~ 39° ; Kontrolle der Periodizität.
* Setzt man nun die hsl(h,s,l) Kodierung und den identischen Rotationswinkel ein, erhält man den richtigen Hue Wert auch gleich in der richtigen Position.
* Der Farb-Komet wird mit cx und cy in die Umlaufbahn gebracht und mit negativem Winkel rotate(-hue, 1920, 1080) um den Nullpunkt cx und cy des Umkreises gedreht.

#

* Identisch und linear sowie auch noch genauer kann man in RGB rechnen.
  + rgb(0) = 1530/4 - 218,571 = 382,500 - 218,571 = 163,929 ~~ 164 ;
  + rgb(i) = i \* 218,571 + 163,929 ;
  + r-y
  + rgb(0) = 0 \* 218,571 + 163,929 = 163,929 ; 163,929 mod 255 = 163,929 ~~ 164 ;
  + y-g
  + rgb(1) = 1 \* 218,571 + 163,929 = 382,500 ; 382,500 mod 255 = 127,500 ~~ 127 ;
  + y-g
  + rgb(2) = 2 \* 218,571 + 163,929 ;
  + g-c
  + rgb(3) = 3 \* 218,571 + 163,929 ;
  + c-b
  + rgb(4) = 4 \* 218,571 + 163,929 ;
  + b-m
  + rgb(5) = 5 \* 218,571 + 163,929 ;
  + m-r
  + rgb(6) = 6 \* 218,571 + 163,929 = 1475,355 ; 1475,355 mod 255 = 200,355 ~~ 200 ;
  + r-y
  + rgb(7) = 7 \* 218,571 + 163,929 = 1693,926 ; 1693,926 - 1530 = 163,926 ~~ 164 ;
  + 1693,926 mod 255 = 163,926 ~~ 164 ; Kontrolle der Periodizität.
* Wie bei Hue ändert sich bei RGB linear jeweils nur ein einzelner Werte, allerdings in wechselnden Farb-Komponenten und Vorzeichen. Der lineare Farb-Vektor im RGB Color Cube ergibt sich schlicht wie folgt :
* R / Y \ G / C \ B / M \ R .
* Von Red auf zu Yellow, runter zu Green, auf zu Cyan, runter zu Blue, auf zu Magenta bis runter wieder zu Red.
* Also von 0 % Red bis 100 % Red : R + G - R + B - G + R - B = R .
* Bei den Farben auf den Würfel Kanten ist immer eine Komponente genau NULL, eine andere immer MAX und nur eine Größe folgt vom Offset mit fortlaufenden Schrittweiten.
* Da wir bei Rot mit rgb(255,0,0) beginnen, und der Offset 164 beträgt, wird zunächst Grün um 164 erhöht.
* R
* rgb(255,164,0) ;
* Y
* rgb(127,255,0) ;

[…]

* y = m \* x + n oder hue(sector) / 60 ° = rgb(edge) / 255 ist ebenso einfach wie etwa DEG in RAD umrechnen.
* Im Vergleich zu w3schools.com unterscheiden sich die Werte um 1 oder 2 dezimale Farben, zum Beispiel je nach Rundung wie 127,5.

Meine Lösung ist noch etwas erweitert. Sie zeigt die Farb-Eckpunkte in HSL Kreisen und RGB Quadraten auch statisch. Auch die Geometrie Positionen sind sowohl kartesisch (Polygon) und polar (Rotation) implementiert mit doppelter Sicherheit zum Vergleich. Die 3 Animationen (rgb, cx, cy) sind in Values zusammengefasst.

Ich hoffe, ich habe jetzt nicht zu viele Schreibfehler drin und die Hinweise helfen, so dass jeder die Beispiele zum Laufen bringt und versteht. Sie sollten die Schulkenntnisse von kartesischen, polaren Koordinaten und Transformationen einschließlich Kreis-Gleichung r2 = x2 + y2 bis Kugel-Gleichung r2 = x2 + y2 + z2 sicher anwenden können. Wenn es weitere Fragen gibt, müssen wir das noch (nicht nur wegen der Klausur) festigen und vertiefen. Weitere Lösungen folgen. Gutes Gelingen.

1. **Pattern Matching**

Eine mögliche Klausur Aufgabe, die ergänzend zur Computer Grafik kurz erläutert werden soll.

Muster Erkennung vom Passwort, über QR Code bis Face ID, 3D Venen Scanner bis Spacherkennung, RADAR oder LIDAR sind heute Alltag.

Egal in welchen Media Arten oder in welcher Dimension (1D, 2D, 3D, Multi-D) und Komplexität es realisiert wird,

soll hier ein einfaches Beispiel gezeigt und auf Besonderheiten verwiesen werden.

Eine Video Sequenz oder ein 3D Objekt lässt sich durch Capture und Projektion auf eine 2D Fläche mit Farb/Grauwert Tiefe einfrieren. Diese Merkmalsreduktion kann man bis zu einer 1D Liste fortsetzen (Skalarer Merkmalsvektor), und dann über Schwellwerte zu einer Entscheidung führen.

Achtung ! Anders als bei strikten Passworten oder Binär Dateien, sind Medien in Raum und Zeit nie eindeutig oder vollständig identisch. Mikro-Bewegungen und Quantenrauschen (Abtastraster, zeitliche Quantisierung, thermisches Rauschen, elektronisches Rauschen, Digitalisierungsrauschen, Rechengenauigkeit, Rundungsfehler, und so weiter) lassen nur eine gewisse Näherung und Unschärfe zu. Die Identifikation wird schließlich binär über Schwellwerte getroffen, mit einer Aussage zur Wahrscheinlichkeit oder Fehlerabweichung.

Das wird eigentlich auch schon bei Schulexperimenten geübt, wenn etwa bei Temperaturmessungen die Messgenauigkeit durch mehrere Messungen erhöht werden soll, und man Mittelwert und Standard-Abweichung statistisch bestimmt.

-----------------------------------------

Schauen Sie sich zunächst ein vereinfachtes Beispiel an.

[ GreenBoard 05 \_Pattern Matching \_Motion Vector of Target 1 \_by simple Differencing \_1280x0720px\_64color\_08bpp.png ]

Es ist ein 3x3 Muster der Menge 1 oder zum Zeitpunkt t1. Die 1 in der Mitte des Rasters hat sich zum Zeitpunkt t2 nach links oben bewegt, und man sieh es scheinbar sofort. Reale Merkmale sind komplexer, auch wenn ein QR Code nur ein 2D bit Muster ist, wird es in der Kamera oder im Scanner doch zu einem verzerrten 3D Rauschmuster. Und die Maschine hat kein paralleles Erkennungsvorwissen. Es muss deshalb der gesamte Merkmalsraum abgescannt werden, mit Roher Gewalt (Brute Force). Da es hier 9 mögliche Positionen gibt, werden also auch alle 9 Möglichkeiten auf Übereinstimmung geprüft. Die Abweichung stellt einen Fehler dar, und kann dann nach Fehlergröße sortiert werden.

Zunächst einfache lineare Differenz delta = (t2 - t1) in den einzelnen Positionen.

Es gibt außerdem eine eindeutige und vollständige Abarbeitungs-Reihenfolge, hier von links oben nach rechts unten.

Hier sind vereinfacht auch alle Differenzen Nicht-Negativ, zwischen 0 und 7. Das lässt sich nun einfach sortieren von Min bis Max Fehler. Es wird eine Identität gefunden, und damit liegt der Bewegungsvektor mit der Differenz genau 0 nach links oben fest.

3 Besonderheiten :

In der Realität wird die Differenz kaum genau 0 sein.

Deshalb können auch Objekte gefunden werden, die scheinbar besser passen, (Kollateralschaden).

Außerden ist das Objekt und die Umgebung mit dem besagten Quanten- oder Quantisierungs-Rauschen (wenigstens Delta +/-1) überzogen, und deshalb von Situation zu Situation nicht identisch.

Es können durch die begrenzte Rechengenauigkeit und Echtzeitforderungen Mehrdeutigkeiten auftreten.

Hier Delta = 4 oben Mitte und unten Mitte. Eine Gleichrangigkeit gibt es bei eindeutigen Verfahren nicht, deshalb gilt die Abarbeitungsreihenfolge, oder zusätzlich festgelegte Prioritäten und Parameter zum Ranking.

Deshalb oben 5 und unten 6 !

Schließlich können Differenzen positiv und negativ sein, diese dürfen sich nicht gegenseitig kompensieren.

Ein vollständig nicht passendes Muster könnte sonst trotzdem auf den Gesamtfehler 0 kommen, (+3 -3 = 0), und zu fehlerhafter Identifikation führen.

-----------------------------------------

Beispiel Klausur Aufgabe :

[ BA01210L \_Seite 9-6 \_Frage 3 \_Pattern Matching \_Lösung \_0882x1080px \_08bpp.png ]

Im Video MPEG gibt es zur besseren Kompression so genannte Group of Pictures (GoP) mit Intra-Frames und Predictive Frames, also I-Frames und P-Frames. Im Datenstrom werden dann nur die differentiellen Änderungen übertragen, oder anders gesagt, die Schätzfehler beziehungsweise Muster-Abweichungen. Das entspricht auch dem Pattern Matching zum Beispiel für eine Bewegungs-Kompensation, Motion Compensation (MC).

Das gesuchte Objekt wird durch Muster-Vergleich gefunden. Hier ist es das Symbol Y aus [1 1 1 1].

Das Symbol kann nun in der näheren Umgebung in 9 Positionen vorkommen, von links oben, über zentriert bis rechts unten. In diesen 9 möglichen Positionen werden nun durch Überlagerung die Abweichungen als Differenzen ermittelt, positive ebenso wie negative.

Links oben entsprechend [2 2 2 5] - [1 1 1 1] = [1 1 1 4].

Bis rechts unten entsprechend [0 6 2 9] - [1 1 1 1] = [-1 5 1 8].

Damit sich positive und negative Differenzen nicht aufheben, werden die Differenzen quadriert zu auschließlich positiven Abweichungen. Diese Exponentielle Transfer Function (Übertragungs Funktion) hat noch eine weitere entscheidende Bedeutung. Identität bleibt erhalten, kleine Fehler bleiben klein und große Abweichungen werden sanktioniert. 0² = 0 ; 1² = 1 ; aber 9² = 81.

Die 4 einzelnen Y Fehler werden zu einer quadratischen Abweichung zusammengefasst, und durch exponentionelle Kantrastanhebung gleichzeitig gespreizt. Einzelabweichungen bisher nur -1 bia 8. Die Sortierreihenfolge reicht nun von minimal 12 bis maximal 91. Das Ranking lässt sich nun als Liste von 1 bis 9 gut sortieren.

Bei großen Datenmengen sind trotz exponentieller Spreizung Doppeldeutigkeiten und Mehrdeutigkeiten nicht immer zu vermeiden.

Hier tritt die Abweichung 60 zweimal auf.

Es ist bei eindeutigem Ranking nicht korrekt, wie bei einer Siegerehrung 2 gleiche Plätze zu vergeben.

Weder 7 und 7, noch 7/8 und 7/8 oder 8 / 8. Da keine weitere Sortierbedingungen und Priorätäten angegeben sind, bekommt in der Abarbeitungsreihenfolge die 60 in der ersten Zeile Rank 7 und die 60 in der zweiten Zeile den Rank 8.

-----------------------------------------

In der Klausur haben fast alle die Aufgabe vernünftig gelöst.

Allerdings haben nicht alle die gemachten Hinweise berücksichtigt, und dadurch ein paar Punkte verschenkt.

Die 12 Punkte dafür bedeuten ungefähr auch, dass Sie das in etwa 12 Minuten richtig berechnen können.

Wenn es Fragen dazu gibt, bitte ich um konkrete Rückmeldung.

Diese Methoden sind natürlich hoch automatisiert und laufen heute in Real Time und allen Medien Typen bis hin zu Autonomen Systemen.

Schauen Sie sich ergänzend dazu durchaus die Videos von :

- Sound of Football, ( https://www.youtube.com/watch?v=ADXOGrwiPok ) ;

- 3D Morphable Face Models, (3DMM), 1999, und nachfolgend ;

- VoCo Voice Converter ;

- Photo Style Transfer ;