

# CIE-Normfarbtafel

---

## CieView

```
export const CieView = (  
  {  
    signalYCBCR,  
    withOverlays = false,  
    encodedVideoStandard = 1,  
    encodedBitDepthIdx = 0  
  }) => {...}
```

**signalYCBCR:** siehe [Signal Arrays](#)

**withOverlays:** Buttons und Labels anzeigen (true/false)

**encodedVidStdIdx:** Index des Videostandards, dem signalYCBCR entspricht ([0..2] für Rec.601, 709 oder 2020)

**encodedBitDepthIdx:** Quantisierungsgrad, in dem signalYCBCR vorliegt ([0..2] für 8-, 10- oder 12-Bit)

- Rechnet Signal für Visualisierung um
- Stellt Einstellungs-Menüs für Visualisierung bereit

## Optimierungen

- Einen Weg finden, um [OrbitControls](#) zu integrieren (bisher Problem mit Paketversion von [drei](#))
- Farbmanagement des Canvas für korrekte Farbwiedergabe anpassen

Verwendete GeneralComponents:

[ScopesCamera](#),  
[VideoStandardAlertView](#),  
[SettingsPopoverContainer](#)

---

## Subkomponenten

---

## CiePlot

```
const CiePlot = (  
  {  
    signalxyY,  
    signalRGB,  
    dotSize = 0.01  
  }) => {...}
```

**signalxyY, signalRGB:** siehe [Signal Arrays](#)

**dotSize:** Radius der Punkte im Plot

- Bringt Bildpunkte des signalxyY in eine CIE-Normfarbtafel mit den entsprechenden Farben aus dem signalRGB

### Hinweise

- Kann nur innerhalb eines React-Three-Fiber Canvas verwendet werden

### Optimierungen

- Alternativ zu der aktuell relativen Darstellung der Farben: Modus um "echte Farben" darzustellen, sodass (durch sRGB-Farbraum-Begrenzung aus React Native) alle Farben außerhalb sRGB immer voll gesättigt dargestellt werden.
- Hier lässt sich die Performance noch deutlich optimieren, indem systematischer mit den Hooks "useMemo()", "useRef()",... gearbeitet wird und unter Verwendung eines [InstancedMesh](#). Hier muss aber auf die Paketversion von Three geachtet werden, da neuere Versionen mit anderen Bibliotheken (wie react-three-fiber) Probleme bereiten kann.

---

## COS

```
export const COS = (props) => {...}
```

**props:** --

- Koordinatensystem

## Hinweise

- Kann nur innerhalb eines React-Three-Fiber Canvas verwendet werden

## Optimierungen

- Achsenbeschriftungen fehlen
  - Achsen dürfen nicht bunt sein
- 

## CieBounds

```
export const CieBounds = () => {
```

### props: --

- Reine Spektrallinie des CIE inklusive Purpur-Linie

## Hinweise

- Kann nur innerhalb eines React-Three-Fiber Canvas verwendet werden

## Optimierungen

–

---

## GamutBounds

```
export const GamutBounds = (  
  {  
    showRec601,  
    showRec709,  
    showRec2020  
  }) => {...}
```

**showRec601:** Farbraum-Grenzen für Rec.601 einblenden (true/false)

**showRec709:** Farbraum-Grenzen für Rec.709 einblenden (true/false)

**showRec2020:** Farbraum-Grenzen für Rec.2020 einblenden (true/false)

- Zeichnet Farbraum-Grenzen in die CIE-Normfarbtafel

Hinweise

- Kann nur innerhalb eines React-Three-Fiber Canvas verwendet werden

Optimierungen

--

---

## GamutLabels

```
export const GamutLabels = (  
  {  
    showRec601,  
    showRec709,  
    showRec2020  
  }) => {...}
```

**showRec601:** Farbraum-Beschriftung für Rec.601 einblenden (true/false)

**showRec709:** Farbraum-Beschriftung für Rec.709 einblenden (true/false)

**showRec2020:** Farbraum-Beschriftung für Rec.2020 einblenden (true/false)

- Zeigt eine Legende mit den eingeblendeten Farbraum-Grenzen am Bildschirmrand an

Hinweise

--

Optimierungen

- könnte etwas schöner aussehen, ggf. auch direkt an die Gamut-Grenzen schreiben
-