

Vektorskop

VectorscopeView

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

signalYCBCR: siehe [Signal Arrays](#)

withOverlays: Buttons und Labels anzeigen (true/false)

encodedVideoStandard: Videostandard, 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

Verwendete GeneralComponents:

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

Subkomponenten

VectorscopePlot

```
const VectorscopePlot = (  
  {  
    signalSmallyCBCR,  
    signalRGB,  
    useDiscreteSignalRepresentation  
  }) => {...}
```

signalSmallyCBCR, signalRGB: siehe [Signal Arrays](#)

useDiscreteSignalRepresentation: Signal als diskrete Punkte oder als Linienzug darstellen (true/false)

- Bildet Bildpunkte des signalSmallyCBCR mit den entsprechenden Farben aus dem signalRGB im Vektorskop ab.

Hinweise

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

Optimierungen

- 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.

LinePlot

```
export const LinePlot = ({ signalSmallyCBCR }) => {...}
```

signalSmallyCBCR: siehe [Signal Arrays](#)

- Zeichnet die Bildpunkte wie ein analoges Vektorskop als Linienzug über die Bildzeilen ein.

Hinweise

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

Optimierungen

- Achsenbeschriftungen fehlen
-

VectorscopeBounds

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

props: --

- Zeichnet Achsen und Kreis mit Radius 1.

Hinweise

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

Optimierungen

– Lässt sich mit den Anmerkungen zu [VectorscopePlot](#) ggf. noch optimieren.

PeakSignalHexagon

```
export const PeakSignalHexagon = ({ videoStandard }) => {...}
```

videoStandard: Videostandard abgekürzt als String ("601", "709", "2020")

- Zeichnet abhängig vom Videostandard die Chroma-Signal-Grenzen als Hexagon ein.

Hinweise

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

Optimierungen

– Lässt sich mit den Anmerkungen zu [VectorscopePlot](#) ggf. noch optimieren.
