

Signal-Vorschau

SignalPreviewView

```
export const SignalPreviewView = (  
  {  
    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

Optimierungen

- Settings und VideoStandardAlertView einbinden
- Hinzufügen von Videostandard-Wechsel erfordert, dass für die Vorschau ein Wechsel zwischen relativer Farb-Darstellung und absoluter Farb-Darstellung besteht (alles über sRGB hinaus klippt)

Verwendete GeneralComponents:

--

Subkomponenten

SignalPreviewPlot

```
const SignalPreviewPlot = (  
  {  
    signalSmallRGB,  
    signalRGB = undefined,  
    signalYCBCR = undefined  
    labelIdx = 0  
  }) => {...}
```

signalSmallRGB, signalRGB, signalYCBCR: siehe [Signal Arrays](#)

labelIdx: Auswahl welche Beschriftung auf Bildpunkt-Gruppe eingeblendet werden soll:
keins, signalRGB oder signalYCBCR (0, 1, 2)

- Erzeugt eine Vorschau des signalSmallRGB, indem die Bildpunkt-Werte auf ein Vollbild gestreckt werden.
- Blendet optional die Signalwerte von signalRGB oder signalYCBCR ein, welche sich an den Videostandards der Rec.601, 709 und 2020 orientieren.

Hinweise

- Kann außerhalb von React-Three-Fiber Canvas verwendet werden, weil nnur React Native Views verwendet werden.

Optimierungen

- Hier lässt sich die Performance wahrscheinlich noch optimieren, indem systematischer mit den Hooks "useMemo()", "useRef()",... gearbeitet wird.
 - Manchmal erscheinen weiße Linien zwischen den Bildpunkt-Gruppen.
-