

# Backscrub

## 1) Ersetzung des Hintergrundes

- Wenn an einer Webkonferenz teilgenommen wird, besteht der Wunsch ein eigenen Hintergrund zu verwenden
- Jitsi-Meet bietet diese Funktionalität
- Andere Webkonferenzsysteme erlauben es nicht ohne weiteres.

## 2) Was ist backscrub

- Backscrub liest die Daten der verwendete Kamera
- Generiert per KI (TFLite) eine Maske.
- Ersetzt das Hintergrund.
- Fügt die maskierte Person auf das Hintergrund.
- Sendet da Ergebnis an ein V4L2 Loopback Gerät.
- Der Browser erhält die Daten des Virtuellen Gerätes.

## 3) Vorteil von backscrub gegenüber Browser basierte Lösungen

- Schneller.
- Größere Flexibilität.

## 4) backscrub Quellen

- [auf Github](#)
- [überarbeitete Version \(Branch Testing\)](#)

## 5) Warum eine veränderte Version

- Backscrub wurde auf Ubuntu abgestimmt.
- Dadurch kam es zu Ungereimtheiten auf beispielsweise Fedora.
- Manches schien mir nicht gerade sehr gut durchdacht zu sein.
- Es kam immer wieder zu Fehler.
- Die Größe der Eingabe und Ausgabegeräte können unterschiedlich gewählt werden. dies kann zur Verzerrung des Bildes führen (Pixel Aspect Ratio).

## 6) Arbeiten

- Als erste Anpassung des Codes, damit es nicht zu diese Verzerrung kommt.
- Überarbeitung des Codes, damit es nicht zu Fehler kommt.
- Anpassungen damit, unabhängig vom verwendeten OpenCV Backend (ffmpeg, gstreamer) das System die gewünschte Ergebnissen liefert.
- Unterstützung von HDMI-Grabber als Camera Eingangsquelle:
  - Sehr große Format (1920x1080)

- Zu hohe Bildwiederholung Frequenz
- Maskierung suboptimal (Maske des vorhergehendes Bildes auf das aktuelles Bild).
- Überarbeitung der Erfassung der übergebenen Argumente und freundlichere Ausgabe.
- Diverse kleine Optimierungen.

## 7) Erweiterungen im Detail

Nachstehende Optionen wurden hinzugefügt:

### 7.1) Begrenzung der Wiederholungsrate

- -mf. -max-fps <Rate>

Eine über HDMI-Grabber angeschlossene Kamera kann einer zu hohe Bildwiederholungsrate bedingen. Mit der Option wird sichergestellt, dass jeden Nte Bild, zwar gelesen aber nicht weiter verarbeitet wird. Dies erniedrigt die CPU-Belastung.

Die Bildwiederholungsfrequenz der Kamera wird nach der Initialisierung ermittelt. Danach wird ein Divisor berechnet. Mit einer maximalen <Rate> von beispielsweise 30 FPS wird beispielsweise eine Wiederholungsfrequenz von 25 bzw. 30 Hz, je nach Kamera/Grabber Fähigkeit (50 oder 60 FPS) erreicht. Für WebRTC sind die Werte in Ordnung, normalerweise wird maximal 30 FPS angeboten.

### 7.2) Maskierung der Person

- -vd, -video-delayed
  - Eine reine sequenzielle Bearbeitung würde schnell zur Verringerung der Bildwiederholungsfrequenz führen.
  - Multithreading erlaubt es Berechnungen zu parallelisieren.
- Bakscrub nutzt dementsprechend Multithreading:
  - Folge ist, dass die generierte Maske nicht für das aktuelle Bild berechnet wurde:
    - Bei Bewegungen erscheint das realer Hintergrund an den Rändern!
- Zwei Bilder zu speichern, mit dem neuesten die Maskenberechnung anzustoßen und die vorhergehenden Maske und Bild danach zu verarbeiten löst einiges.

#### 7.2.1) Nachteile des verzögerten senden des Bildes

- Die Synchronität des Tones und des Bildes sind schlechter.
- Ein voreilender Ton wird, laut EBU Recommendation R37, nicht wahrgenommen werden, bei 80 ms wird es von 1/3 der Teilnehmern registriert.
- Die Verzögerung darf, je nach Hardware, 35 bis 55 ms betragen.

#### 7.2.2) Vorteile

- Störenden Artefakten im Video treten nicht auf.
- In den meisten Fällen wird die Verzögerung nicht als störend empfunden.