

# Dokumentation

## Projekt Shot-Detection

In diesem Projekt geht es darum, ein Video (Sintel Trailer 720p) zu analysieren und mittels HTML5, CSS3 und JavaScript die Szenenwechsel zu erkennen. Dabei werden zusätzlich in Echtzeit Thumbnails der Shots generiert und neben des allgemeinen Recalls (Gesamtanzahl der erkannten Shots) deren Präzision (Recall minus „false positives“) ermittelt. Des Weiteren wurde eine Funktion implementiert, die es erlaubt mit Klick auf die Thumbnails zum jeweiligen Zeitpunkt im Video zu springen.

### Aufbau / Struktur der Implementierung

Das System besteht aus folgenden Files:

index.html	HTML5-Seite, welche mit dem video-Tag das Video anzeigt. Das canvas-Tag dient zur Darstellung des Histogramms und das div mit der id „sceneContainer“ zur Anzeige der erfassten Shots in Form von Thumbnails und der Videozeit.
app.css	Formatierung der index.html
sintel_trailer-720p.mp4	Zu analysierendes Video
videoprocessor.js	Dieser JS-Code dient zum Laden des Videos (doLoad). Darüber hinausgehend wird die Verarbeitung von Frames darin ausgelöst (für das Histogramm) , welche dann von den beiden nachfolgenden Web Workern asynchron ausgeführt wird. Außerdem wird hier jedes Frame mit seiner Zeit zwischengespeichert, sollte es später als Schnitt erkannt und zur Anzeige benötigt werden
frame-selection.js	Hier verbirgt sich die Logik um redundante Frames auszusortieren. Dies ist notwendig, da in videoprocessor.js mit einem interval von 0 ms gearbeitet wird um die höchste Framerate unterstützen zu können.
frame-selection.worker.js	Ein Web Worker um die frame-selectionmultithreaded zu verarbeiten.
compute-frame.worker.js	Dieser Web Worker übernimmt die Aufgabe des Errechnens des Histogrammes aus dem Frame.

histogram.js	Die histogram.js dient zur Erfassung der RGB-Werte jedes erfassten Frames in Echtzeit und berechnet den Y-Wert (Helligkeit des Frames).
histogram-drawer.js	Hier werden die Daten eines Histogramm-Objektes (histogram.js) auf dem Dokument visualisiert.
frame.js	Die frame.js generiert die Thumbnails und deren onClick-Funktion, um zu einem gewissen Zeitpunkt im Video zu springen. Es wird auch die dadurch erfasste Videozeit formatiert und unter dem Thumbnail mitausgegeben.
shotdetection.js	In der shotdetection.js werden die Einstellungen für die Szenenwechserkennung gesetzt, das heißt es werden die Histogramme gewählt, die zur Berechnung des Szenenwechsel verwendet werden (je nach Einstellung können Histogramme/Frames übersprungen werden). Die Berechnungen werden auf extra Threads durch den scene-change(-multiple).worker.js durchgeführt.
scene-change detector.js	Der sceneChangeDetector eruiert das Auftreten eines Szenenwechsel anhand der Daten (Histogramme), die er zugeführt bekommt. Dies geschieht via der ermittelten Unterschiede im Histogramm beziehend auf die Helligkeit (histogramdifferences).
scene-change.worker.js	Dieser Web Worker verwendet den scene-change-detector.js um einen Szenenwechsel von 2 Histogrammen asynchron festzustellen und zur shotdetection.js zurückzumelden.
scene-change-multiple.worker.js	Ähneln dem vorhergehenden Web Worker, er unterscheidet sich aber darin, dass er beliebig viele Histogramme vergleicht und daraus maximal einen Szenenwechsel zurückgibt.
statistics.js	Die statistics.js analysiert die erfassten Shots und gibt Recall und Precision aus. Dafür wurden im Vorfeld manuell die Sekunden, in denen ein Szenenwechsel vorkommt, definiert (sintelTrailerCutsAtSecond). Weiters wurde die Anzahl der korrekten (C), fehlenden (M) und falschen (F) Treffer jeweils ermittelt um im späteren Verlauf daraus den Recall und die Präzision zu berechnen. Recall wird mit $C / (C+M)$ berechnet und die Präzision mit $C / (C+F)$ .

## User Manual

Das Projekt ist ganz simpel in einen lokalen Server einzubinden und via localhost im Browser aufzurufen um Browserbeschränkungen zu vermeiden.

Wenn das Projekt im Browser geöffnet ist, braucht man nur das Video starten.

Man ist nun in der Lage das Histogramm, den Recall und die Precision rechts vom Video-Player in Echtzeit beobachten zu können. Währenddessen werden die erkannten Shots unter dem Player mit einem Thumbnail und der Zeit aus dem Video für jeden Szenenwechsel erstellt.

Mit Klick auf eines der Thumbnails springt das Video zur erfassten Zeit und pausiert es. **ACHTUNG:** Wenn dieser Schritt durchgeführt wird, stimmen die statistischen Werte von Recall und Precision nicht mehr. Es wird daher empfohlen die Anwendung im Browser zu aktualisieren um keine verfälschten Ergebnisse zu erhalten!

Da die Werte der „echten“ Shots in `sintelTrailerCutsAtSecond` fix drin stehen (also von Menschenhand definiert wurden), kann man leider kein anderes Video verwenden als das dafür vorgesehene `sintel_trailer-720p.mp4` file. Dies gilt jedoch nur, wenn Wert auf Recall und Precision gelegt wird. Die ShotDetection soll natürlich auch mit anderen Videos funktionieren (dazu muss man nur im `source`-Attribut des `video`-Tags im `index.html` den Dateinamen ändern und dementsprechend in derselben Ordner Ebene speichern).

Es wurde zusätzlich auch an einem Projekt mit Szenenerkennung durch Edge Detection gearbeitet, allerdings fehlten die Ressourcen um dieses auf eine stabile Version zu bringen bzw. es mit der Histogramm Szenenerkennung zusammenzuführen. Eine kombinierte Szenenerkennung hat mit Sicherheit mehr Potential richtige Entscheidungen zu treffen.