

Key-Frame-based Video Browser with HTML5

Mario Graf, Verena Dittmer, Jameson Steiner

10. Februar 2016

Implementierung

Wir haben uns für die Umsetzung dazu entschieden, die Applikation rein unter Verwendung des Canvas-Elements zu implementieren. Dadurch wirkt der Quellcode der index.html Datei sehr übersichtlich und aufgeräumt:

```
<body>
<div id="container">
  <video width="854" height="480" id="videoElement">
    <source src="media/bunny.mp4" type="video/mp4">
    Your browser does not support HTML5 video.
  </video>
  <canvas id="keyframeBrowser" width="854" height="100"></canvas>
  <canvas id="timeline"></canvas>
  <div id="controls">
    <button type="button" id="play-pause" class="play">Play/Pause</button>
    <input type="range" id="volume-bar" min="0" max="1" step="0.1" value="1">
  </div>
</div>
</body>
```

Es werden zwei Canvas Elemente eingesetzt:

- **keyFrameBrowser:** Wird verwendet um die Thumbnails und deren Animationen zu rendern. Die Implementierung dazu befindet sich in **KeyframeBrowser.js**.
- **timeline:** Wird verwendet um die dazugehörige Timeline zu rendern. Die Implementierung dazu befindet sich in **timeline/Timeline.js**.

Eine lauffähige Version dieses Projektes kann unter www.jmsn.at/videoBrowser ausprobiert werden. Wir haben unsere Implementierung unter Chrome und Safari getestet.

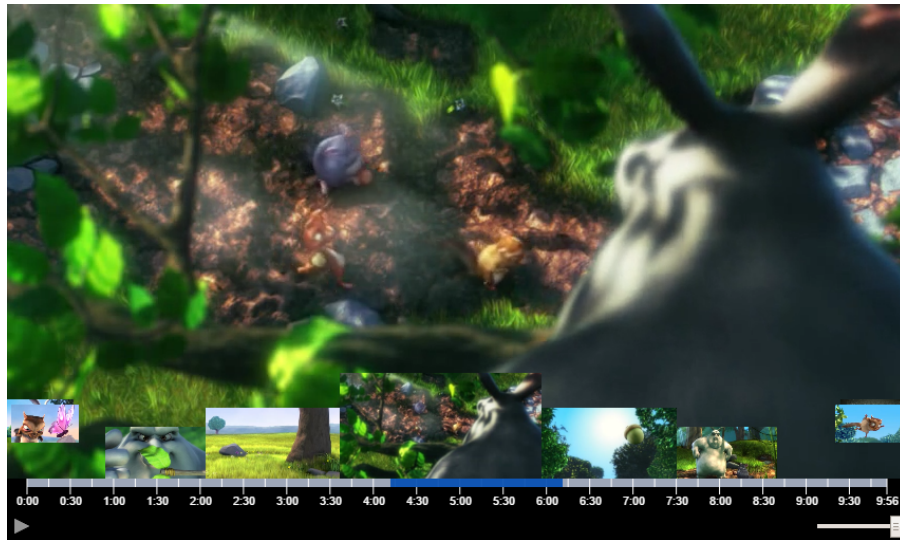


Abbildung 1: Die 5 Keyframes und die Stapel der Nachbarclusters des zweiten Levels

Keyframes- und Segment-Generierung

Als Testvideo für unseren Player haben wir Big Buck Bunny verwendet. Hierfür wurden die Thumbnails mittels ffmpeg extrahiert. Alle 5 Sekunden wird dabei ein Keyframe mit reduzierter Auflösung erstellt.

Aus den Thumbnails wird mittels dem **Clustering** Java-Programm (zu finden im Ordner clustering) ein Json-File erstellt, welches die unterschiedlichen Level der Segmente anzeigt. In dem Json-File steht für jeden Keyframe der Dateiname, die **FromTime** und **ToTime** (d.h die früheste und späteste Zeit von allen Frames der Level drunter), die Abspielzeit in Sekunden im Video, und die jeweiligen Keyframes von einem Level drunter. Das mittlere Keyframe ist dabei der Repräsentant für ein Level darüber.

Darstellung der Thumbnails

Um die zahlreichen Animationen der darzustellenden Thumbnails bewerkstelligen zu können, haben wir eine eigene Render- und Animationsengine implementiert (**RenderingEngine.js**). Diese Engine hält eine Referenz zu allen dargestellten Thumbnails. Jedes Thumbnail ist eine Instanz von **Renderable** (**Renderable.js**) und liefert die nötigen Informationen für die Berechnung der Animation.

Die Engine ist in der Lage Position, Größe (Skalierungsfaktor) und Transparenz der einzelnen **Renderables** zu animieren.

Besonders herausfordernd war dabei, die Animationen und Übergänge zwischen den Hierarchien derart zu gestalten, dass der Benutzer auch versteht bzw. einen Eindruck davon bekommt was gerade durch seine Mausinteraktion passiert (zoom-in Effekt, side-scroll Effekt).

Timeline

Die Timeline kann parametrisiert über die Videolänge und die gewünschte Breite der Timeline erstellt werden. Hierfür werden innerhalb eines Canvas der blaue Hintergrundstreifen, die Timestamps und die zugehörigen Striche in entsprechenden Abständen generiert. Eine weitere Funktion bietet die Möglichkeit, einen bestimmten Bereich des Videos einzufärben. Dies wird für das Anzeigen der Positionen der Keyframes innerhalb des Videos genutzt. Hierfür wird die **FromTime** des linken und die **ToTime** des rechten Keyframes der Timeline zum Einfärben übergeben.

Abspielen des Videos

Durch Klicken auf einen Keyframe wird das Video ab dieser Stelle abgespielt. Hierfür wird die Abspielzeit, die im Json-File zu dem zugehörigen Keyframe gespeichert war, dem Video zum Abspielen übergeben.