



**Kostenloses eBook**

# LERNEN

---

## video

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#video**

# Inhaltsverzeichnis

Über.....	1
<b>Kapitel 1: Erste Schritte mit Video.....</b>	<b>2</b>
Bemerkungen.....	2
Examples.....	2
Grundlegendes zu eigenständigen Mediendateien.....	2
<b>Allgemeines.....</b>	<b>3</b>
<b>Video.....</b>	<b>3</b>
<b>Audio.....</b>	<b>5</b>
<b>Zusammenfassung der Analyse.....</b>	<b>5</b>
<b>Mehr.....</b>	<b>6</b>
<b>Kapitel 2: Medienpräsentation verstehen.....</b>	<b>7</b>
Bemerkungen.....	7
Examples.....	7
Grundlegendes zu eigenständigen Mediendateien.....	7
<b>Allgemeines.....</b>	<b>8</b>
<b>Video.....</b>	<b>8</b>
<b>Audio.....</b>	<b>10</b>
<b>Zusammenfassung der Analyse.....</b>	<b>10</b>
<b>Mehr.....</b>	<b>11</b>
Komponenten von Videospuren verstehen.....	11
Grundlegendes zu adaptiven DASH-Streaming-Präsentationen.....	14
<b>Kapitel 3: Video-Seitenverhältnisse.....</b>	<b>19</b>
Bemerkungen.....	19
Examples.....	19
Seitenverhältnis anzeigen (DAR).....	19
Bildseitenverhältnis (PAR).....	20
Beispielseitenverhältnis (SAR).....	20
Pixel-Seitenverhältnis.....	21
<b>Credits.....</b>	<b>22</b>



You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [video](#)

It is an unofficial and free video ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official video.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Kapitel 1: Erste Schritte mit Video

## Bemerkungen

Die Videowiedergabe steht bei einer Vielzahl moderner Lösungen im Vordergrund und die Software und Standards entwickeln sich schnell. Um dieses Feld zu verstehen, müssen Sie zunächst die verschiedenen Aspekte der Arbeit mit Video verstehen:

- Rohfarbeninformationen, die von der physischen Welt erfasst werden, werden im Allgemeinen mit einem Codec *codiert* - einem Algorithmus, dessen Zweck es ist, diese Daten in komprimierter Form darzustellen, wobei häufig einige visuelle Details zugunsten einer stärkeren Komprimierung geopfert werden.
- Für die Wiedergabe wird der inverse Algorithmus ausgeführt - die Daten werden *decodiert*, um wieder eine rohe Farbinformation zu werden, die einem Ausgabegerät (z. B. einem Monitor) zugeführt werden kann.
- Zwischen Codieren und Decodieren wird die komprimierten Daten für die Lagerung *verpackt*, die Spuren von verschiedenen Typen in einer einzigen Datei oder die Segmentierung der Inhalt in eine große Anzahl von kleinen Segmenten beinhalten kann kombiniert.
- Das Video wird auf den Endbenutzers Gerät *liefert* eine Lieferung Technologie, die über HTTP so einfach wie eine Datei heruntergeladen werden kann oder wesentlich komplexe, Live - Feedback von der Netzwerkinfrastruktur und die automatische Anpassung der Qualitätsstufen beteiligt ist .
- Premium-Inhalte werden normalerweise vor dem Verpacken *verschlüsselt* und können nur in einem Player abgespielt werden, der mit einer DRM-Technologie ausgestattet ist, die die Sicherheit des Entschlüsselungsschlüssels während der Verwendung gewährleistet und aktiv vor der Erfassung der Ausgabe schützt.

Während der visuelle Teil offensichtlich in der Sichtbarkeit dominiert, spielen Audio und Text auch eine wichtige Rolle in Medienpräsentationen und bieten mehrsprachige Funktionen, die Inhalte für ein breites Publikum zugänglich machen. In den meisten Workflows werden Audio- und Textspuren auf ähnliche Weise wie Videospuren gehandhabt, codiert, decodiert, verpackt und entlang derselben Linien geliefert.

All diese Aspekte - und mehr - müssen in einer modernen Lösung gepflegt werden, um ein angenehmes Erlebnis für die Endbenutzer zu gewährleisten.

## Examples

### Grundlegendes zu eigenständigen Mediendateien

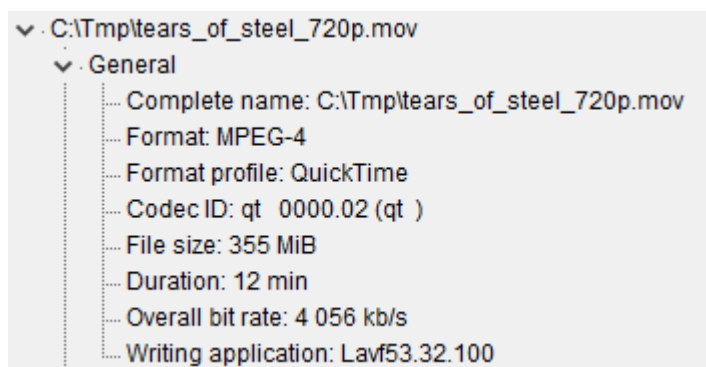
Der hier verwendete Beispieldinhalt ist Tears of Steel von Blender Foundation. Insbesondere verwenden wir den Download mit dem Titel "HD 720p (~ 365MB, mov, 2.0)". Dies ist eine einzelne Datei, die mit der Erweiterung "mov" endet und in nahezu jedem modernen Mediaplayer abgespielt werden kann.

Beachten Sie, dass die Downloadseite Untertitel als separate SRT-Dateidownloads enthält. In diesem Beispielinhalt werden keine Untertitel in derselben Datei bereitgestellt. Wir lassen die Untertitelanalyse daher nicht in diesem Beispiel.

Eine einfache Möglichkeit, verschiedene Mediendateien zu analysieren, ist das Tool / die Bibliothek [MediaInfo](#) . Während die hier gezeigte Analysefunktionalität der Einfachheit halber die GUI verwendet, sind alle Funktionen auch über die MediaInfo-API verfügbar.

Wenn Sie diese Datei in der MediaInfo-GUI öffnen und zur Strukturansicht wechseln, werden drei Bereiche angezeigt: Allgemein, Video und Audio. Die erste enthält grundlegende Informationen zur Datei, während die beiden anderen jeweils einen in dieser Datei gefundenen Medientrack beschreiben. Untersuchen wir die relevantesten Informationen in jedem Abschnitt der Ausgabe.

## Allgemeines



Die ersten Parameter von Interesse sind *Format* und *Formatprofil* . Der erste besagt, dass das **Verpackungsformat** aus der MPEG-4-Standardsuite stammt. MPEG-4 definiert das *ISO Base Media File-Format* und das *MP4-Paketformat* . Darüber hinaus hat Apple eine eigene Spezifikation erstellt, die von diesen abgeleitet wird und in MediaInfo als "QuickTime" -Profil bezeichnet wird.

Hinweis: Achten Sie darauf, dass Sie MP4 und MPEG-4 nicht verwechseln. Ersteres bezieht sich auf ein bestimmtes Verpackungsformat in der MPEG-4-Suite internationaler Standards, das auch Video- und Audio-Codecs enthält. Dies kann zu Verwirrung führen. Vermeiden Sie daher die Verwendung des Begriffs MPEG-4, wenn Sie sich auf etwas anderes als die vollständigen Normen beziehen.

Alle auf dem ISO-Basismediendateiformat basierenden Verpackungsformate, die in der MPEG-4-Standardfamilie definiert sind, sind sehr ähnlich und können häufig mit denselben Tools verarbeitet werden, wobei es sich bei ihren Unterschieden im Wesentlichen um kundenspezifische Erweiterungen handelt, die oft sicher sind ignoriert Daher können wir erwarten, dass das Beispielvideo hier mit allen modernen Video-Playern kompatibel ist.

## Video

```
Video
  ID: 1
  Format: AVC
  Format/Info: Advanced Video Codec
  Format profile: Main@L3.1
  Format settings, CABAC: No
  Format settings, ReFrames: 2 frames
  Format settings, GOP: M=4, N=18
  Codec ID: avc1
  Codec ID/Info: Advanced Video Coding
  Duration: 12 min
  Bit rate: 4 000 kb/s
  Width: 1 280 pixels
  Height: 534 pixels
  Display aspect ratio: 2.40:1
  Frame rate mode: Constant
  Frame rate: 24.000 FPS
  Color space: YUV
  Chroma subsampling: 4:2:0
  Bit depth: 8 bits
  Scan type: Progressive
  Bits/(Pixel*Frame): 0.244
  Stream size: 338 MiB (95%)
  Writing library: x264 core 122
  Encoding settings: cabac=0 / ref=2 / deblock=1:0:0 / ar
  Language: English
```

Das wichtigste Detail der Videospur ist der Codec, mit dem Rohfarbdaten in eine komprimierte Form umgewandelt werden. Der Name des Codecs wird vom Parameter *Format* angegeben.

AVC ist auch als H.264 bekannt. Es ist der heute am weitesten verbreitete Videocodec, der auf praktisch allen modernen Geräten und Softwareplattformen unterstützt wird. Eine mit AVC codierte Videospur kann mit fast jedem Player abgespielt werden.

Codecs verfügen häufig über mehrere *Profile*, die eine Unterteilung der Codec-Funktionalität in Stufen ermöglichen und eine kontrollierte Weiterentwicklung der Technologie ermöglichen. Der Parameter *Profilparameter* gibt an, dass dieses Video das Hauptprofil verwendet. Dieses Profil ist relativ ungewöhnlich, da fast alle modernen Geräte das High-Profil unterstützen, das eine höhere Komprimierungseffizienz bietet.

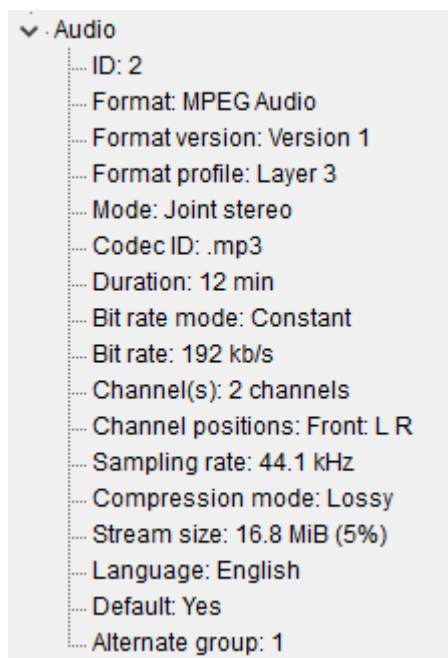
Die Qualität der Videospur ist oft von größter Bedeutung. Hier sehen wir die kritischen Faktoren, die durch die Parameter *Bitrate*, *Breite* und *Höhe* ausgedrückt werden. Die letzten beiden weisen darauf hin, dass dies eine 720p-Videospur sein soll, was als HD-Qualität der unteren Preisklasse angesehen wird. Das Bild ist vertikal kürzer als das Standard-720p-Bild von 1280 x 720 Pixeln.

Die Bitrate misst die Datenmenge, die die komprimierte Form des Videostroms durchschnittlich für jede Sekunde der Wiedergabe einnimmt. Dies ist ein entscheidender Parameter für die Optimierung, da die Menge der gelieferten Daten bei großen Videolösungen eine große Kostenquelle darstellt.

Die oben genannten Datenpunkte zur Videoqualität sind einfache Fakten, die wir aus der Analyse erhalten. Jede Beurteilung der Angemessenheit dieser Parameter ist ein Thema, das weitaus

mehr Analyse erfordert und von separaten Themen in dieser Dokumentationskategorie behandelt wird, wie auch vielen anderen Feinheiten von Arbeit mit Videospuren.

## Audio



Auch hier ist es von entscheidender Bedeutung, den für die Codierung der Audiodaten verwendeten Codec zu kennen. Dies wird durch die *Profilparameter Format* und *Format* ausgedrückt. "MPEG Audio Layer 3" wird häufiger als MP3 bezeichnet und ist ein universell unterstütztes Audioformat, von dem erwartet wird, dass es überall abgespielt wird.

Wie bei Video sind Audioqualitätsparameter die zweitwichtigsten Datenpunkte, die hauptsächlich durch den Parameter *Bitrate* ausgedrückt werden.

## Zusammenfassung der Analyse

Der Inhalt wird in einem sehr beliebten Paketformat verpackt, das auf den Standards für MPEG-4-Standards basiert. Es wird mit universell verwendeten Video- und Audio-Codecs codiert. Daraus ergibt sich, dass das Video für jeden Betrachter leicht zugänglich sein soll - Kompatibilität und Verfügbarkeit waren für die Autoren von entscheidender Bedeutung.

Die Verwendung von MP3 zeigt das Alter der Beispieldateien, da es nicht mehr den heutigen Konkurrenten gewachsen ist - stattdessen ist AAC (Advanced Audio Coding) der Verursacher von Audio-Codecs.

Gleiches gilt für die Verwendung des H.264-Hauptprofils. Es ist sehr selten, dass ein H.264-Profil außer High verwendet wird, da fast alle Decoder dies unterstützen, sodass jeder die verbesserte Effizienz nutzen kann, die durch die High-Profile-Funktionen ermöglicht wird.

Die verwendeten Bitraten sind etwas höher als in der heutigen Umgebung erwartet. Dies kann

durch den Wunsch der Autoren nach hoher Qualität oder einfach durch die Einschränkungen der Encoder erklärt werden, die bei der Erstellung des Inhalts verfügbar waren.

---

## Mehr

Weitere nützliche Tools für die Mediendatei-Analyse sind [FFprobe](#), das Teil des FFmpeg-Softwarepakets ist, und die [Bento4-Tools](#) zum Arbeiten mit MP4-Dateien. Beide sind auch in Bibliotheksform verfügbar. Sie sind in der Lage, eine tiefere Analyse durchzuführen als MediaInfo, wenn Sie einzelne Elemente untersuchen müssen, aus denen Mediendateien bestehen.

**Erste Schritte mit Video online lesen:** <https://riptutorial.com/de/video/topic/5690/erste-schritte-mit-video>



---

# Kapitel 2: Medienpräsentation verstehen

## Bemerkungen

Die Videowiedergabe steht bei einer Vielzahl moderner Lösungen im Vordergrund und die Software und Standards entwickeln sich schnell. Um zu verstehen, was eine Medienpräsentation ausmacht, müssen Sie zunächst die verschiedenen Aspekte der Arbeit mit Video verstehen:

- Rohfarbeninformationen, die von der physischen Welt erfasst werden, werden im Allgemeinen mit einem Codec *codiert* - einem Algorithmus, dessen Zweck es ist, diese Daten in komprimierter Form darzustellen, wobei häufig einige visuelle Details zugunsten einer stärkeren Komprimierung geopfert werden.
- Für die Wiedergabe wird der inverse Algorithmus ausgeführt - die Daten werden *decodiert*, um wieder eine rohe Farbinformation zu werden, die einem Ausgabegerät (z. B. einem Monitor) zugeführt werden kann.
- Zwischen Codieren und Decodieren wird die komprimierten Daten für die Lagerung *verpackt*, die Spuren von verschiedenen Typen in einer einzigen Datei oder die Segmentierung der Inhalt in eine große Anzahl von kleinen Segmenten beinhalten kann kombiniert.
- Das Video wird auf den Endbenutzers Gerät *liefert* eine Lieferung Technologie, die über HTTP so einfach wie eine Datei heruntergeladen werden kann oder wesentlich komplexe, Live - Feedback von der Netzwerkinfrastruktur und die automatische Anpassung der Qualitätsstufen beteiligt ist .
- Premium-Inhalte werden normalerweise vor dem Verpacken *verschlüsselt* und können nur in einem Player abgespielt werden, der mit einer DRM-Technologie ausgestattet ist, die die Sicherheit des Entschlüsselungsschlüssels während der Verwendung gewährleistet und aktiv vor der Erfassung der Ausgabe schützt.

Während der visuelle Teil offensichtlich in der Sichtbarkeit dominiert, spielen Audio und Text auch eine wichtige Rolle in Medienpräsentationen und bieten mehrsprachige Funktionen, die Inhalte für ein breites Publikum zugänglich machen. In den meisten Workflows werden Audio- und Textspuren auf ähnliche Weise wie Videospuren gehandhabt, codiert, decodiert, verpackt und entlang derselben Linien geliefert.

Alle diese Aspekte - und mehr - wirken sich auf die Komposition, Formatierung und Verwendung einer Medienpräsentation aus. Sie müssen als Ganzes verstanden werden, um Medientechnologien effektiv zu nutzen.

## Examples

### Grundlegendes zu eigenständigen Mediendateien

Der hier verwendete Beispieldinhalt ist [Tears of Steel](#) von Blender Foundation. Insbesondere verwenden wir den Download mit dem Titel "HD 720p (~ 365MB, mov, 2.0)". Dies ist eine einzelne Datei, die mit der Erweiterung "mov" endet und in nahezu jedem modernen Mediaplayer abgespielt werden kann.

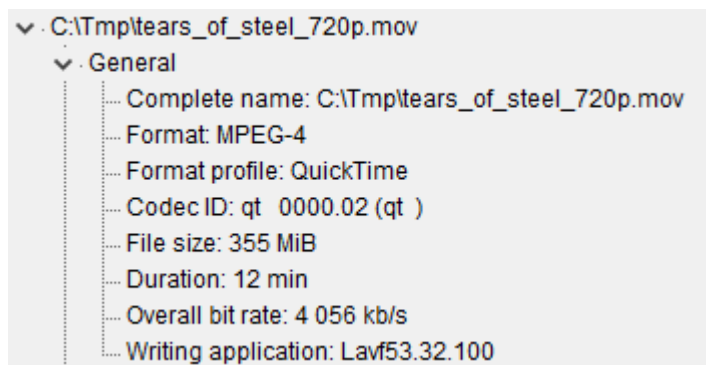
Beachten Sie, dass die Downloadseite Untertitel als separate SRT-Dateidownloads enthält. In diesem Beispielinhalt werden keine Untertitel in derselben Datei bereitgestellt. Wir lassen die Untertitelanalyse daher nicht in diesem Beispiel.

Eine einfache Möglichkeit, verschiedene Mediendateien zu analysieren, ist das Tool / die Bibliothek [MediaInfo](#) . Während die hier gezeigte Analysefunktionalität der Einfachheit halber die GUI verwendet, sind alle Funktionen auch über die MediaInfo-API verfügbar.

Wenn Sie diese Datei in der MediaInfo-GUI öffnen und zur Strukturansicht wechseln, werden drei Bereiche angezeigt: Allgemein, Video und Audio. Die erste enthält grundlegende Informationen zur Datei, während die beiden anderen jeweils einen in dieser Datei gefundenen Medientrack beschreiben. Untersuchen wir die relevantesten Informationen in jedem Abschnitt der Ausgabe.

---

## Allgemeines



Die ersten Parameter von Interesse sind *Format* und *Formatprofil* . Der erste besagt, dass das **Verpackungsformat** aus der MPEG-4-Standardsuite stammt. MPEG-4 definiert das *ISO Base Media File-Format* und das *MP4-Paketformat* . Darüber hinaus hat Apple eine eigene Spezifikation erstellt, die von diesen abgeleitet wird und in MediaInfo als "QuickTime" -Profil bezeichnet wird.

Hinweis: Achten Sie darauf, dass Sie MP4 und MPEG-4 nicht verwechseln. Ersteres bezieht sich auf ein bestimmtes Verpackungsformat in der MPEG-4-Suite internationaler Standards, das auch Video- und Audio-Codecs enthält. Dies kann zu Verwirrung führen. Vermeiden Sie daher die Verwendung des Begriffs MPEG-4, wenn Sie sich auf etwas anderes als die vollständigen Normen beziehen.

Alle auf dem ISO-Basismediendateiformat basierenden Verpackungsformate, die in der MPEG-4-Standardfamilie definiert sind, sind sehr ähnlich und können häufig mit denselben Tools verarbeitet werden, wobei es sich bei ihren Unterschieden im Wesentlichen um kundenspezifische Erweiterungen handelt, die oft sicher sind ignoriert Daher können wir erwarten, dass das Beispielvideo hier mit allen modernen Video-Playern kompatibel ist.

---

## Video

```
Video
  ID: 1
  Format: AVC
  Format/Info: Advanced Video Codec
  Format profile: Main@L3.1
  Format settings, CABAC: No
  Format settings, ReFrames: 2 frames
  Format settings, GOP: M=4, N=18
  Codec ID: avc1
  Codec ID/Info: Advanced Video Coding
  Duration: 12 min
  Bit rate: 4 000 kb/s
  Width: 1 280 pixels
  Height: 534 pixels
  Display aspect ratio: 2.40:1
  Frame rate mode: Constant
  Frame rate: 24.000 FPS
  Color space: YUV
  Chroma subsampling: 4:2:0
  Bit depth: 8 bits
  Scan type: Progressive
  Bits/(Pixel*Frame): 0.244
  Stream size: 338 MiB (95%)
  Writing library: x264 core 122
  Encoding settings: cabac=0 / ref=2 / deblock=1:0:0 / ar
  Language: English
```

Das wichtigste Detail der Videospur ist der Codec, mit dem Rohfarbdaten in eine komprimierte Form umgewandelt werden. Der Name des Codecs wird vom Parameter *Format* angegeben.

AVC ist auch als H.264 bekannt. Es ist der heute am weitesten verbreitete Videocodec, der auf praktisch allen modernen Geräten und Softwareplattformen unterstützt wird. Eine mit AVC codierte Videospur kann mit fast jedem Player abgespielt werden.

Codecs verfügen häufig über mehrere *Profile*, die eine Unterteilung der Codec-Funktionalität in Stufen ermöglichen und eine kontrollierte Weiterentwicklung der Technologie ermöglichen. Der Parameter *Profilparameter* gibt an, dass dieses Video das Hauptprofil verwendet. Dieses Profil ist relativ ungewöhnlich, da fast alle modernen Geräte das High-Profil unterstützen, das eine höhere Komprimierungseffizienz bietet.

Die Qualität der Videospur ist oft von größter Bedeutung. Hier sehen wir die kritischen Faktoren, die durch die Parameter *Bitrate*, *Breite* und *Höhe* ausgedrückt werden. Die letzten beiden weisen darauf hin, dass dies eine 720p-Videospur sein soll, was als HD-Qualität der unteren Preisklasse angesehen wird. Das Bild ist vertikal kürzer als das Standard-720p-Bild von 1280 x 720 Pixeln.

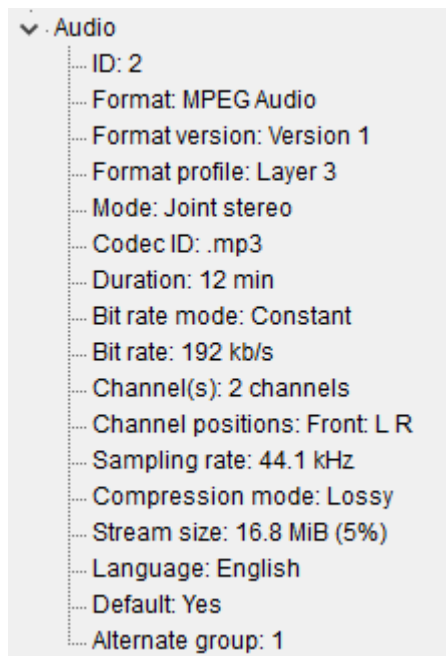
Die Bitrate misst die Datenmenge, die die komprimierte Form des Videostroms durchschnittlich für jede Sekunde der Wiedergabe einnimmt. Dies ist ein entscheidender Parameter für die Optimierung, da die Menge der gelieferten Daten bei großen Videolösungen eine große Kostenquelle darstellt.

Die oben genannten Datenpunkte zur Videoqualität sind einfache Fakten, die wir aus der Analyse erhalten. Jede Beurteilung der Angemessenheit dieser Parameter ist ein Thema, das weitaus

mehr Analyse erfordert und von separaten Themen in dieser Dokumentationskategorie behandelt wird, wie auch vielen anderen Feinheiten von Arbeit mit Videospuren.

---

## Audio



Auch hier ist es von entscheidender Bedeutung, den für die Codierung der Audiodaten verwendeten Codec zu kennen. Dies wird durch die *Profilparameter Format* und *Format* ausgedrückt. "MPEG Audio Layer 3" wird häufiger als MP3 bezeichnet und ist ein universell unterstütztes Audioformat, von dem erwartet wird, dass es überall abgespielt wird.

Wie bei Video sind Audioqualitätsparameter die zweitwichtigsten Datenpunkte, die hauptsächlich durch den Parameter *Bitrate* ausgedrückt werden.

---

## Zusammenfassung der Analyse

Der Inhalt wird in einem sehr beliebten Paketformat verpackt, das auf den Standards für MPEG-4-Standards basiert. Es wird mit universell verwendeten Video- und Audio-Codecs codiert. Daraus ergibt sich, dass das Video für jeden Betrachter leicht zugänglich sein soll - Kompatibilität und Verfügbarkeit waren für die Autoren von entscheidender Bedeutung.

Die Verwendung von MP3 zeigt das Alter der Beispieldateien, da es nicht mehr den heutigen Konkurrenten gewachsen ist - stattdessen ist AAC (Advanced Audio Coding) der Verursacher von Audio-Codecs.

Gleiches gilt für die Verwendung des H.264-Hauptprofils. Es ist sehr selten, dass ein H.264-Profil außer High verwendet wird, da fast alle Decoder dies unterstützen, sodass jeder die verbesserte Effizienz nutzen kann, die durch die High-Profile-Funktionen ermöglicht wird.

Die verwendeten Bitraten sind etwas höher als in der heutigen Umgebung erwartet. Dies kann

durch den Wunsch der Autoren nach hoher Qualität oder einfach durch die Einschränkungen der Encoder erklärt werden, die bei der Erstellung des Inhalts verfügbar waren.

---

## Mehr

Weitere nützliche Tools für die Mediendatei-Analyse sind [FFprobe](#), das Teil des FFmpeg-Softwarepakets ist, und die [Bento4-Tools](#) zum Arbeiten mit MP4-Dateien. Beide sind auch in Bibliotheksform verfügbar. Sie sind in der Lage, eine tiefere Analyse durchzuführen als MediaInfo, wenn Sie einzelne Elemente untersuchen müssen, aus denen Mediendateien bestehen.

### Komponenten von Videospuren verstehen

In diesem Beispiel wird erläutert, wie das Layout einer Videospur angezeigt wird und wie die einzelnen Bilder darin extrahiert werden.

Der hier verwendete Beispielinhalt ist [Tears of Steel](#) von Blender Foundation. Insbesondere verwenden wir den Download mit dem Titel "HD 720p (~ 365MB, mov, 2.0)". Dies ist eine einzelne Datei, die mit der Erweiterung "mov" endet und in nahezu jedem modernen Mediaplayer abgespielt werden kann.

Wir werden die Tools mp4info und mp4dump aus der [Bento4-Suite](#) für die [Spurlayout-](#) und [Strukturanalyse](#) und [FFmpeg verwenden](#), um die einzelnen Bilder der Videospur zu extrahieren.

Der Beispielfilm verwendet das Verpackungsformat "QuickTime" (MOV), das auf dem ISO Base Media File Format basiert - einem internationalen Standard, der allen Verpackungsformaten der MP4-Dateiformatfamilie zugrunde liegt. Dadurch ist es sehr gut mit den meisten verfügbaren Tools kompatibel und ermöglicht eine einfache Analyse.

Untersuchen wir zuerst die Gesamtstruktur der Datei. Alle Mediendateien, die auf dem ISO Base Media-Dateiformat basieren, sind als eine Hierarchie von Boxen strukturiert - ein Mini-Dateisystem. Verwenden Sie das Dienstprogramm mp4dump, um die Boxstruktur zu extrahieren, indem Sie den folgenden Befehl ausführen:

```
mp4dump tears_of_steel_720p.mov
```

Die Ausgabe wird der folgenden ähneln:

```
[ftyp] size=8+12
  major_brand = qt
  minor_version = 200
  compatible_brand = qt
[wide] size=8+0
[mdat] size=8+371579623
[moov] size=8+598972
  [mvhd] size=12+96
    timescale = 1000
    duration = 734167
    duration(ms) = 734167
  [trak] size=8+244250
```

```
[tkhd] size=12+80, flags=f
  enabled = 1
  id = 1
  duration = 734167
  width = 1280.000000
  height = 534.000000
...
```

Dies stellt die interne Struktur der Datei dar. Zum Beispiel sehen Sie hier eine *Moov*-Box mit einem 8-Byte-Header und 598972 Byte Inhalt. Dieses Feld ist ein Container für verschiedene Metadatenfelder, die den Inhalt der Datei beschreiben. Weitere Informationen zur Bedeutung der verschiedenen Boxen und ihrer Eigenschaften finden Sie unter [ISO / IEC 14496-12](https://www.iso.org/standard/54466.html) .

Die eigentlichen Medien-Samples selbst - die komprimierten Bilder und *Audiowellenformen* - werden in der *mdat*-Box gespeichert, deren Inhalt für das Dienstprogramm `mp4dump` undurchsichtig ist.

Um irrelevante Daten auszuschließen und den Analyse-Workflow zu vereinfachen - dieses Beispiel konzentriert sich auf die Videospur -, entfernen wir nun die Audiospur aus unserem Beispielfilm. Führen Sie den folgenden Befehl aus:

```
ffmpeg -i tears_of_steel_720p.mov -an -vcodec copy video_track.mov
```

Beachten Sie, dass der obige Schritt auch verschiedene benutzerdefinierte Erweiterungselemente aus dem Eingabevideo entfernt, das Wesentliche des visuellen Inhalts in eine neue Containerdatei packt und alles andere verwirft. Wenn Sie dies in einem Produktionsszenario tun, stellen Sie sicher, dass Sie wirklich alle anderen Elemente in der Eingabedatei löschen können!

Kodierte Videospuren sind eine Folge von Bildern. Mit dem hier verwendeten H.264-Codec - und allen anderen gängigen modernen Codecs - können die Bilder unterschiedlich sein:

- I-Bilder - dies sind unabhängige Bilder, die ausschließlich mit den im Bild enthaltenen Daten decodiert werden können.
- P-Bilder - diese nehmen ein anderes Bild als Grundlinie auf und wenden eine Transformation auf dieses Bild an (z. B. "Diese bestimmten Pixel um 5 Pixel nach rechts verschieben").
- B-Bilder - ähnlich wie P-Frames, aber bidirektional - sie können auch Bilder aus der Zukunft referenzieren und Transformationen definieren, wie z.

Die genaue Kombination von Bildtypen kann vom Codierungsworkflow frei gewählt werden, wodurch viele Optimierungsmöglichkeiten geschaffen werden, obwohl bestimmte Anwendungsfälle die verfügbare Flexibilität einschränken können, indem z. B. ein I-Frame in genau 2 Sekunden Intervallen vorhanden sein muss.

Führen Sie den folgenden Befehl aus, um die Bildstruktur der Videospur anzuzeigen:

```
mp4info --show-layout video_track.mov
```

Neben der Anzeige einer von Menschen lesbaren Form der gesamten Metadaten der Datei wird ein detaillierter Ausdruck des Bildlayouts der Videospur angezeigt.

```
...
00000959 [V] (1) size= 7615, offset=15483377, dts=491008 (39958 ms)
00000960 [V] (1) * size=104133, offset=15490992, dts=491520 (40000 ms)
00000961 [V] (1) size= 16168, offset=15595125, dts=492032 (40042 ms)
00000962 [V] (1) size= 4029, offset=15611293, dts=492544 (40083 ms)
00000963 [V] (1) size= 24615, offset=15615322, dts=493056 (40125 ms)
00000964 [V] (1) size= 4674, offset=15639937, dts=493568 (40167 ms)
00000965 [V] (1) size= 18451, offset=15644611, dts=494080 (40208 ms)
00000966 [V] (1) size= 95800, offset=15663062, dts=494592 (40250 ms)
00000967 [V] (1) size= 30271, offset=15758862, dts=495104 (40292 ms)
00000968 [V] (1) size= 10997, offset=15789133, dts=495616 (40333 ms)
00000969 [V] (1) size= 28458, offset=15800130, dts=496128 (40375 ms)
00000970 [V] (1) size= 9593, offset=15828588, dts=496640 (40417 ms)
00000971 [V] (1) size= 24548, offset=15838181, dts=497152 (40458 ms)
00000972 [V] (1) size= 6853, offset=15862729, dts=497664 (40500 ms)
00000973 [V] (1) size= 27698, offset=15869582, dts=498176 (40542 ms)
00000974 [V] (1) size= 7565, offset=15897280, dts=498688 (40583 ms)
00000975 [V] (1) size= 24682, offset=15904845, dts=499200 (40625 ms)
00000976 [V] (1) size= 5535, offset=15929527, dts=499712 (40667 ms)
00000977 [V] (1) size= 38360, offset=15935062, dts=500224 (40708 ms)
00000978 [V] (1) * size= 82466, offset=15973422, dts=500736 (40750 ms)
00000979 [V] (1) size= 13388, offset=16055888, dts=501248 (40792 ms)
00000980 [V] (1) size= 2315, offset=16069276, dts=501760 (40833 ms)
00000981 [V] (1) size= 21983, offset=16071591, dts=502272 (40875 ms)
00000982 [V] (1) size= 3384, offset=16093574, dts=502784 (40917 ms)
00000983 [V] (1) size= 22225, offset=16096958, dts=503296 (40958 ms)
...
```

Jede Zeile in diesem Ausdruck ist ein Bild, das in der Videospur enthalten ist. Die mit einem Stern (\*) sind I-Bilder. Sie können sehen, wie sie am größten sind, wobei die anderen eine stärkere Komprimierung ermöglichen, indem Sie auf vorhandene Bilder verweisen und nur die Unterschiede beschreiben.

Die Auflistung enthält auch den Versatz der Bilddaten in der Videodatei und den Dekodierungszeitstempel des Bildes, um eine weitere Korrelation und Analyse zu ermöglichen. Bitte beachten Sie, dass die Dekodierungsreihenfolge / das Timing der Bilder nicht unbedingt mit der Präsentationsreihenfolge / Timing identisch ist! Wenn B-Bilder im Video vorhanden sind, können sie nur *nach* Referenzbildern dekodiert werden, auch wenn sie *vor* den referenzierten Bildern dargestellt werden!

Nachdem Sie sich mit der Struktur der Videospur vertraut gemacht haben, führen Sie den folgenden Befehl aus, um die 30 Bilder ab der 40-Sekunden-Marke als PNG-Dateien zu extrahieren:

```
ffmpeg -i video_track.mov -ss 00:00:40 -vframes 30 picture%04d.png
```

Die extrahierten Bilder werden vollständig decodiert, da sie in einem Videoplayer erscheinen würden. Es ist nicht möglich (ohne extrem spezielle Werkzeuge), eine visuelle Darstellung der Rohdaten in P-Frames oder B-Frames zu erhalten.



Beobachten Sie, wie das 7. erzeugte Bild eine vollständige Szenenänderung im Video darstellt. Sie können dies leicht mit der Ausgabe von mp4info oben korrelieren - das 7. Bild ab der 40-Sekunden-Marke (Nummer 00000966) ist viel größer als das nahegelegene. Szenenwechsel sind schwer zu kodieren, da sie das gesamte Bild auffrischen und viele neue Daten enthalten. Wenn der Encoder nicht ausreichend nachgiebig ist, um für Szenenwechsel zu optimieren (dh kein großes Bild erzeugen dürfen), ist die visuelle Ausgabe bis zum nächsten I-Bild von geringer Qualität oder "blockig". Durch Untersuchen der Zuweisung von Bandbreite (Bytes) zu verschiedenen Bildern erhalten Sie Einblick in visuelle Artefakte, die plötzlich in einem Video auftreten können.

## Grundlegendes zu adaptiven DASH-Streaming-Präsentationen

DASH ist die am weitesten verbreitete adaptive Streaming-Technologie in modernen Lösungen, die zur Bereitstellung von Video in einer Vielzahl von Szenarien eingesetzt wird. Der beste Weg, um DASH-Präsentationen zu verstehen, besteht darin, die Netzwerkaktivität zu beobachten, die während der Wiedergabe stattfindet.

In diesem Beispiel wird [Fiddler](#) zum Erfassen und Analysieren des Netzwerkverkehrs im Browser verwendet. Ein ähnliches Tool ist jedoch ebenfalls ausreichend. Wir werden den Open-Source-Player [dash.js](#) für die Videowiedergabe verwenden.

Für unseren Demo-Inhalt verwenden wir die [Axinom DASH-Testvektoren](#) , speziell die 1080p-Variante des "Clear" -[Testvektors mit einer](#) Periode.



Sample Streams ▾

[http://media.axprod.net/TestVectors/v7-Clear/Manifest\\_1080p.mpd](http://media.axprod.net/TestVectors/v7-Clear/Manifest_1080p.mpd)

00:02:37.583  
Frame 3782 @ 24.00 fps  
H.264 High Profile 1920x1080 @ 3000 kbps



Öffnen Sie bei laufendem Netzwerk-Capture den [Dash.js Nightly Build-Beispielplayer](#) in einem beliebigen modernen Browser und geben Sie die URL [http://media.axprod.net/TestVectors/v6-Clear/Manifest\\_1080p.mpd](http://media.axprod.net/TestVectors/v6-Clear/Manifest_1080p.mpd) in das Textfeld ein. Drücken Sie Load, um die Wiedergabe zu starten. Sie werden feststellen, dass die folgenden Dateien heruntergeladen werden:

```
http://media.axprod.net/TestVectors/v7-Clear/Manifest_1080p.mpd
http://media.axprod.net/TestVectors/v7-Clear/2/init.mp4
http://media.axprod.net/TestVectors/v7-Clear/15/init.mp4
http://media.axprod.net/TestVectors/v7-Clear/18/init.mp4
http://media.axprod.net/TestVectors/v7-Clear/18/0001.m4s
http://media.axprod.net/TestVectors/v7-Clear/2/0001.m4s
http://media.axprod.net/TestVectors/v7-Clear/15/0001.m4s
http://media.axprod.net/TestVectors/v7-Clear/15/0002.m4s
http://media.axprod.net/TestVectors/v7-Clear/5/init.mp4
http://media.axprod.net/TestVectors/v7-Clear/5/0002.m4s
http://media.axprod.net/TestVectors/v7-Clear/18/0002.m4s
http://media.axprod.net/TestVectors/v7-Clear/5/0003.m4s
...
```

Die erste Datei ist das Präsentationsmanifest - ein XML-Dokument, dessen Format in [ISO / IEC 23009-1 definiert ist](#) . Dies beschreibt die DASH-Präsentation in einer Tiefe, die dem Spieler ermöglicht, die Wiedergabe zu verstehen.

Wenn Sie in das Manifest hineinschauen, sehen Sie verschiedene `AdaptationSet` Elemente, von denen jedes eine einzelne Anpassung des Inhalts beschreibt. Zum Beispiel gibt es einen Anpassungssatz für die Videospur, drei Anpassungssätze für drei Audiosprachen und fünf Anpassungssätze für fünf Untertitelsprachen.

Innerhalb der Anpassungssätze befinden sich `Representation` . Für den Videoanpassungssatz gibt es mehrere davon - jede Darstellung enthält denselben visuellen Inhalt, der mit einem anderen Qualitätsniveau codiert ist. Jeder Audio- und Textanpassungssatz hat nur eine Darstellung.

```
AdaptationSet [ segmentAlignment=true group=1 maxWidth=1920 maxHeight=1080 maxFrameRate=24 par=16:9 ]
  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" xmlns="urn:mpeg:dash:schema:mpd:2011" />
  <SegmentTemplate timescale="1200000" media="$RepresentationID/$Number%04d$.m4s" startNumber="1" duration="1" />
  <Representation id="8" mimeType="video/mp4" codecs="hev 1.2.4.L63.90" width="512" height="288" frameRate="24" />
  <Representation id="9" mimeType="video/mp4" codecs="hev 1.2.4.L63.90" width="640" height="360" frameRate="24" />
  <Representation id="10" mimeType="video/mp4" codecs="hev 1.2.4.L90.90" width="852" height="480" frameRate="24" />
  <Representation id="11" mimeType="video/mp4" codecs="hev 1.2.4.L93.90" width="1280" height="720" frameRate="24" />
  <Representation id="12" mimeType="video/mp4" codecs="hev 1.2.4.L120.90" width="1920" height="1080" frameRate="24" />
AdaptationSet [ segmentAlignment=true group=2 lang=en ]
  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" xmlns="urn:mpeg:dash:schema:mpd:2011" />
  <SegmentTemplate timescale="24000" media="$RepresentationID/$Number%04d$.m4s" startNumber="1" duration="1" />
  <Representation id="15" mimeType="audio/mp4" codecs="mp4a.40.29 audioSamplingRate=48000 startWithSAP=1" />
  <AudioChannelConfiguration schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="1" />
AdaptationSet [ segmentAlignment=true group=2 lang=en-AU ]
  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" xmlns="urn:mpeg:dash:schema:mpd:2011" />
  <SegmentTemplate timescale="24000" media="$RepresentationID/$Number%04d$.m4s" startNumber="1" duration="1" />
  <Representation id="16" mimeType="audio/mp4" codecs="mp4a.40.29 audioSamplingRate=48000 startWithSAP=1" />
  <AudioChannelConfiguration schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="1" />
AdaptationSet [ segmentAlignment=true group=2 lang=et-ET ]
  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" xmlns="urn:mpeg:dash:schema:mpd:2011" />
  <SegmentTemplate timescale="24000" media="$RepresentationID/$Number%04d$.m4s" startNumber="1" duration="1" />
  <Representation id="17" mimeType="audio/mp4" codecs="mp4a.40.29 audioSamplingRate=48000 startWithSAP=1" />
  <AudioChannelConfiguration schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="1" />
AdaptationSet [ segmentAlignment=true group=3 lang=en ]
  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" xmlns="urn:mpeg:dash:schema:mpd:2011" />
  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="subtitle" xmlns="urn:mpeg:dash:schema:mpd:2011" />
  <SegmentTemplate timescale="1000" media="$RepresentationID/$Number%04d$.m4s" startNumber="1" duration="1" />
  <Representation id="18" mimeType="application/mp4" codecs="wvtt" startWithSAP="1" bandwidth="428" xmlns="urn:mpeg:dash:schema:mpd:2011" />
```

Für die Wiedergabe muss ein Spieler entscheiden, welche Anpassungssätze dem Zuschauer präsentiert werden sollen. Es kann diese Entscheidung auf der Grundlage einer beliebigen benutzerdefinierten oder integrierten Geschäftslogik treffen, die es wünscht (z. B. die bevorzugte Reihenfolge der Sprache). Die Anpassungssätze, die der Inhaltsautor für primär hält, haben im Manifest ein `Role` Element, das die "Haupt" -Rolle deklariert.

Darüber hinaus muss der Spieler entscheiden, welche Repräsentation dem Zuschauer präsentiert werden soll (wenn ein Adaptionssatz mehrere Repräsentationen enthält). Die meisten Spieler beginnen konservativ und wenden einen heuristischen Algorithmus an, der versucht, das maximale Qualitätsniveau darzustellen, das die Netzwerkverbindung des Betrachters aufrechterhalten kann.

Der Spieler kann den aktiven Satz von Darstellungen und / oder Anpassungssätzen jederzeit ändern, entweder als Reaktion auf eine Benutzeraktion (Auswahl einer anderen Sprache) oder

automatisierte Logik (Bandbreitenheuristik führt zu einer Änderung des Qualitätsniveaus).

Das `SegmentTemplate` Element definiert die URL-Struktur, mit der der Player auf die verschiedenen Darstellungen zugreifen kann. Ein Schlüsselfaktor für DASH-Präsentationen ist, dass der Inhalt in kleine Abschnitte von jeweils wenigen Sekunden (4 Sekunden bei unserem Beispielfilm) aufgeteilt wird, die unabhängig heruntergeladen werden. Jede Darstellung hat auch ein Initialisierungssegment mit dem Namen "init.mp4" für diesen Beispielfilm, das eine darstellungsspezifische Decoderkonfiguration enthält und daher geladen werden muss, bevor andere Segmente dieser Darstellung verarbeitet werden können.

Das hier beschriebene Verhalten gilt für das DASH Live-Profil, die am häufigsten verwendete Variante von DASH. Es gibt auch andere Profile mit leicht unterschiedlichem Verhalten, die hier nicht behandelt werden. Achten Sie auf das Attribut "profile" im DASH-Manifest-Stammelement, um sicherzustellen, dass diese Beschreibung für Ihre Videos gilt!

Wenn Sie die Liste der URLs prüfen, die Sie aus der Erfassung des Netzwerkverkehrs erhalten, und mit den Informationen des Manifests vergleichen, werden Sie zu dem Schluss kommen, dass der Spieler nach dem Herunterladen des Manifests die folgenden Aktionen ausgeführt hat:

1. Laden Sie die Initialisierungssegmente für Repräsentationen 2 (360p-Video), 15 (englisches Audio) und 18 (englische Untertitel) herunter.
2. Laden Sie die ersten Segmente der obigen drei Darstellungen (0001.m4s) herunter.
3. Laden Sie das zweite Segment der Audiodarstellung herunter.
4. Wechseln Sie ab dem zweiten Videosegment zum 1080p-Videostream! Dies wird durch Herunterladen des Initialisierungssegments und des zweiten Repräsentationssegments 5 (1080p-Video) angezeigt.
5. Laden Sie weitere Segmente der aktiven Repräsentationen herunter.

Durch Beobachten der Netzwerkaktivität können Sie leicht die Entscheidungen beobachten, die ein adaptiver Streaming-Player von DASH im Betrieb trifft. Ein solcher Player ist einfach ein Mechanismus, der Segmente verschiedener Tracks herunterlädt und diese nacheinander an eine Medienwiedergabemaschine übermittelt, wobei die Tracks entsprechend umgeschaltet werden.

Die [Axinom DASH-Testvektoren](#) enthalten auch Archivdateien, mit denen Sie die gesamte Präsentation für die Analyse auf Dateisystemebene herunterladen können. Sie werden feststellen, dass die Dateien auf der Festplatte genau so sind wie auf der Netzwerkebene. Dies bedeutet, dass DASH-Präsentationen von beliebigen HTTP-Servern bereitgestellt werden können, ohne dass eine benutzerdefinierte serverseitige Logik erforderlich ist.

Ein Aspekt des Live-Profil-DASH, der die Analyse kompliziert, besteht darin, dass die Medienproben über eine große Anzahl von Segmenten verteilt sind. Die meisten Medienanalyse-Tools können keine einzelnen Segmente verarbeiten und arbeiten nur für ganze Spuren. Sie können diese Einschränkung oft überwinden, indem Sie einfach die Segmente einer einzelnen Darstellung verketteten, beginnend mit dem Initialisierungssegment. Unter Windows können Sie beispielsweise den folgenden Befehl verwenden:

```
copy init.mp4 /b + 0001.mp4 /b + 0002.mp4 /b + 0003.mp4 /b track.mp4
```

Dadurch wird eine track.mp4-Datei erstellt, die die ersten drei Mediensegmente einer Repräsentation enthält. Eine solche Datei ist zwar nicht mit einer eigenständigen MP4-Datei identisch, kann jedoch von den meisten Tools (wie mp4info und FFmpeg) ohne erheblichen Funktionsverlust analysiert werden.

Medienpräsentation verstehen online lesen:

<https://riptutorial.com/de/video/topic/5879/medienpräsentation-verstehen>



# Kapitel 3: Video-Seitenverhältnisse

## Bemerkungen

Seitenverhältnisse werden oft als Breiten- / Höhenverhältnis ausgedrückt, das häufig - aber nicht immer - vereinfacht wird und manchmal auch als einfache Gleitkommazahl.

Alle folgenden Seitenverhältnisse haben denselben Wert, der auf unterschiedliche Weise ausgedrückt wird:

- 1280: 720
- 16: 9
- 1.77777777777777777777777777777778

## Examples

### Seitenverhältnis anzeigen (DAR)



720

1280

Display Aspect Ratio (DAR) 16:9

Dies ist ein Screenshot eines Videos, das abgespielt wird. Sie sehen ein normales 16: 9-Video, wie Sie es von einer modernen Videolösung erwarten würden. Dieses - das Seitenverhältnis, das der Betrachter sieht - wird als *Anzeige-Seitenverhältnis* oder DAR bezeichnet.

Aus den dargestellten Parametern ergibt sich, dass  $DAR = 1280: 720 = 16: 9 = 1.77777777777777777777777777777778$ .

## Bildseitenverhältnis (PAR)

Intern sind alle Videos nur eine Bilderserie. Schauen wir uns ein solches Bild an.



720

Picture Aspect Ratio (PAR) 9:16

Das sieht seltsam aus, oder? Tatsächlich. Die Bilder, aus denen ein Video besteht, können ein Seitenverhältnis aufweisen, das sich von der DAR unterscheidet, meist aus algorithmischen Gründen (z. B. können nur Größen, die ein Vielfaches von 16 sind, durch den gewählten Algorithmus komprimiert werden). Dies wird als *Bildseitenverhältnis* oder PAR bezeichnet.

In diesem Beispiel haben wir die Bildabmessungen genau umgekehrt zwischen der angezeigten Form (16: 9) und dem tatsächlichen Bild, sodass der PAR-Wert 9:16 beträgt. Normalerweise sind die Unterschiede geringer, aber dieses Beispiel ist zur Verdeutlichung übertrieben.

Aus den dargestellten Parametern sehen wir, dass  $PAR = 720: 1280 = 9:16 = 0,5625$

## Beispieleseitenverhältnis (SAR)

Wie das Beispiel für das *Bildseitenverhältnis* zeigt, handelt es sich bei Videos um Serien von Bildern, die nicht unbedingt das gleiche Seitenverhältnis wie das für den Benutzer anzuzeigende Endergebnis haben.

Wie kommen Sie also von diesen gestreckten Bildern zur normalerweise angezeigten Ausgabe? Du brauchst einen Dehnfaktor! Dieser Streckungsfaktor wird auf das Bild angewendet, um es auf das richtige Seitenverhältnis für die Anzeige zu bringen. Dies ist das *Beispieleseitenverhältnis* oder SAR.



1280

Sample Aspect Ratio (SAR) 81:256



1280

720

Picture Aspect Ratio (PAR) 9:16

Der Streckungsfaktor wird oft als Verhältnis von zwei ganzen Zahlen ausgedrückt. Sie berechnen es als  $SAR = PAR / DAR$ .

Die dargestellten Parameter zeigen, dass  $SAR = 9:16 / 16:9 = (9/16) / (16/9) = 81/256 = 3,1604938271604938271604938271605$

## Pixel-Seitenverhältnis

Dies ist ein anderer Name für *Beispielseitenverhältnis* und sollte vermieden werden, da das natürliche Akronym (PAR) mit dem *Bildseitenverhältnis* in Konflikt steht.

Video-Seitenverhältnisse online lesen: <https://riptutorial.com/de/video/topic/5713/video-seitenverhaeltnisse>

---

# Credits

S. No	Kapitel	Contributors
1	Erste Schritte mit Video	<a href="#">Community</a> , <a href="#">Sander</a>
2	Medienpräsentation verstehen	<a href="#">Sander</a>
3	Video-Seitenverhältnisse	<a href="#">Sander</a>