Freie Universität Berlin
Institut für Informatik

Kurs: Digitales Video

Ausarbeitung:

Der Dirac-Wavelet-Codec und automatisierter Qualitätsvergleich mit Hilfe des GStreamer-Frameworks

Kai Lüke
kai.lueke@fu-berlin.de

Dozent: Christian Zick

Wintersemester 2011/2012

Kurzfassung

Wavelet-Codecs werden am Beispiel von Dirac (BBC Research) vorgestellt. Es existieren „objektive“ Qualitätswerte für Videokompression, sodass die Qualität von Codecs im Bezug zur Dateigröße in einer Statistik dargestellt werden kann. Dafür wurde das GStreamer-Multimediaframework verwendet.

Inhaltsverzeichnis

1 Einleitung

Die Wahl eines Videocodecs fällt bei der großen Menge an Möglichkeiten keineswegs leicht. Neben der Qualität und Performance gilt es auch die Unterstützung, Lizenzprobleme und zukünftige Verfügbarkeit zu berücksichtigen. Der wenig bekannte Dirac-Codec des BBC-Research-Departments ist nicht nur frei verfügbar, sondern auch technisch ein Sonderling neben allen DCT-Codecs, da er auf der Diskreten Wavelettransformation beruht. Ob diese die ihr zugesprochene bessere Kompression als DCT auch praktisch sichtbar macht, glaubte ich vor ca. drei Jahren beantwortet, als ich kurz nach Verfügbarkeit des Codecs eine 720p-Dirac-Datei in Filmlänge auf DVD-Größe in guter Qualität reduzierte. Doch einfache Aussagen können nur schwer getroffen werden und der subjektive Eindruck ist nicht bei allen Menschen gleich. Der Dirac-Codec soll hier vorgestellt werden, jedoch nicht in allen seinen Einzelheiten, sondern durch charakteristische Ausschnitte aus einem komprimierten Video und im Vergleich zu anderen Codecs.

Um Werte zu haben, die die Qualität beschreiben und somit eine vermeintlich feste Basis zum Codec-Vergleich bieten, wurde eine Webapplikation erstellt, die als Testframework dient. Als Vergleichswert wurde das SSIM-Verfahren gewählt. Die anhand einer kurzen Videosequenz erzielten Ergebnisse für einige Codecs werden vorgestellt und ausgewertet. Wie auch andere Vergleiche[1] vor den diesigen zeigen, sind Wavelet-Codecs (noch) nicht auf dem Level von modernen MPEG/H.264-Encodern oder VP8 angelangt. Die in Python durch GStreamer und Django entwickelte Webapplikation ist bedauerlicherweise noch in einem experimentellen Status, sodass der Quellcode in seiner Veröffentlichung kaum Nutzen erbringt. Jedoch sind die fehlenden Aufgaben nicht überwältigend komplex, sodass in einigen Monaten hier mit einer Ankündigung zur Veröffentlichung zu rechnen ist. Für Fragen, Materialanforderungen und ähnliches stehe ich gerne zur Verfügung.

1 Z.B. in http://wweb.uta.edu/ee/Dip/Courses/EE5359/dirac3.pdf bzw. http://www-ee.uta.edu/Dip/Courses/EE5359/ArunaRavi_MSThesis.pdf oder http://etill.net/projects/dirac_theora_evaluation/include/halbach-2009-dirac_theora-paper.pdf (Theora falsch) und http://keyj.emphy.de/video-encoder-comparison/ (nur dirac-research, eigenartige Optionen) wobei in http://hirntier.blogspot.de/2010/03/libtheora-vs-libschrodinger-vs-x264.html Dirac einen besseren SSIM-Wert erzielt. Sichttest hier: http://www.saintdevelopment.com/media/

2 Wavelet

2.1 Mathematisch[2]

Das Integral einer Wavelet-Funktion hat den Wert 0. Somit muss der Graph nach einer Steigung, die das Integral über 0 wachsen ließ, irgendwann wieder abfallen und grob betrachtet ergeben sich so Wellen. Es gibt viele verschiedene charakteristische Wavelets, z.B. Coiflet- und Daubechies-Wavelets. Die Wavelet-Transformation ähnelt der Short-Time-Fourier-Transformation, hat aber Vorteile in der Auflösung von Zeit und Frequenzbereich. Nach der Wahl eines „Mutter-Wavelets“ wird durch Überlagerung der passend gestauchten, gestreckten und verschobenen Versionen das Eingabesignal rekonstruiert. Es ergeben sich viele Koeffizienten, die die einzelnen Bestandteile beschreiben, wobei manche weniger wichtig sind als andere. Diese können weggelassen werden, wenn eine Annäherung genügt.

2.2 Anwendung in der Videokompression

Wie bei der in der Videokompression üblichen Diskreten Kosinustransformation (DCT), kann die Diskrete Wavelettransformation (DWT) im zweidimensionalen Raum der Bildpixel durchgeführt werden (meist wird das Bild in Blöcke unterteilt), wobei sich so viele Koeffizienten wie Pixel im Block ergeben, die verschieden großen Informationsgehalt haben und quantisiert werden können.[3]

Während die Änderung eines DCT-Koeffizienten alle Pixel beeinflusst, bestimmt ein DWT-Koeffizient lokale Muster des Blockes (nämlich dort, wo die Frequenz/das Signal bezüglich zum Koeffizienten auftaucht). DWT lohnt sich also über eine größere Menge, um Redundantes zu erfassen, was zu überlappenden Blöcken mit variabler Größe führt. Die Grenzen der DCT-Blöcke sind hingegen scharf (deshalb gibt es Deblocking-Filter für weichere Blockgrenzen) und bilden Artefakte, die wiederum die Bildqualität in manchen Fällen positiv beeinflussen, wenn es um harte Kanten geht. Wavelet-Codecs wird somit eine typische Weichzeichnung nachgesagt. Der Unterschied bringt neben Vorteilen auch neue Probleme für Encoder mit sich, die teilweise noch offen sind.[4]

---

2 Vgl. Prof. Dr. A. Stoffel. http://alex.nt.fh-koeln.de/wavelet.html abgerufen im Februar 2012
3 Nach http://de.wikipedia.org/w/index.php?title=Wavelet-Kompression&oldid=97707426
4 Eine Meinung über Probleme von Wavelet-Codecs: http://x264dev.multimedia.cx/archives/317

2.3 Geschichte der Wavelet-Codecs

Wavelet wurde in der Bildkompression erst ab ca. 1980 eingesetzt, die Anzahl der standardisierten Formate ist dennoch überschaubar. In Bildformaten wurde die Wavelet-Transformation in DjVuPhoto, JPEG 2000, MrSID und PGF (2002) genutzt. Als Videocodecs waren CineForm RAW (2002, proprietär) und Rududu (Open Source, Entwicklung stockte) populär und REDCODE RAW wurde in Kameras der Entwicklerfirma eingesetzt. Bevor das Theora-Projekt entstand, wurde bei Xiph.org an Tarkin gearbeitet, einem 3D-DWT-Codec (also auch Zeit!), wobei viel versprechende Ergebnisse mit diesem neuartigen Ansatz erreicht wurden. Doch die Entwicklung wurde zugunsten von Theora eingestellt. Ab 2004 wurde im FFmpeg-Projekt der Snow-Codec entwickelt, der wie Dirac (auch ab 2003/2004, BBC Research) verlustfreie und verlustbehaftete Kompression unterstützt. Er hat eine geringere Komplexität als Dirac und als Besonderheit die Nutzung der patentfreien Bereichskodierung (Dirac benutzt u.a. arithmetisches Kodieren). Zur jetzigen Aktivität der Entwicklung von Snow sind nicht viele Informationen erhältlich.

3 Der Dirac-Videocodec

3.1 Entstehung und Bedeutung

2004 kündigte BBC Research & Development den Dirac-Codec[5] an, welcher 2008 unter der Version 1.0 fertiggestellt wurde. Thomas Davies war für den Algorithmus verantwortlich, der namentliche Bezug zum Physiker Paul Dirac hat keine tragende Bedeutung. Zu einer Zeit, wo MPEG2 weit verbreitet war, Theora sich in starker Entwicklung befand und zu MPEG4-ASP und den Anfängen von H.264 kaum patentfreie, nicht gebührenpflichtige Alternativen bereit standen, sollte Dirac eine hochqualitative, freie Lösung bieten, die über alle Anwendungsbereiche gut skaliert. So sind keine Patente von BBC auf den Standard angemeldet (bzw. werden nicht in Anspruch genommen), welcher – da auf Wavelets basierend und von Grund auf neu entwickelt – auch keine bestehenden tangiert. Mit einer Halbierung der Datenrate im Vergleich zu MPEG-2 im HD-Bereich wurde Dirac beworben.[6] Im Hinblick auf die BBC-interne Benutzung im Umstieg auf HD-Übertragung soll es Dirac möglich

5 Mehr in WHP 124 und WHP 117. BBC R&D White Paper. September 2005.
6 Dirac Overview. BBC R&D. http://www.bbc.co.uk/rd/projects/dirac/overview.shtml Feb. 2012.

machen, HD-Signale über die bestehenden SD-Leitungen zu senden. Auch 2008 bei den Olympischen Spielen in Běijīng wurde Dirac Pro zur Übertragung eingesetzt.

Den anfänglichen Erwartungen – gerade aus der Open-Source-Szene – als gleichwertige Alternative zu H.264 konnte Dirac nur wenig gerecht werden und verschwand im Zuge der Zuspitzung um die Auswahl des HTML5-Videostandards. Gerade das Auftreten des VP8-Codes, welcher durch Google unter freier Lizenz und ohne Patentansprüche veröffentlicht wurde und auch zu Theoras gesunkener Bedeutung beitrug, führte zu einer stagnierenden Entwicklung der Dirac-Implementierungen.

3.2 Standard

Dirac basiert auf der Diskreten Wavelet Transformation (DWT) und bietet verschiedene Wavelet-Filter an, bricht aber nicht mit dem traditionellen Codecaufbau. Der Vorteil gegenüber DCT-basierten Codecs ist die Möglichkeit der variablen Blockgrößen, die auch Überlagerungen erlauben, um Redundantes besser zu erfassen. Dies schlägt sich im „Overlapped Block Motion Compensation“-Verfahren nieder (OBMC), aber auch die Kombinationen der Motion Compensation sind flexibler als bei MPEG-4 mit Group of Pictures (GOP) üblich.[7] Jeder Frame hat eine Indexnummer und hält das Offset des nächsten Frames bereit.

Zur Entropiekodierung wird Exponential-Golomb coding und Arithmetisches Kodieren eingesetzt (letzteres nicht im Low-Delay-Modus[8]). Das neue YCoCg-Farbmodell findet Verwendung und bietet so RBG-Unterstützung. Es gibt nahezu keine Limitierungen der Auflösung, auch frei gewählte sowie 4k-Auflösung und sogar Ultra HDTV (7680 × 4320) sind möglich. Zur Archivierung existiert ein Modus zur verlustfreien Komprimierung, die eine deutliche Verbesserung gegenüber HuffYUV ist (in eigenen Tests ca. 43% der Dateigröße von HuffYUV bei 360p, 5 Sek. und 20 fps).

Die Spezifikation Dirac 1.0 beschreibt den Dirac Pro-Standard, welcher nur Intra-Frames benutzt. Dadurch sind Schnitte jederzeit möglich ohne den bekannten Blockaufbau-Effekt beim Spulen. Er ist für Low-Latency-Direktübertragungen und verlustfreie Archivierungen gedacht und komprimiert die Daten zu etwa der Hälfte

7 Dirac Specification. Version 2.2.3. 23. Sept. 2008. Abschnitt 15.8.1. http://diracvideo.org/download/specification/dirac-spec-latest.pdf sowie The technology behind Dirac. Zeile 28. http://www.bbc.co.uk/rd/projects/dirac/technology.shtml (abgerufen im Feb. 2012)

8 Dirac Specification. Ebd. Abschnitt 13.5 Punkt 2.

oder einem Viertel. Durch die SMPTE ist er unter dem Namen VC-2 standardisiert.

3.3 Implementierungen und Verbreitung

Für Dirac Pro existieren Hardwareimplementierungen, die in Echtzeit umsetzen.[9] Als Referenzimplementierung in C++ existiert dirac-research unter einer freien Lizenz, seit 2009 gab es jedoch keine weiteren Änderungen. Auf Schrödinger, einer in C geschriebenen Implementierung, liegt der Hauptfokus. Qualität und Geschwindigkeit übertreffen dirac-research, aber auch deren Entwicklung schreitet zur Zeit eher langsam voran. Sie ist jedoch verlässlich in Multimediaframeworks wie FFmpeg und GStreamer und Stand-alone-Player wie VLC und Mplayer eingebunden und dadurch in Desktop-Applikationen der gängigen Linuxdistributionen in gleichem Maße einsatzbereit wie z.B. x264, Theora und VP8. Es existieren Backends für die Benutzung der GPU anstelle der CPU (Standard), sodass diese die Geschwindigkeit meist steigern könnte. Zur Zeit werden beide Encoder zusammen hauptsächlich von David Schleef und wenigen anderen nicht vollzeitlich verwaltet.[10]

4 Objektive Qualitätsvergleiche

Da die subjektive gesehene Qualität von verschiedenen Faktoren der Betrachtung abhängt, werden mathematische Verfahren der Bewertung des Unterschieds von Originalbild und verlustbehaftetem komprimiertem Bild genutzt. Etabliert hat sich PSNR (peak signal-to-noise), wobei die mittlere quadratische Abweichung (Mean Square Error: MSE) aller Farbkomponenten jedes Bildpixels zu dem der Eingabequelle bestimmt wird. Da das menschliche Sehverhalten eher veränderte Strukturen als Weichzeichnung und Farbrauschen wahrnimmt (jede noch so kleine Farbveränderung geht in den PSNR ein), wurde 2004 der SSIM-Wert (Structural SIMilarity) eingeführt. Der SSIM-Wert beträgt bei identischen Bildern 1.0 geht gegen 0.0 je größer der Unterschied ist, die Bestimmung ist rechenintensiv.[11]

Die Qualität eines Codecs ist verglichen zu einem anderen also besser, wenn bei gleicher Dateigröße ein höherer SSIM-Durchschnitt erreicht wird. Dies kann in einem

9 Produktliste unter http://www.numediatechnology.com/diracpro.html

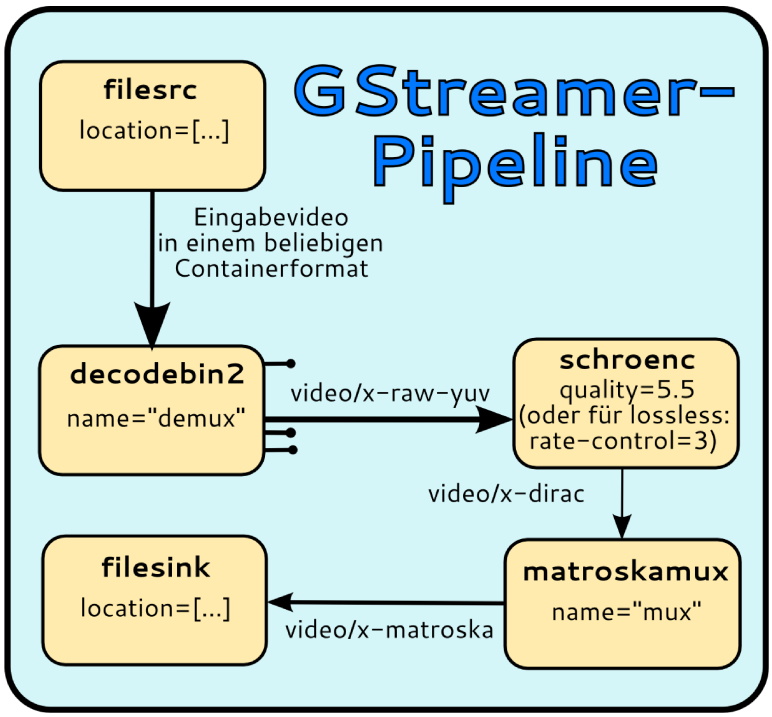
10 Auf http://diracvideo.org/ wird über die Entwicklung berichtet. Letzte Veröffentlichungen sind 2009: dirac-research-1.0.2, 2010: Schrödinger-1.0.10, 2012: Schrödinger-1.0.11

11 Veröffentlicht in Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. In: IEEE Transactions on Image Processing, vol. 13, no. 4, Apr. 2004.

Graphen abgebildet werden, der auf der x-Achse die Dateigröße und auf der y-Achse den SSIM-Wert aufzeigt.

5 Automatisierte Testumgebung

Um solche Graphen zu erstellen, müssen die Qualitätseinstellungen für die zu testenden Codecs von gut bis schlecht durchlaufen werden und für die resultierende Datei der SSIM-Wert bestimmt werden. Dieser enorme Rechenaufwand kann reduziert werden durch größere Schrittweiten in der Qualitätseinstellung und das Beschränken auf wenige Beispielframes für die SSIM-Bestimmung. Das GStreamer-Framework wurde für das Kodieren benutzt. Da jedoch (noch) kein Gstreamer-Element für die SSIM-Berechnung aller Codecs existiert, wurde auf eine C#-Implementierung des SSIM-Verfahrens zurückgegriffen, welche zwei PNG-Dateien als Parameter erwartet. Anfängliche Shell-Scripts kamen bezüglich der Datenverarbeitung an ihr Limit, sodass eine Webapplikation in Python mit dem Django-Framework und einer RabbitMQ-Schnittstelle zum Celery-Lastenverteiler erstellt wurde, die eine SQLite-Datenbank verwendet.

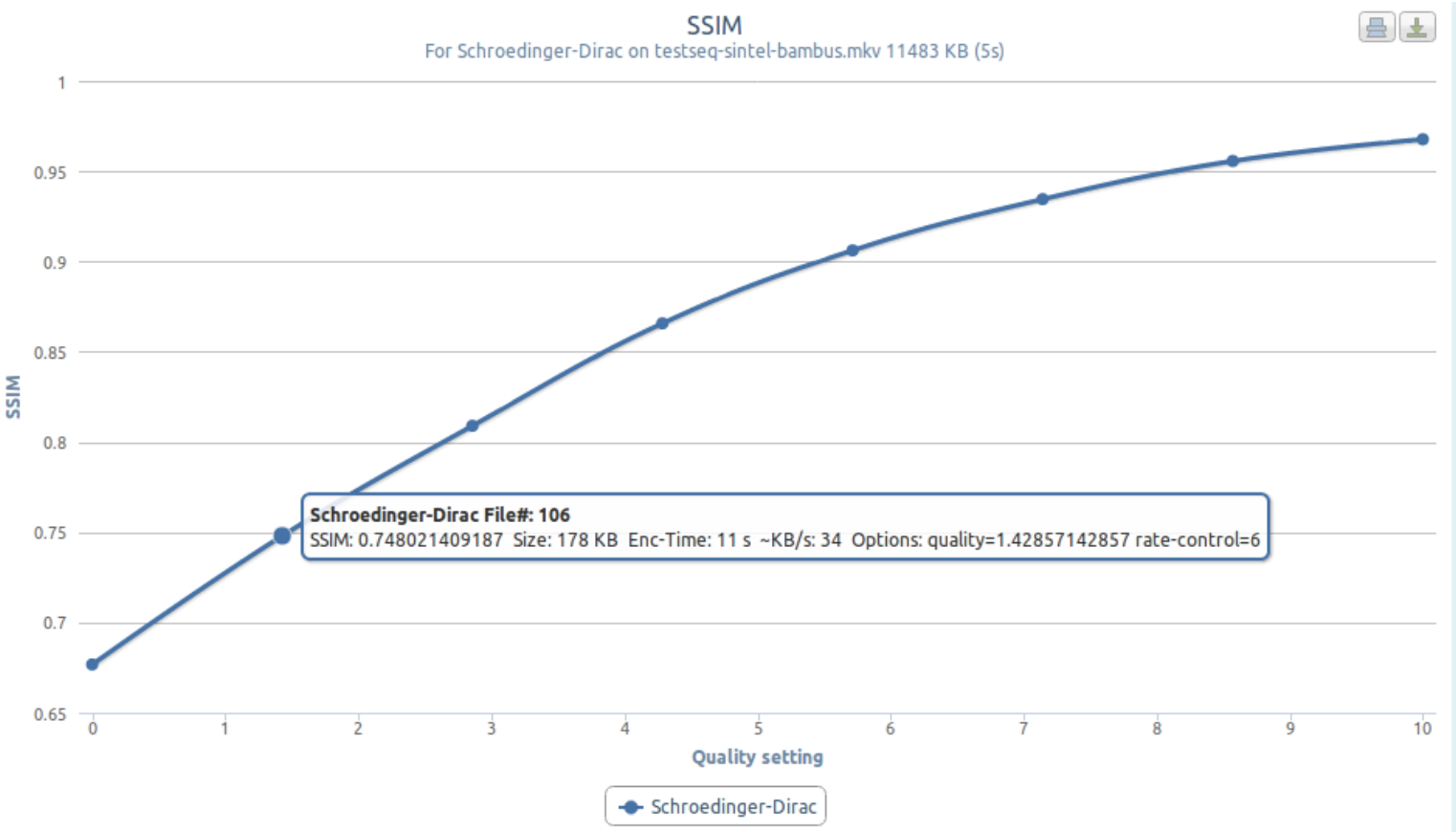


Links die zugrunde liegende modulare Struktur von GStreamer am Beispiel einer Pipeline zum Umwandeln in Dirac in einem Matroska-Container. Jede mögliche Audiospur oder sonstige weitere Quellen aus dem Demuxing werden weggelassen.

Funktion:

Nachdem eine am Besten verlustfrei vorliegende Eingabevideodatei hochgeladen wurde, kann ein Auftrag zum Kodieren mit einem Codec gestellt werden. Dabei ist es möglich, zwischen statischen Optionen und der durchgehend geänderten Qualitätsoption zu unterscheiden. Auch die Anzahl der zu erstellenden Dateien und

somit die Schrittweite kann ebenso wie die Start- und Endwerte der Qualität angegeben werden. Nachdem dies für die zu vergleichenden Codecs auf der Eingabedatei gemacht wurde und die Verarbeitung abgeschlossen ist, können die entstandenen („Roh“-)Daten betrachtet werden. Die erzeugten Videos werden mit Größe, Kodierdauer, angewandten Optionen und SSIM-Durchschnittswert aufgelistet. Für jeden Codec-Auftrag ist auch ein SSIM-Graph verfügbar, der den SSIM-Wert im Bezug zur Qualitätseinstellung abbildet. Die Qualitätseinstellung des Schrödinger-Dirac-Encoders ist eine Gleitkommazahl zwischen 0 und 10:



Hier wurde die Berechnung des SSIM nur auf Basis von 10 Frames des kurzen Eingabevideos bestimmt (Grafik also keineswegs allgemeingültig). Durch das Ansteuern des zweiten Datenpunktes ist eine Hover-Information zur Datei sichtbar. Auch die Option zum Abspielen und genaueren Untersuchen der Datei wird eingeblendet (hier nicht sichtbar).

Jede Datei kann auch (je nach Browser- oder Pluginunterstützung) abgespielt werden. Die Frames sind in einer weiteren Ansicht nach Millisekunden anzusteuern, wobei zwischen Original und kodiertem Bild gewechselt werden kann. Auch die Differenz beider ist anzeigbar, sowie der SSIM-Wert.

Doch der Hauptfokus liegt auf den generierten Graphen, die anfangs beschrieben

wurden. Für jedes Eingabevideo werde alle gewählten Codecs durch ihre Kurven abgebildet. Es ist so schnell sichtbar, ob die Qualität (laut SSIM-Wert) eines Codec andere übertrifft (Kurve liegt höher) und wie klein oder groß die Ausgabevideos sind.

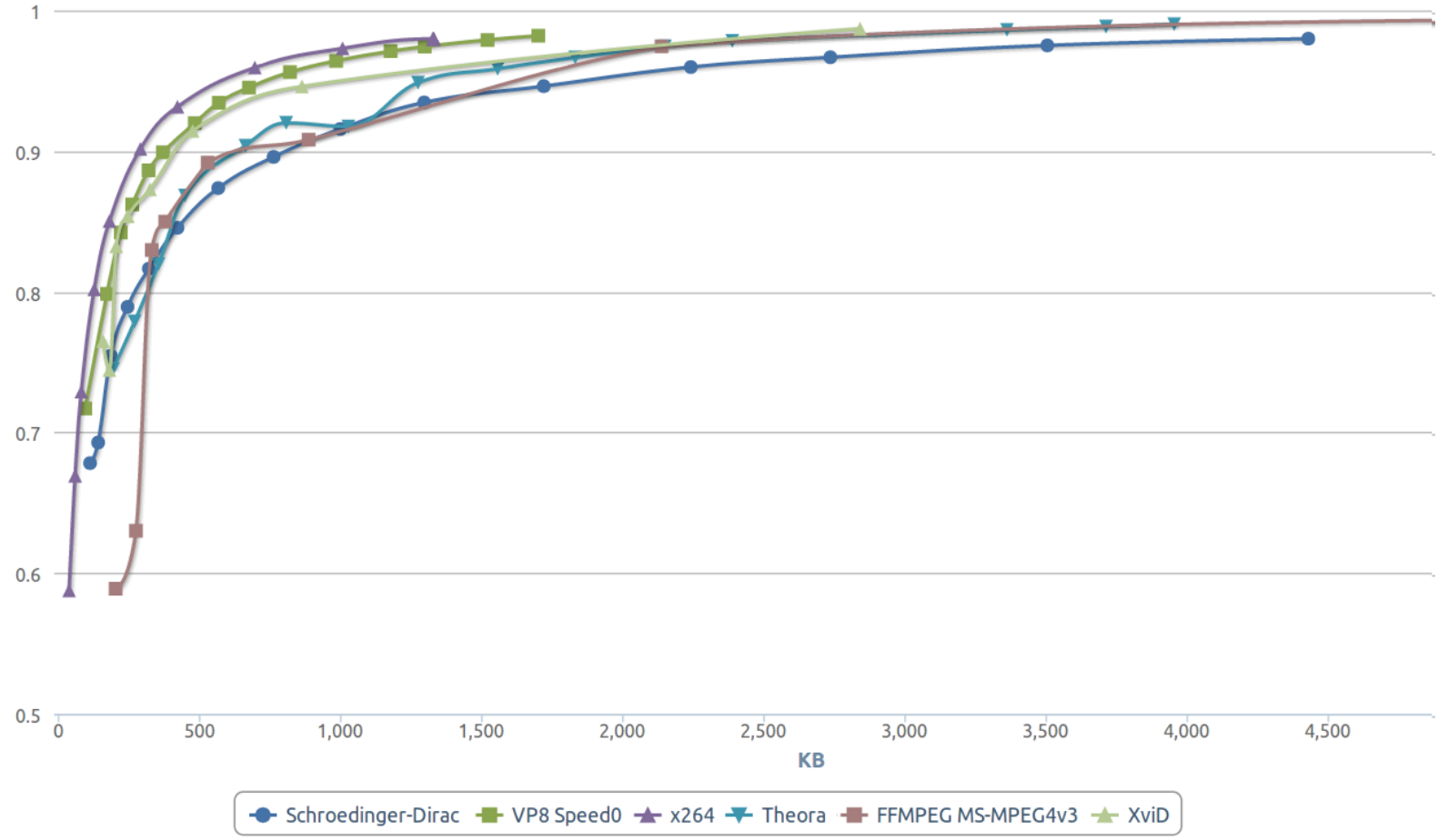
6 Auswertung anhand eines Beispiels

6.1 Testvoraussetzung und -ablauf

Als Testsequenz wurde aus dem Film *Sintel*[12] ein kurzer HD-Ausschnitt mit zwei Bildwechseln gewählt und durch Schödinger im Lossless-Modus per GStreamer in eine Dirac-Datei im Matroska-Container umgewandelt nach Herunterrechnung auf 640×360 Pixel bei 20 Frames/s. Auch FFmpeg, MPlayer und VLC geben sie korrekt wieder. Die Länge beträgt 5 Sekunden, die Größe 12 MB. Eine Person bewegt sich durch einen Bambuswald, ein Schmetterling fliegt.

Die Encoder x264 (H.264), Schrödinger (Dirac), libtheora (Theora), libvpx (VP8), FFmpeg-MSMPEG4 (Microsoft MPEG4 v3) wurden durch ihre jeweiligen GStreamer-Elemente benutzt, wobei nicht die Bitrate als Qualitätseinstellung, sondern der entsprechende Qualitätsparameter variiert wurde (z.B. alle Quantisierungswerte durchlaufen). Die Bitrate zeigte sich in ersten Tests als wenig verlässlich und die Qualitätsergebnisse sind schwer auf Eingabematerial mit anderer Framerate oder Auflösung zu übertragen. Die Unterstützung für den immer noch experimentellen Snow-Wavelet-Encoder (FFmpeg) in GStreamer ist noch unvollständig, sodass er leider nicht mit einbezogen werden konnte. Der Test ergab folgendes Ergebnis für den SSIM-Wert, das je nach Eingabevideo, Schrittweite und Versionsnummern der Encoder gewiss abweichen kann:

12 Verfügbar unter http://sintel.org/ (Creative Commons Attribution license)

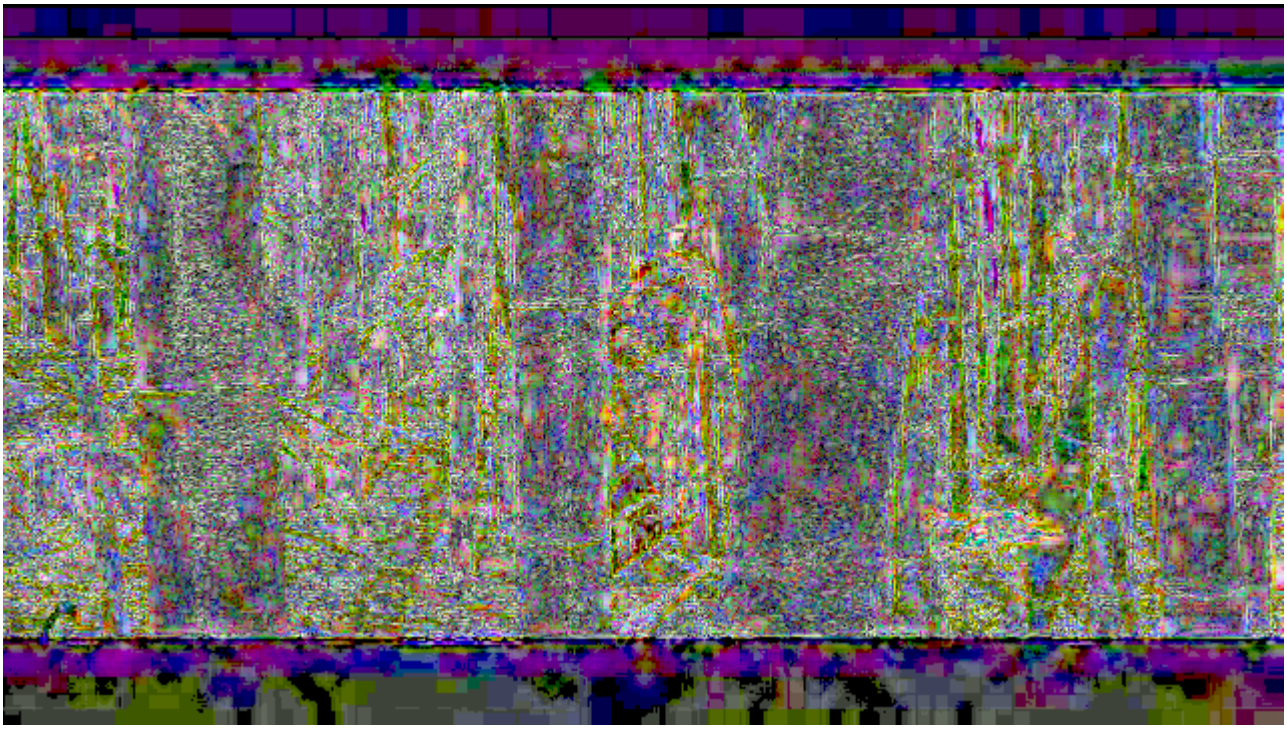


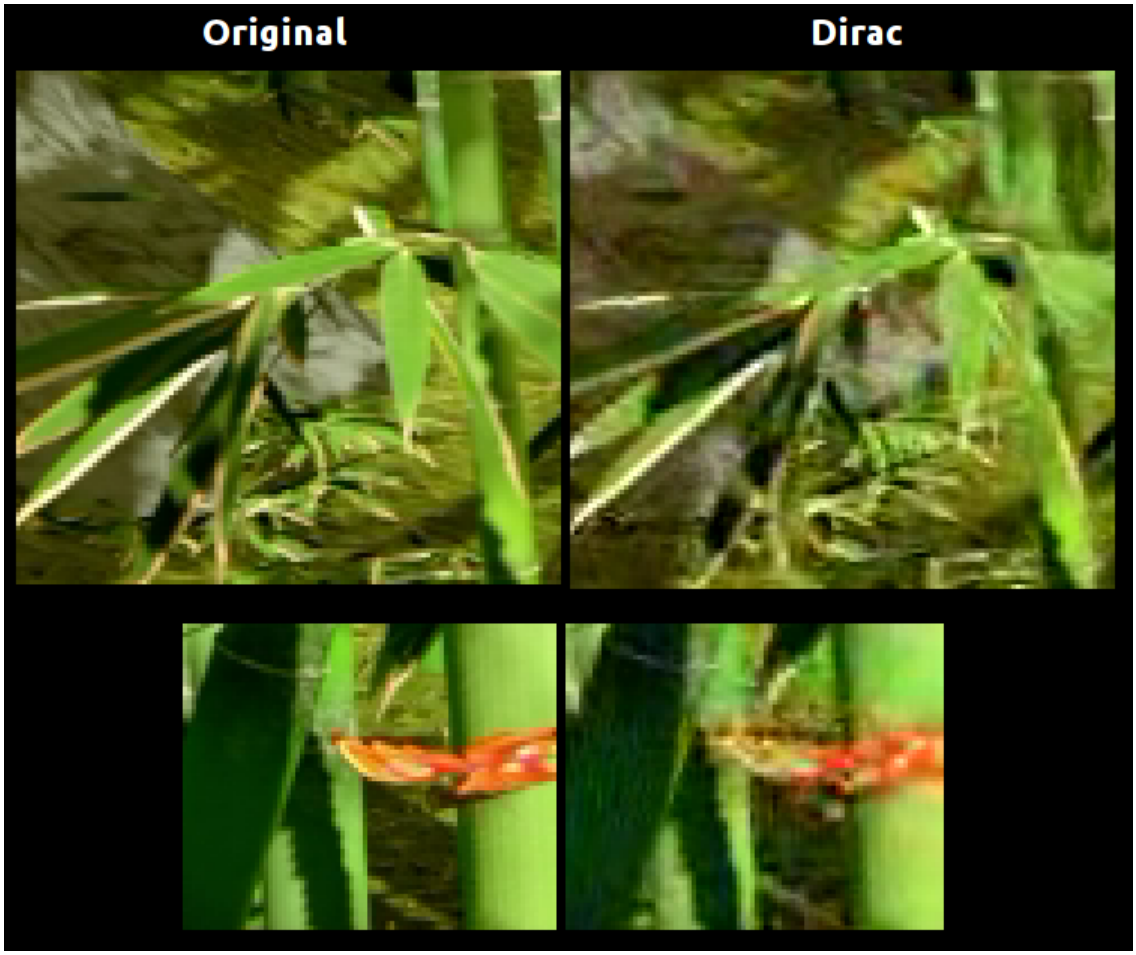
Es ist zu sehen, dass x264 nicht nur die kleinsten Dateien erzeugt, sondern auch immer den höchsten SSIM-Durchschnittswert über alle Frames hat. Die unterste Kurve bildet Schrödinger ab und ffmpeg-MS-MPEG liegt etwas darüber, wiederum nah unter Theora. Schrödinger scheint aber in kleineren Dateigrößen beide leicht zu übertreffen. Die Kurve von XviD schneidet an Anfang und Ende Theora, hebt sich jedoch in der Mitte ab und teilt einige Punkte mit VP8. Dessen Kurve liegt knapp unter der von x264, berührt sie teilweise. Interessant ist, dass VP8 bei schlechter Qualität comic-artige Simplifizierungen zeigt und wie ein Ölgemälde aussieht. x264 neigt zu eckigen/gezackten Kanten.

Da der Focus auf Dirac liegt, wurden einige Ausschnitte aus einer mit Schrödinger entstandenen Datei niedriger Qualität ausgewählt, um Auffälligkeiten des Codecs zu demonstrieren. Es folgt das Originalbild der Eingabe. Der Boden ist grün, am Bambus können feine Rillen gesehen werden, die Bewegungen der Person sind leicht verwischt, der Schmetterling deutlich und die Balken am oberen und unteren Bildrand schwarz.

Im komprimierten Bild (unten, SSIM: 0,86834214037) sind nicht die typischen DCT-Blöcke zu sehen, sondern unruhige wellenartige Streifen. Der weißliche Hintergrund ist verschwommen, ebenso z.B. das Muster des Bodens und die Kleidung. Bei näherer Betrachtung fällt auf, dass Farbfehler vorkommen und der Boden somit bräunlich, die Kleidung violett und der Bambus farbloser und teilweise bläulich erscheint. Die Farben des Bildes laufen auch in die schwarzen Balken aus. Sogar am Rand sind sie nicht durchgängig schwarz, sondern gemustert.

Die Differenz beider Bilder lässt erahnen, welche Informationen verloren gingen. Um es zu verdeutlichen, wurde eine „automatische Tonwertkorrektur“ angewandt, welche (zweites Bild) auch die Veränderungen in den schwarzen Balken verstärkt.

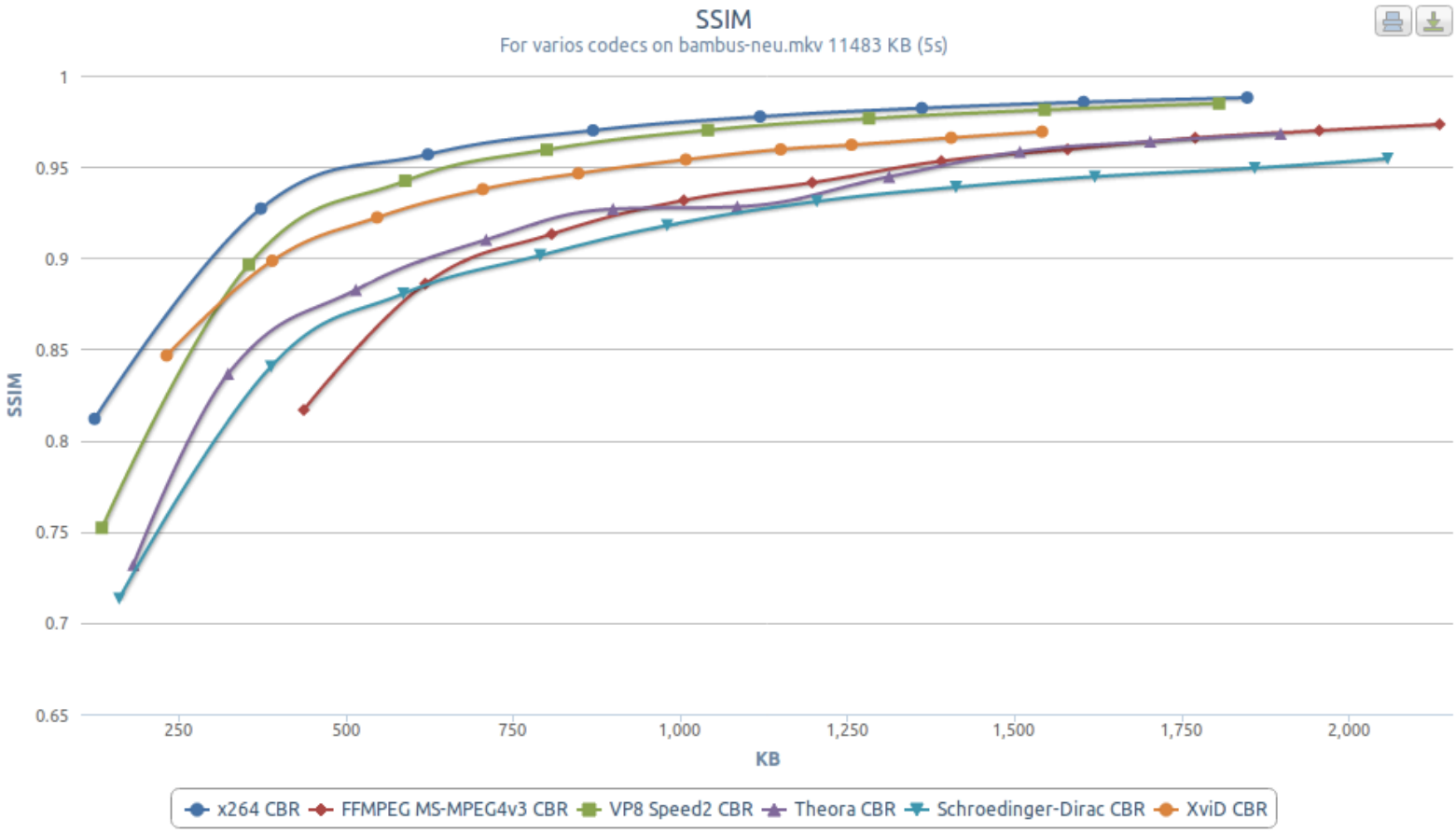




Auf dem oberen rechten Ausschnitt eines anderen Frames wird die braun-violette Verfärbung deutlich, rechts unten ist links vom Schmetterling ein Rauschen auf der sonst eintönigen Fläche zu sehen. Die Struktur der Flügel wirkt wie ausgefranst.

Nutzung von CBR: Obwohl bei einigen Codecs die Bitrate nicht bevorzugter Qualitätsparameter ist, manche mit einer konstanten Bitrate nur schwerfällig arbeiteten und bei gleicher Einstellung die Datengröße der verschiedenen Codecs enorm schwankte, wurde ein Testlauf mit CBR durchgeführt.

Die Geschwindigkeit des libvpx-Encoders wurde auf 2 gestellt von 0 (langsam) bist 7 (schnell). Zu beachten ist, dass im Vergleich zum Graphen mit konstanter Qualität dessen X-Achse einen doppelt so hohen Endwert hat. Laut dieser nicht zu verallgemeinernden Grafik hat auch hier der Schrödinger-Dirac-Encoder (bis auf MS-MPEG4v3 im Bereich der niedrigen Bitrate) den niedrigsten SSIM-Wert und die Ergebnisse ähneln denen des ersten Tests. Beim Anschauen sind die charakteristischen Unterschiede der Codecs deutlich geworden, auch wenn sie am SSIM-Wert gemessen eine gleichwertige Qualität (d.h. strukturelle Ähnlichkeit zum Original) besitzen.



6.2 Interpretation der Ergebnisse

Die SSIM-Kurven von Dirac (Schrödinger-Encoder), Theora und Microsoft MPEG4v3 (ffmpeg) liegen so dicht beieinander und kreuzen sich, dass nur schwer gesagt werden kann, wer die beste Qualität bringt (wobei Dirac dennoch leicht hinter Theora liegt). Auch die Bildbeobachtung zeigte eher einen charakteristischen Unterschied des Bildes. Theora wirkt weichzeichnend und selektiv in der Bildgenauigkeit, MS-MPEG4v3 lässt viele größere Blöcke erkennen und Schrödinger ein gelegentliches Farbflimmern sowie die gerundeten Musterungen bzw. unruhigen Störungen auf manchen Flächen.

Der Dirac-Codec ist nicht so komplex wie H.264, doch die Menge der Einstellungen ist schwer überschaubar, sodass – wie bei allen anderen auch – nur die Default-Werte verwendet wurden. Theora hat in den letzten Jahren viele Verbesserungen erhalten (und wird nach der Veröffentlichung von 1.2 wahrscheinlich einen deutlich besseren SSIM-Wert erzielen, da nicht mehr auf PSNR optimiert wird), sodass die etwas bessere Qualität als Dirac verständlich ist. Es ist also davon auszugehen, dass Schrödinger (immerhin der jüngste Encoder) noch einiges an

Potential nach oben hat, welches hier nicht deutlich wird. Die hohe Optimierung von x264 gerade in Detailfragen und das Hinzuziehen von verschiedensten Verfahren (z.B. spatial adaptive quantization) scheinen mehr auszumachen als die Grundentscheidung zwischen DCT und DWT (Wavelets). Dies könnte auch der Grund sein, warum XviD und VP8 so gut abschneiden. Da die Codebasis des VP8-Encoders zur Zeit stark erneuert wird, ist in naher Zukunft damit zu rechnen, dass sich der Abstand zu x264 verkleinern wird.

6.3 Fazit

Die Geschichte von DCT ist um einiges älter als die der Wavelet-Codecs. Somit könnte Dirac – wie damals MPEG1 – Erneuerungen und Erweiterungen erfahren und ein zukünftiger Wavelet-Codec bessere Ergebnisse als aktuell H.264 erzielen. Jedoch ist JPEG2000 im Gegensatz zu JPG trotz besserer Qualität kaum verbreitet und Google bringt mit WebP ein weiteres DCT-basiertes Bildformat. In naher Zukunft wird es wohl – gerade im Internet-/Heimbereich (z.B. HTML5-Video und BluRay) durch die Einigung auf Standards – bei der hohen Verbreitung von DCT bleiben.

Dirac ist heute schon ein mit Theora (oder „besser“, auch weil flexibler) vergleichbarer Codec, der durchaus einsatzbereit ist. Gerade die Möglichkeit der verlustfreien Kompression und die Offenheit des Standards lassen auch die Anwendung zur Archivierung als sinnvoll erscheinen. Da keine älteren Codecs wie MPEG2 vertreten waren, konnte der Abstand zwischen ihnen und Dirac nicht dargestellt werden.

Wie auch PSNR-Vergleiche ist der Vergleich per SSIM oder auch VHQ nicht in der Lage, etwas über die psychovisuelle Wahrnehmung der Qualität auszusagen. Dafür müsste für die verschiedensten Eingabequellen ein Personentest erfolgen. Die Benutzung des Attributs „Qualität“ für den SSIM-Index ist eine Bequemlichkeit, die nicht die wahrnehmbare Qualität, sondern den höheren SSIM-Wert meint. Dennoch zeigte sich der SSIM im Test als guter Richtwert, was die ungefähre „Rangfolge“ der gesehenen Qualität abbildet.

7 Anmerkungen

7.1 Softwarestand

Die geplante Zeitverkürzung der SSIM-Berechnung durch Wahl einiger Frames über das Video verteilt war in gewählter Form mit dem GStreamer-Element „gnlfilesource“ nicht mit allen Codecs zuverlässig, sodass die Frameansteuerung versagte und ein falscher SSIM-Wert berechnet wurde. Deshalb werden die Videos zur Berechnung immer noch komplett in PNGs umgewandelt.

Um die Testumgebung nicht nur schnell, sondern auch stabil und flexibel zu gestalten sind noch folgende Änderungen nötig. Der aktuelle Zustand ergibt keine Veröffentlichung, die von weitem Nutzen wäre.

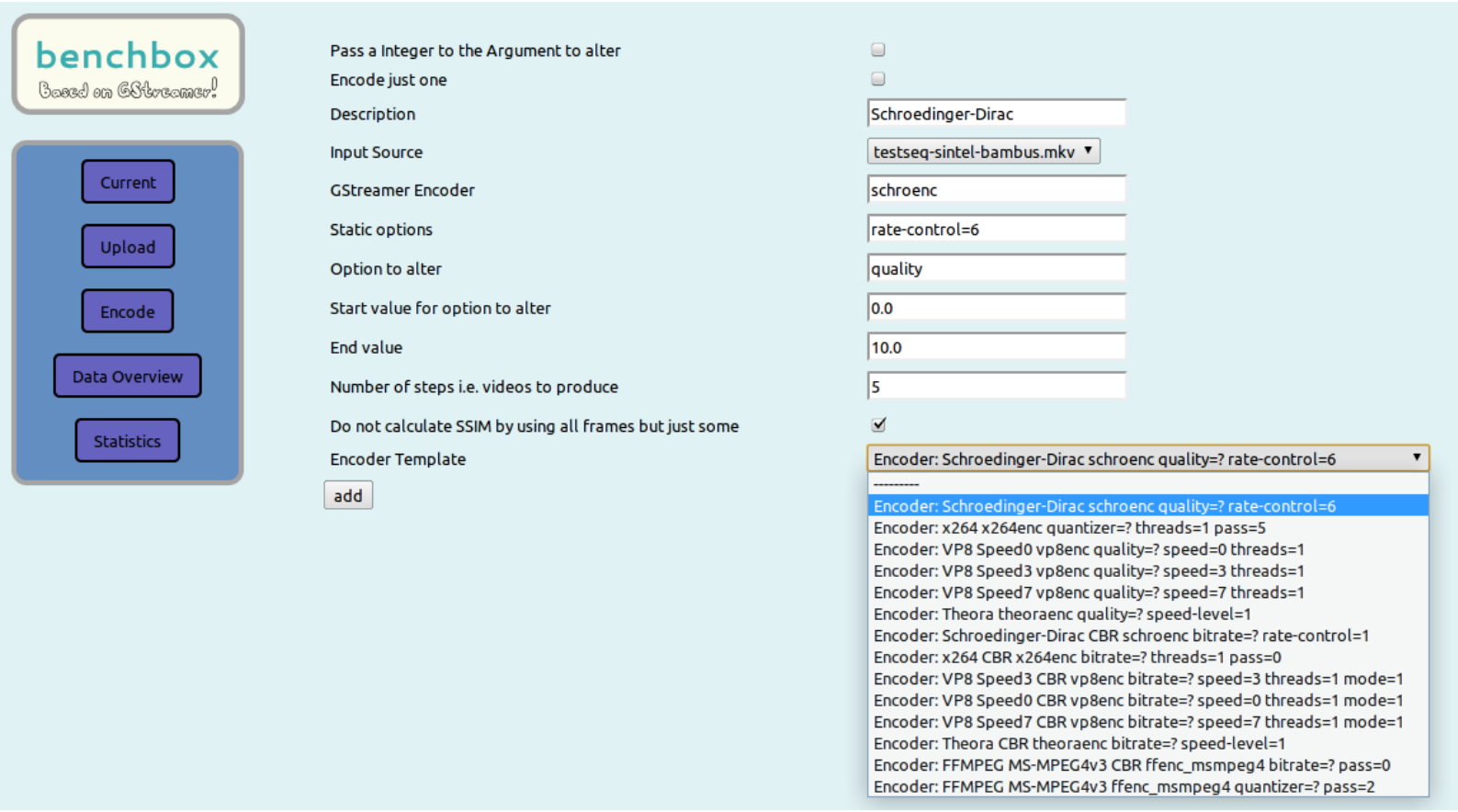
- Auslagerung der GStreamer-Mainloop in einen per DBus ansprechbaren Prozess und integriertere Verwendung von GStreamer, um das Erzeugen von temporären PNG-Dateien zu vermeiden, wenn das Differenzbild, das Original oder die kodierte Version eines bestimmten Frames angezeigt werden soll oder der SSIM berechnet wird.
- Dazu wäre auch die Entwicklung eines SSIM-GStreamer-Plugins hilfreich.
- FFmpeg sowie MEncoder (MPlayer) und andere Direktaufrufe einbinden.
- Erweiterte Job-Verwaltung (z.B. Löschen), besseres Interface, Dokumentation/Hilfe, Benutzerauthentifizierung und generell mehr Einstellungsmöglichkeiten
- Der Quellcode würde unter der GNU AGPL veröffentlicht.

7.2 Beispielhafter Ablauf eines Tests

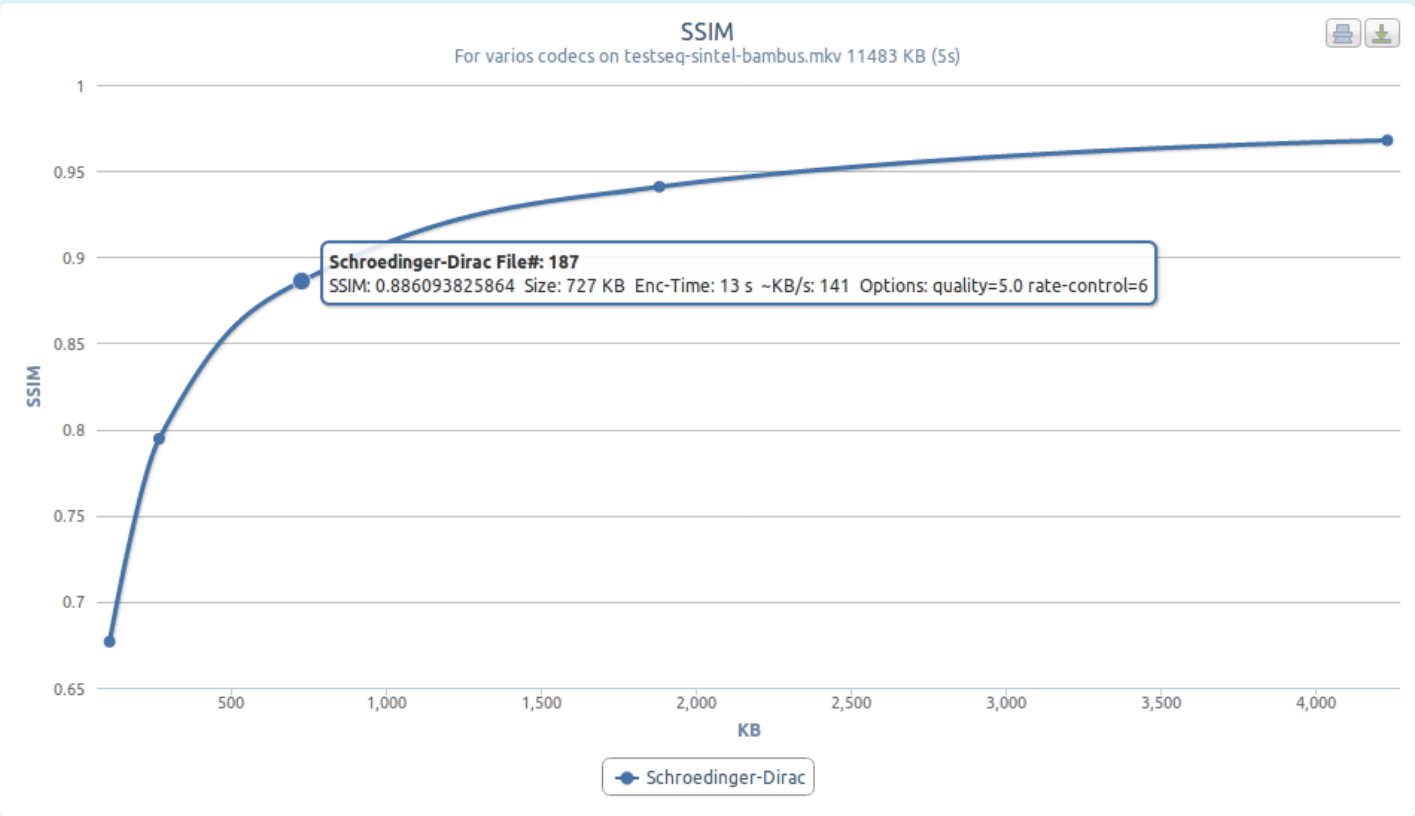
Es ist zu empfehlen, eine verlustfreie Eingabequelle ohne Audiospur zu benutzen. Diese kann durch skalieren eines hochauflösenden Videos erzeugt werden, das per Schrödinger-Dirac im lossless-Modus in einen Matroska-Container gespeichert wird. Der passende GStreamer-Aufruf vom Terminal aus wäre folgender:

```
gst-launch-0.10 filesrc location="$1" ! decodebin2 name=demux \
      { matroskamux name=mux ! filesink location="$2" } \
      { demux. ! queue ! colorspace ! videoscale ! \
      video/x-raw-yuv, width=640, height=360 ! schroenc rate-control=3 ! mux. }
```

Anschließend wird das Video hochgeladen und nun wird das Menü „Encode“ aufgerufen:



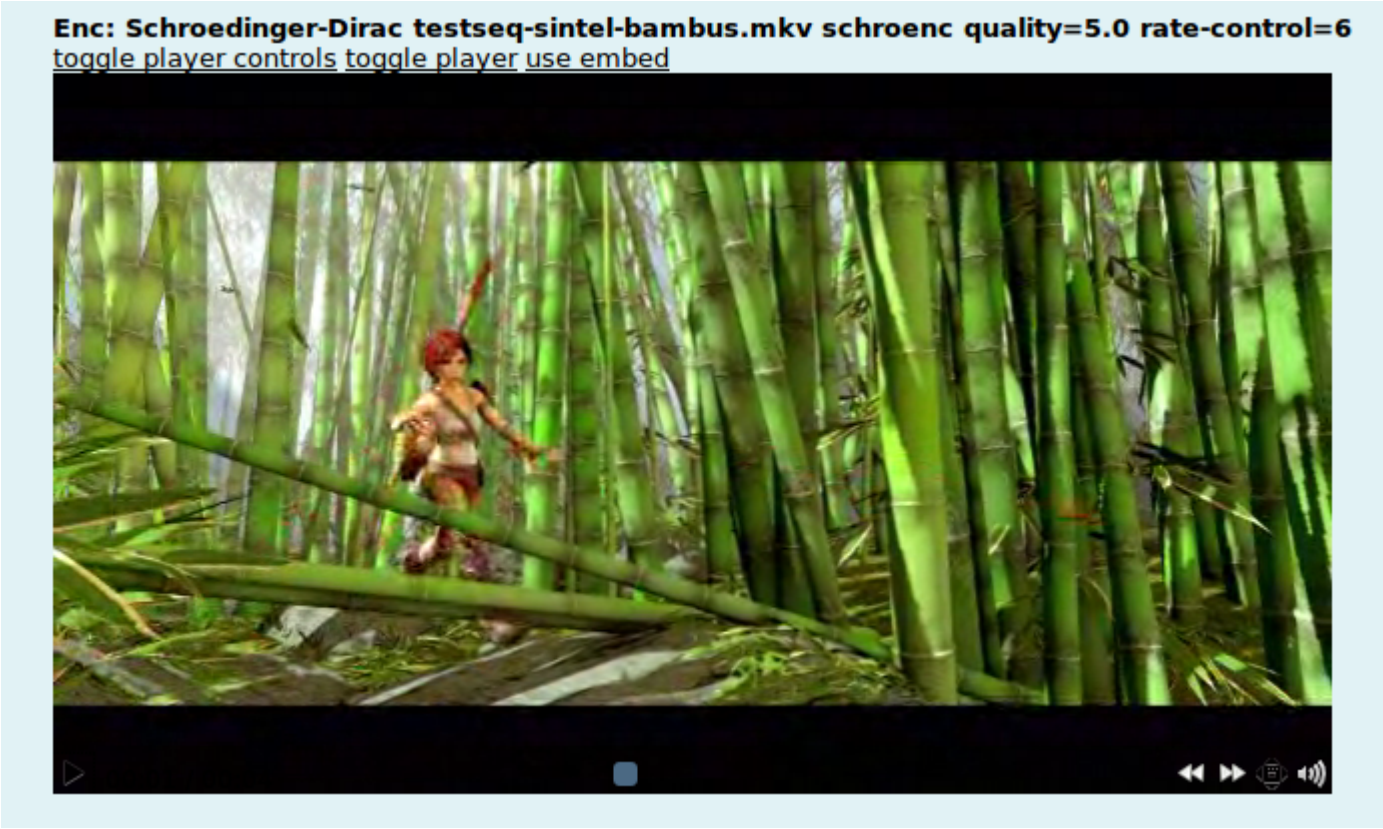
Die Eingabesequenz soll in fünf Schrödinger-Videos umgewandelt werden, deren konstante Qualität (*rate-control=6*) Werte von 0.0 bis 10.0 annehmen soll (also gestaffelt: 0.0, 2.5, 5.0, 7.5, 10.0 für *quality*). Die Berechnung des SSIM-Wertes soll schnell durchgeführt werden (bei Dirac funktioniert es verlässlich). Es dauert nicht lang, dann sind 100% erreicht und der Status ist „Finnished“ und es kann in „Statistics“ folgendes für die Eingabedatei betrachtet werden:



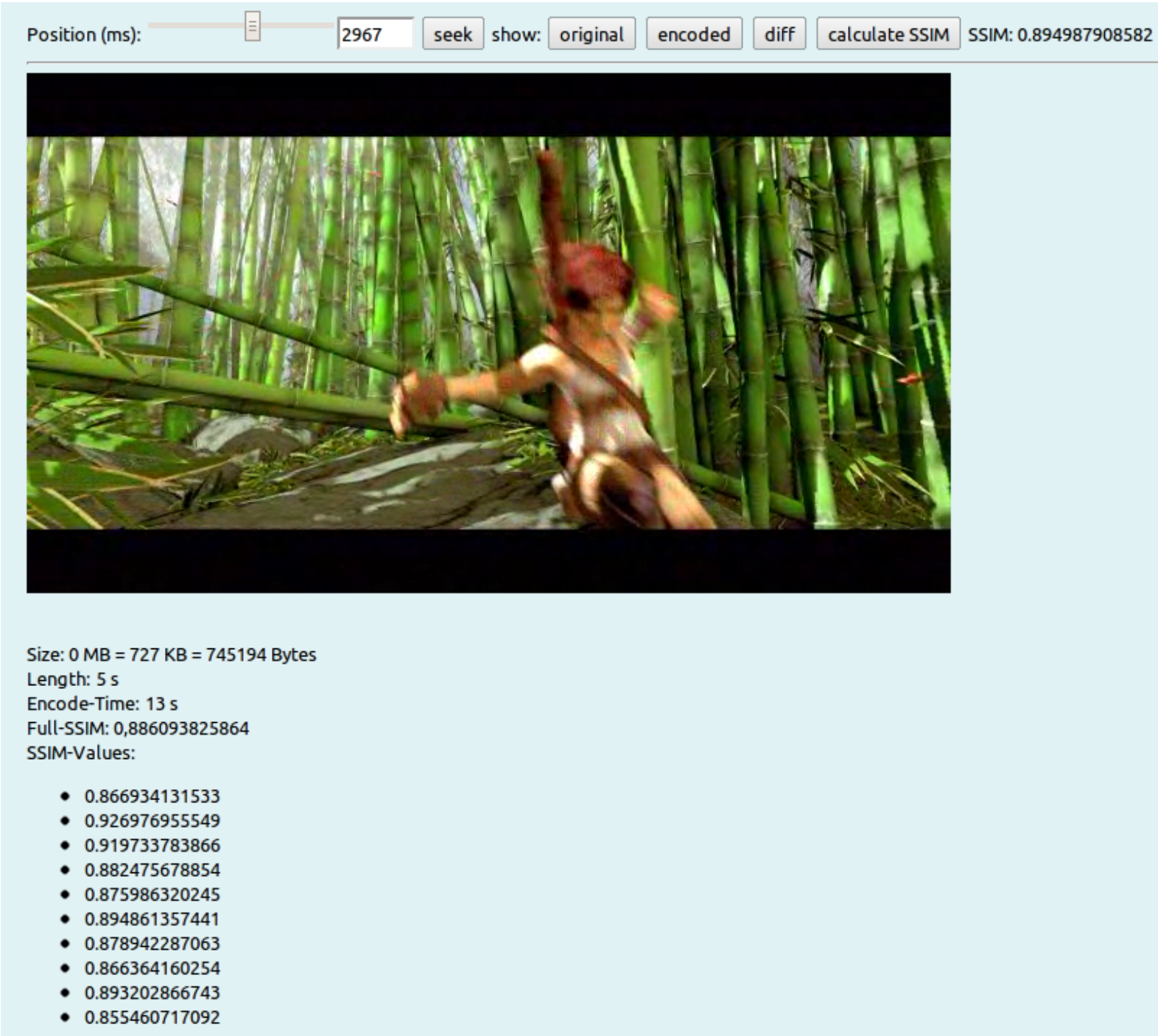
Play last hovered File#: 187

Chart powered by Highcharts JS under CC-BY-NC license

Wobei zu diesem Zeitpunkt für andere Codecs noch keine Kurven erstellt wurden und Schrödinger-Dirac also alleine ist. Die Details für die Datenpunkte sind per Mousehover sichtbar. Nun wird die Datei #187 genauer betrachtet:



Der Browser Epiphany benutzt GStreamer für HTML5-Videoelemente und spielt somit das Dirac-Video direkt ab. Ansonsten müsste „use embed“ gedrückt werden, damit das Video per <embed>-Tag durch ein Browser-Mulitmedia-Plugin abgespielt wird. Etwas weiter unten ist ein Ansteuern der einzelnen Frames möglich:

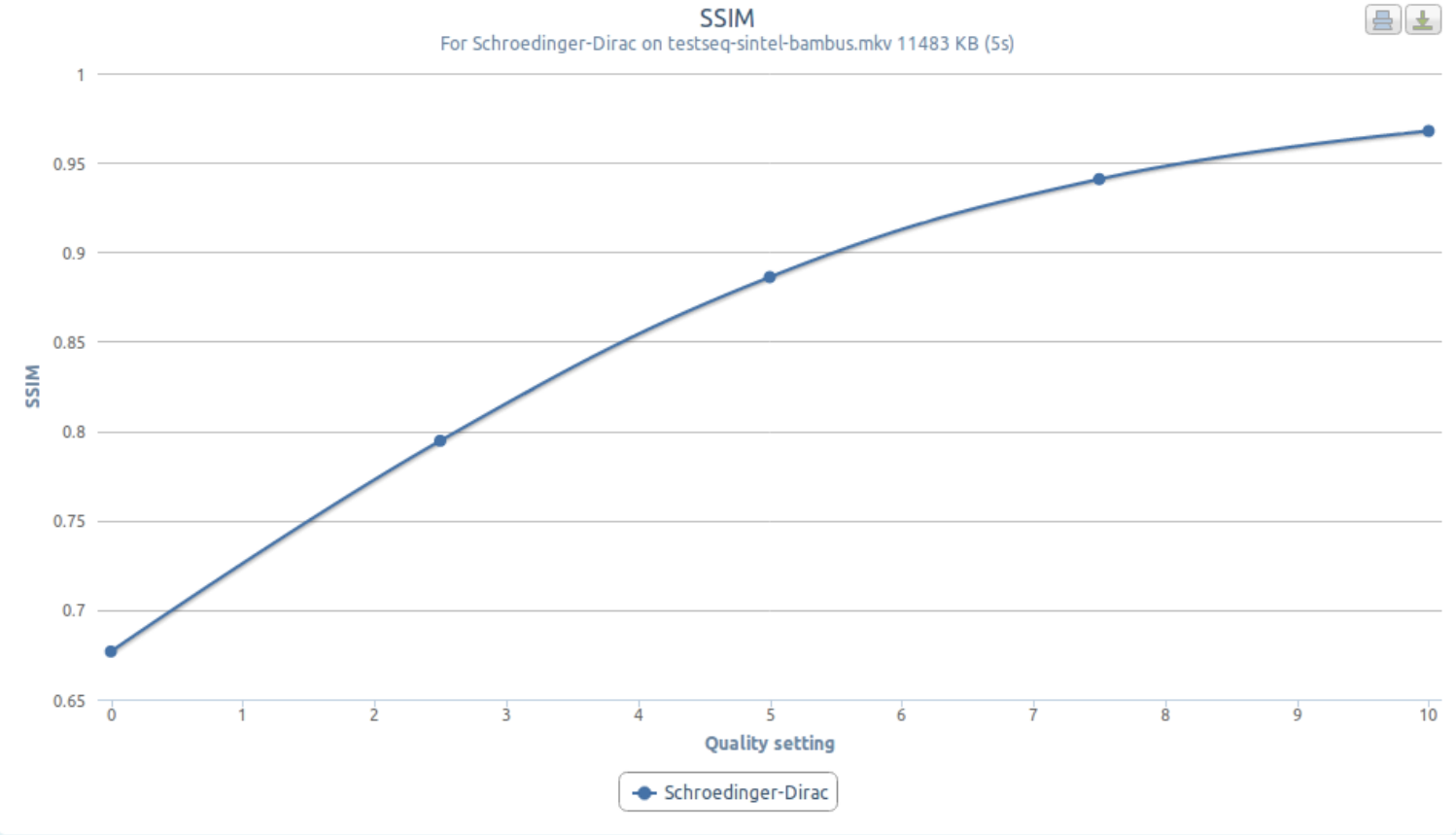


Die Position wird in Millisekunden angegeben, es kann das Originalbild, das komprimierte Bild (hier) oder auch die Differenz angezeigt werden. Der SSIM für das Bild ist also ca. 0,894987908582. Nach den Details sehen wir auch die Liste der SSIM-Werte, welche in die Berechnung des Durchschnitts eingingen. Es folgt ein Blick auf „Data Overview“:

Schroedinger-Dirac: testseq-sintel-bambus.mkv (5 sec)
Status: 100,0% Finished
SSIM-Graph

- quality=0.0 rate-control=6 | 13 sec | 107 KB | SSIM: 0,676734182157
- quality=2.5 rate-control=6 | 11 sec | 267 KB | SSIM: 0,794515397738
- quality=5.0 rate-control=6 | 13 sec | 727 KB | SSIM: 0,886093825864
- quality=7.5 rate-control=6 | 13 sec | 1882 KB | SSIM: 0,940869709906
- quality=10.0 rate-control=6 | 12 sec | 4232 KB | SSIM: 0,96773680108

Die Dateiliste (Parameter, rechts) würde wieder zu den vorherigen zwei Screenshots führen. Ein Klick auf „SSIM-Graph“ liefert den SSIM-Wert in Abhängigkeit von der Qualitätseinstellung:



Play last hovered File#: 186