Erfahrungen mit visuellen Datenformaten

Arbeitspaket 4

Sönke Brummerloh
OFFIS e.V.
Escherweg 2
26121 Oldenburg

Telefon: +49 441 9722 – 189
Telefax: +49 441 9722 – 102
E-Mail: soenke.brummerloh@offis.de

1 Einleitung

In Arbeitspaket 4 (Effiziente Interaktion mit großen Datenobjekten mit Hilfe von Quicklooks) wurden animierte GIF-Bilder, Videos und Super Overlays eingesetzt,

um das Transfervolumen von speicherintensiven Grafiken zu reduzieren. Die Ausgangsgrafiken lagen dabei in allen Fällen als JPEG-Dateien vor. Die gesammelten Erfahrungen werden im Folgenden zusammengefasst.

2 Videos

Das Ziel von Video-Codecs ist es, die Einzelbilder eines Videos in einer möglichst guten Bildqualität zu speichern und dabei den Speicherbedarf des fertigen Videos möglichst gering zu halten. Das macht moderne Video-Codecs zu einem attraktiven Werkzeug, um Bilderserien effizient zu komprimieren.

Zur Abschätzung der Leistungsfähigkeit wurden vier moderne weitverbreitete Codecs evaluiert. Dies waren zum einen die drei MPEG-4 kompatiblen Codecs FFmpeg MPEG-4, Xvid[1] und x264[2]. Zum anderen wurde zum Erzeugen von Flash-Videos der Sorenson H.263 kompatible Codec der FFmpeg-Bibliothek[3] genutzt. Die Wahl fiel auf diese Codecs, da die mit ihnen erzeugten Videos auf den meisten modernen Computern ohne die Installation eines neuen Codecs abgespielt werden können. Open-Source-Codecs wie Dirac[4] oder Theora[5] wurden nicht getestet, weil zum Abspielen zugehöriger Videos die Codecs in der Regel installiert werden müssten.

Um zu überprüfen, wie gut die vier Codecs Bilderserien codieren können, wurde eine beim DLR-DFD typische Bilderserie mit allen vier Codecs mit unterschiedlichen Parametern codiert. Von besonderem Interesse waren die Codierungsdauer sowie die Bildqualität im Verhältnis zur Dateigröße. Die Bildqualität wurde als „Peak Signal to Noise Ratio“ (PSNR) gemessen.

Als Codierungswerkzeug wurde das Kommandozeilen-basierte Werkzeug *mencoder* eingesetzt, das Teil der MPlayer[6]-Installation ist. Die MPlayer-Software steht unter der GNU General Public License[7] (GPL) und ist auf vielen Betriebssystemen lauffähig. Auf Grund seiner vielen Konfigurationsmöglichkeiten und der relativ einfachen Handhabung wurde mencoder den beiden anderen unter Linux verbreiteten Codierungswerkzeugen FFmpeg[8] und Transcode[9] vorgezogen.

Die folgenden Ergebnisse gelten streng genommen nur für das verwendete Bildmaterial und die getesteten Software Versionen. Eine generelle Tendenz für die Codecs bzgl. der untersuchten Merkmale sollte sich aber verallgemeinern lassen.

2.1 Ergebnisse

Bei den Tests hat sich ergeben, dass der H.263-Codec und der FFmpeg MPEG-4-Codec am Performantesten sind. Beide benötigen nur etwa ein Viertel der Laufzeit,

[1] http://www.xvid.org
[2] http://www.videolan.org/developers/x264.html
[3] http://ffmpeg.mplayerhq.hu/
[4] http://dirac.sourceforge.net
[5] http://www.theora.org
[6] http://www.mplayerhq.hu
[7] http://www.gnu.org/licenses/gpl.html
[8] http://ffmpeg.mplayerhq.hu
[9] http://www.transcoding.org/cgi-bin/transcode

die der x264-Codec benötigt. Der Xvid-Codec benötigt etwa 70% der Laufzeit vom x264-Codec. Der Platzbedarf bei einem PSNR von 30 dB verhält sich dagegen umgekehrt: Der x264-Codec erzeugt die kleinsten Video-Dateien. Gegenüber dem x264-Codec erzeugt Xvid um ca. 10%, FFmpeg MPEG-4 um ca. 15% und der H.263-Codec um ca. 20% größere Video-Dateien.

Als der leistungsfähigste Codec hat sich x264 herausgestellt. Er produziert bei niedrigen Bitraten die besten Ergebnisse. Bei höheren Bitraten produzierte bei den verwendeten Ausgangsbildern allerdings Xvid bei vergleichbarer Qualität die kleineren Video-Dateien. Bei dem verwendeten Bildmaterial waren solche hohen Bitraten nicht notwendig, um von der Bildqualität her sehr gute Ergebnisse zu erhalten.

Mit der Grundeinstellung ohne zusätzliche Parameter hat ein Codec im Vergleich zu Einstellungen mit weiteren Parametern eine sehr gute Laufzeit, erzeugt jedoch auch die größte Video-Datei. Die Größe der Video-Datei kann beim gleichen PSNR zum einen durch optimierende Parameter reduziert werden und zum anderen durch das Verwenden des twopass-Modus, bei dem das Ausgangsmaterial erst vollständig analysiert wird, bevor das Video generiert wird.

Wenn keine exakte Dateigröße benötigt wird, sind leicht optimierende Einstellungen zu empfehlen. Mit geeigneten Parametern ist es möglich, die Dateigröße bei gleicher Qualität zu reduzieren, ohne die Laufzeit deutlich zu erhöhen. Starke Optimierungen sind nur zu empfehlen, wenn die Laufzeit keine oder nur eine geringe Relevanz hat, da sich die Laufzeit durch einige Parameter deutlich erhöht. Eine Abschätzung der Auswirkungen von einigen Parametern auf die PSNR ist im MPlayer-Handbuch[10] zu finden. Falls eine exakte Dateigröße benötigt wird, kann die twopass-Codierung genutzt werden. Dadurch wird die Laufzeit allerdings in etwa verdoppelt. Stark optimierende Einstellungen bewirken beim twopass-Modus gegenüber leicht optimierenden Einstellungen kaum eine Verbesserung. Eine qualitätsbasierte Codierung bringt bzgl. Speicherplatz und Laufzeit ähnliche Ergebnisse wie die Codierung mit konstanter Bitrate ohne optimierende Einstellungen und ist daher nicht zu empfehlen.

Wenn die Video-Dateien möglichst wenig Speicherplatz verbrauchen sollen, hat sich eine Reduktion der Auflösung als brauchbare Lösung herausgestellt: Die Laufzeit der Codierung sinkt dadurch erheblich und die erzeugten Video-Dateien benötigen deutlich weniger Speicherplatz. Der Nachteil ist, dass feine Details durch die geringere Auflösung wegfallen. Wenn dies akzeptabel ist, stellt eine Verringerung der Auflösung eine optimale Möglichkeit dar, um die Laufzeit zu verringern und gleichzeitig die Video-Dateigröße zu reduzieren.

Bei der eingesetzten Version (Mplayer-1.0rc2) von mencoder kam es zu einer leichten Farbverfälschung. Diese resultiert aus der automatischen Umwandlung der JPEG-Bilder in ein Motion-JPEG-Zwischenformat durch mencoder bzw. FFmpeg (dessen Bibliothek von mencoder genutzt wird).

Nach einer Begutachtung der Videoqualität durch menschliche Betrachter stellte sich heraus, dass nur x264 ab einer gewissen Bitrate feine Details und Farbschattierungen nicht mehr weichzeichnet. Bei den anderen Codecs trat ein

[10] http://www.mplayerhq.hu/DOCS/HTML/en/encoding-guide.html

Weichzeichnen selbst bei sehr hohen Bitraten auf. Das Weichzeichnen kann durch die Wahl geeigneter Quantizer reduziert werden. Als eine geeignete Einstellung wurde der MPEG-Quantizer identifiziert. Bei den Codecs FFmpeg-MPEG4 und Xvid kann dieser von der Standardeinstellung „H.263“ auf „MPEG“ geändert werden, um ein besseres Ergebnis zu erhalten. Die Laufzeit ändert sich durch diese Einstellung kaum. Der x264-Codec arbeitet bereits mit entsprechend guten Quantizer-Einstellungen, während beim H.263-Codec kein MPEG-Quantizer gewählt werden kann. Zur feineren Abstimmung können manuell eine Reihe weiterer Quantizer-Parameter spezifiziert werden.

2.2 Fazit

Videos als Quicklooks sind dann eine gute Alternative zu Einzelbildern, wenn kleinere Fehler durch Farbverfälschungen und Weichzeichnen kein Problem darstellen. Andernfalls ist von Videos als Darstellungsform abzuraten.

Grundsätzlich müssen bei der Auswahl eines geeigneten Codecs mehrere Kriterien berücksichtigt werden. Es hat sich herausgestellt, dass die Codecs bei gleicher Video-Qualität eine unterschiedliche Laufzeit haben und unterschiedlich große Video-Dateien generieren. Außerdem muss zum Abspielen bei manchen Formaten mit größerer Wahrscheinlichkeit ein Codec nachinstalliert werden als bei anderen Formaten.

Auf Grund der weiten Verbreitung von Flash-Plugins kann davon ausgegangen werden, dass Flash-Videos am ehesten abgespielt werden können. Video-Dateien, die mit einem zum MPEG-4 Part 2 Standard kompatiblen Codec codiert sind, sollten ebenfalls auf einer Vielzahl von Systemen abspielbar sein. Das in den AVI-Dateien angegebene Videoformat sollte bei der Verwendung der MPEG-4 Part 2 kompatiblen Codecs allerdings auf „DX50“ gesetzt werden. Zum Abspielen wird dann der kompatible weitverbreitete Codec DivX 5 genutzt. Beim relativ neuen x264-Codec kann momentan davon ausgegangen werden, dass er auf vielen Systemen noch installiert werden muss.

FFmpeg MPEG-4 und H.263 erzeugen bei vergleichbarer Qualität durchweg größere Dateien als x264 und Xvid. Dafür benötigen FFmpeg MPEG-4 und H.263 bei Codierungen ohne optimierende Einstellungen nur ca. 25% der x264-Codec-Laufzeit. Je nachdem, ob Speicherplatz oder Codierungszeit der begrenzende Faktor ist, sollte daher zwischen x264 und Xvid oder FFmpeg MPEG-4 und H.263 gewählt werden.

Falls viel gleichartiges Bildmaterial in Videos umgewandelt werden soll, ist es empfehlenswert Videos testweise zu erstellen, um zu ermitteln wie gut die Codecs das Ausgangsmaterial codieren können und mit welchen Einstellungen der PSNR bzw. die Bildqualität ausreichend ist.

Da die Codecs weiterentwickelt und optimiert werden, ist es sinnvoll die getesteten Codecs in regelmäßigen Abständen erneut Tests zu unterziehen. Wie Tests des Computermagazins c't gezeigt haben (c't 10/2003, S. 146ff und c't 10/2005, S. 146ff) können mit neueren bzw. optimierten Codecs bessere Ergebnisse erwartet werden.

3 Animierte GIF-Bilder

Eine alternative Möglichkeit Bilderserien zu bündeln sind animierte GIF-Dateien. Diese haben sich gegenüber den getesteten Videoformaten nur in Ausnahmefällen als praktikabel erwiesen. Dies liegt zum einen daran, dass das Erzeugen von GIF-Dateien zu langen Bilderserien mit z. B. ImageMagick[11] sehr viel Arbeitsspeicher benötigt. Zum anderen sind GIF-Bilder in ihrer Darstellung auf 256 Farben beschränkt. Viele Bilddateien verwenden mehr als 256 Farben. Durch eine Reduktion der Farbpalette würde es daher zu einer schlechteren Bildqualität kommen. Außerdem führt die Farbreduktion zu Bildern, die sich schlecht komprimieren lassen. Das führt dazu, dass die animierten GIF-Bilder größer werden als Videos in besserer Qualität. Es kann sogar dazu kommen, dass die GIF-Bilder mehr Speicherplatz benötigen als die ursprünglichen JPEG-Bilder mit mehr Farben.

Falls die Ausgangsbilder allerdings mit 256 Farben auskommen, werden GIF-Bilder eine attraktive Alternative, die gegenüber z. B. JPEG-Bildern nur 15 bis 20% des Speicherplatzes benötigt. Da die GIF-Bilder in einem solchen Fall gegenüber komprimierten Videos in Originalqualität vorliegen, sind sie hier ausnahmsweise die bessere Wahl. Ein Gewinn an Speicherplatz durch ein animiertes GIF-Bild gegenüber einzelnen GIF-Bildern ist vorhanden aber gering.

4 Super Overlays

Als eine mögliche neue Darstellungsform von Quicklooks werden im Folgenden Super-Overlays in Google Earth vorgestellt. Google Earth ist eine von Google für Linux, MacOS X und Microsoft Windows frei verfügbare Anwendung zum Betrachten der Erdoberfläche. In Google Earth wird zunächst die Erde in einer Weltraumansicht dargestellt. In dieser Ansicht lässt sich der Globus frei drehen und zoomen. Dabei ist es möglich, nahezu stufenlos bis auf eine Auflösung von wenigen Metern zu zoomen.

Die Karte von Google Earth kann mit Grafiken überlagert werden. Dazu werden XML-basierte KML-Dateien[12] in Google Earth geladen. Zusätzliche Grafiken, die über die Erdoberfläche gelegt werden, werden als *Ground Overlays* bzw. allgemein als *Overlays* bezeichnet. In einer KML-Datei können mehrere *Region*-Elemente mit Overlays definiert werden. Ein Region-Element umfasst eine Fläche auf der Google Earth Karte. Für jede Region berechnet Google Earth die Fläche, die diese in der aktuellen Ansicht einnehmen würde. Dabei kann für jede Region festgelegt werden, ab wie vielen minimal geladenen Bildpunkten sie aktiviert bzw. dargestellt wird und ab wie viel maximal geladenen Bildpunkten die Region wieder deaktiviert bzw. ausgeblendet wird. So ist es möglich, bestimmte Grafiken erst ab einer bestimmten Zoom-Stufe anzuzeigen und sie ab einer bestimmten Zoom-Stufe auch wieder auszublenden. Auf diese Weise kann in einer Region auf weitere KML-Dateien verwiesen werden, mit dem Ergebnis, dass diese erst beim Aktivieren der Region nachgeladen werden. So ist es z. B. möglich bei geringer Zoom-Stufe eine grob aufgelöste Grafik darzustellen. Sobald der Anwender weiter

[11] http://www.imagemagick.org
[12] http://code.google.com/apis/kml/documentation/

hineinzoomt, werden weitere KML-Dateien mit detaillierteren Grafiken geladen[13]. Dadurch können Hierarchien von Region-Elementen angelegt werden, um schrittweise immer detailliertere Ausschnitte einer großen Grafik zu laden. Eine solche Hierarchie wird als *Super-Overlay* bezeichnet.

Der Vorteil solcher Super-Overlays ist, dass nur geladen wird, was der Nutzer sich genauer anschauen möchte. Dadurch sind die Ladezeiten für den Nutzer und das generelle Datenaufkommen geringer. Da eine Grafik effektiv in mehreren Auflösungen auf dem Server hinterlegt ist, wird auf diesem mehr Speicherplatz benötigt.

Für alle Bilder, die in Super-Overlays umgewandelt werden sollen, muss deren genaue Lage auf der Erde bekannt sein. Falls die Lage nicht bekannt ist, muss sie zunächst manuell in Google Earth ermittelt werden. Satellitenaufnahmen haben darüber hinaus das Problem, dass sie in der Regel in einer Projektion vorliegen, die nicht der in Google Earth verwendeten WGS84 entspricht. Satellitenaufnahmen müssen daher zunächst in eine verwendbare Projektion konvertiert werden. Dies hat gegebenenfalls manuell zu geschehen.

Durch diesen hohen manuellen Aufwand ist der Einsatz von Super-Overlays nur praktikabel, wenn es sich um regelmäßig erstellte Bilder handelt, deren Lage und Projektion nur einmal ermittelt werden muss und die sich über die Zeit nicht ändert. Nachdem die notwendige Projektion ermittelt wurde, kann die Konvertierung der Projektion und das Anlegen der Super-Overlays in Zukunft automatisch erfolgen. Zur Automatisierung der Super-Overlay-Erstellung wurde eine eigene Open-Source Anwendung[14] implementiert, die über die Kommandozeile oder Skripte angesteuert werden kann. Diese ist auch in der Lage Bilderserien mit Zeitbezügen in ein Super-Overlay umzuwandeln.

Zusammenfassung:

- Der visuelle Eindruck der Darstellung in Google Earth ist sehr gut.
- Die zu ladende Datenmenge lässt sich über Super Overlays gut einschränken, da nur geladen wird, was sich der Nutzer genauer anschaut.
- Die Erzeugung von Super Overlays kann allerdings recht arbeitsintensiv sein, wenn mangels geeigneter Metadaten für jedes Super Overlay die exakten Koordinaten der Grafik in Google Earth interaktiv ermittelt werden müssen.
- Alle Satellitenbilddaten müssen vor der Darstellung in Google Earth in die Standardprojektion von Google Earth (WGS84) umgerechnet werden.

5 Fazit

Alle Komprimierungsansätze haben ihre Berechtigung: Bilderserien mit über 256 Farben lassen sich gut über Videos komprimieren. Für Bilderserien mit bis zu 256 Farben können auch gut animierte GIF-Dateien verwendet werden. Wenn Grafiken mit hoher Ausdehnung visualisiert werden sollen, deren genaue Position bekannt ist und diese in einer für Google Earth geeigneten Projektion vorliegen, können sie gut als Super Overlays dargestellt werden.

[13] http://code.google.com/apis/kml/documentation/kml_21tutorial.html#workingregions
[14] http://sourceforge.net/projects/superoverlaygen/

Zu Beachten ist, dass jeder dieser Ansätze rechenintensiv ist. Besonders wenn viele Bilderserien prozessiert werden sollen, sollte ausreichend Rechenkapazität und Zeit zur Verfügung stehen.