

Teilgebiet

Die Performance von Websites hängt von vielen Faktoren ab. Für die Aufgabenstellung von Task 3 wurde die Kompression von Bildern als Teilgebiet gewählt. Die Dateigrößen von statischen Inhalten spielen eine Schlüsselrolle bei der Auslieferungsgeschwindigkeit. Auch wenn die Bandbreiten der Internetnutzer immer weiter ansteigen benötigen größere Bilder länger zum Herunterladen.

Testset

Zur Generierung eines geeigneten Testsets wurde ein Konsolenprogramm erstellt. Das Programm holt sich die aktuelle Alexa Topliste¹ mit 1.000.000-Einträgen, aus dieser Liste wird eine konfigurierbare Anzahl an Websites extrahiert (Standardwert ist 10). Das Programm besucht dann hintereinander die Homepage aller Websites und lädt alle Bilder die mit einem ``-Element ausgezeichnet sind herunter. Daraufhin werden alle Stylesheets die mit einem `<link rel="stylesheet">`-Element referenziert werden heruntergeladen um alle Bilder aus diesen herunterzuladen.

Bildoptimierung

Nachdem alle Bilder heruntergeladen wurden werden alle Bilder, aller Websites der Reihe nach mit verschiedenen Programmen optimiert. Der Ablauf ist dabei wie folgt:

1. Animierte GIF-Bilder werden nicht optimiert und ohne weitere Bearbeitung zu den optimierten Bildern hinzugefügt. Die Browserunterstützung für Animated Portable Network Graphics (APNG) ist sehr dürrig², alternative Formate gibt es nicht. Google hat die Unterstützung von Animationen im WebP-Format vor kurzem aufgenommen. Es gibt jedoch nur experimentelle Builds von Chromium die dieses neue Format unterstützen. Der Autor hat dies ausprobiert, jedoch scheiterte die Konvertierung an der Inkompatibilität von *libgif* und *gif2webp*.

¹ <http://s3.amazonaws.com/alexa-static/top-1m.csv.zip> – Die Alexa Topliste mit den 1.000.000 meist besuchten Websites weltweit (wird täglich aktualisiert).

² <http://en.wikipedia.org/wiki/APNG> – Bei Wikipedia gibt es eine Auflistung von Programmen die APNG unterstützen.

2. (Nicht animierte) GIF-Bilder wurden mit Hilfe von Imagemagick³ (*php5-imagick*) in PNG-Bilder umgewandelt und dieses wurde zur weiteren Bearbeitung herangezogen.⁴
3. Alle Arten von Bilder wurden dann mit imgmin⁵ optimiert. Imgmin ist eine Toolsammlung und hat genau das gleiche Ziel wie diese Arbeit, das Finden der kleinsten Dateigröße ohne dabei Qualität zu verlieren. Es war also nur ein logischer Schritt diese Toolsammlung in den Prozess mit einzubauen.
4. PNG-Bilder werden nun nochmals mit improved pngquant⁶ optimiert. Das Ergebnis wird mit dem von imgmin verglichen und die kleinere Datei wird behalten.
5. Im letzten Schritt werden die optimierten Bilder ins neue WebP-Format von Google umgewandelt. Handelt es sich um eine PNG-Datei wird angenommen, dass die Datei alphantransparente Pixel enthält, so ergeben sich zwei verschiedene Kommandoaufrufe.

Statistik

Im letzten Schritt wird eine Statistik über die optimierten Bilder erstellt. Diese kann im Moment unter <http://alpha.movlib.org/research/stats.html> eingesehen werden (die URL wird sich in absehbarer Zeit ändern, die aktuelle URL wird immer im GitHub-Repository des Autors unter <https://github.com/Fleshgrinder> zu finden sein).

Fazit

Die Ergebnisse sind bemerkenswert. So konnte der Autor Bilder finden (z. B. bei Amazon.com) die nach der Optimierung größer waren als vor der Optimierung. Die Regel war jedoch ganz anders. So gibt es selbst in der Top-10 nur ein Unternehmen, das die Bilder bis zum Maximum optimiert hat und das ist Google auf der Startseite der hauseigenen Suchmaschine Google.com (z. B. bei YouTube.com was auch zu Google zählt ist dies nicht der

³ <http://www.imagemagick.org/> – Offizielle Website

⁴ <http://www.w3.org/QA/Tips/png-gif> – W3C-Artikel zum Thema „PNG versus GIF“.

⁵ <https://github.com/rflynn/imgmin> – Offizielles GitHub-Repository von imgmin.

⁶ <https://github.com/pornel/improved-pngquant> – Offizielles GitHub-Repository von improved pngquant.

Fall). Bemerkenswertestes Negativbeispiel ist Pinterest.com. Auf der Startseite findet sich nur ein Bild (PNG-Format mit unnötiger Alphatransparenz) mit 592,93 KB! Die Größe des Bildes konnte durch automatisierte Optimierung auf 161,57 KB verringert werden. Beim Besuch der Homepage von Pinterest.com mit leerem Cache konnte der Autor beobachten wie es das Bild langsam zeichnete (mit langsam sind hier Millisekunden gemeint, der Autor besitzt eine 50 MB/s-Kabelleitung).

Mögliche Erweiterungen

Das Programm kann um viele weitere interessante Aspekte erweitert werden. So z. B. könnten Bilder auch aus `<iframe>`-Elementen (z. B. Microsoft Live.com), inline Stylesheets, JavaScript und AJAX-Aufrufen extrahiert werden. Animierte GIF-Grafiken in APNG oder animierte Sprites umwandeln. Bei der Umwandlung von PNG in WebP überprüfen ob eine Alphatransparenz im Quellbild vorhanden ist.

Eine weitere, wirklich sinnvolle, Erweiterung wäre die Berechnung der strukturellen Ähnlichkeit (Structural Similarity, SSIM) oder erweiterte Formen des Algorithmus (MS-SSIM, MS-SSIM*). Der Autor konnte leider keine Software finden, die für alle Bildformate den Index berechnen kann bzw. die ohne (für den Autor) immensen Aufwand lauffähig gewesen wäre. Die neueste WebP-Implementierung hat die SSIM-Berechnung bereits integriert. Anhand des SSIM-Indexes könnte eine Statistik über möglichen Qualitätsverlust erstellt werden. Des Weiteren würde sie es ermöglichen, dass Bilder in voller Qualität dem Programm übergeben werden und anhand des SSIM wird die kleinstmögliche Datei ermittelt.