

Aufgabenstellung

Evaluation des „Structural Similarity“-Algorithmus¹ (SSIM) zur Bestimmung der Bildqualität für bessere Kompressionsergebnisse. Auswertung auf die Gegebenheiten der verschiedenen Browser ausrichten.

Ausgangssituation

In der Regel wird zur Bestimmung der Bildqualität beim Abspeichern ein Qualitätsregler betätigt. Bei JPEG²-Programmen handelt es sich hierbei fast immer um einen Schieberegler der von 0 bis 100 % verschoben werden kann. Bei PNG³ hingegen wird der Kompressionslevel zwischen 0 und 9 bestimmt. PNG verwendet zlib⁴ zur Kompression, daher diese Werte, wobei 0 für keine Kompression steht und 9 für die Beste. Diese Werte sind in der Regel wenig aussagekräftig was die tatsächliche Qualität des komprimierten Bildes anbelangt.

Wichtige Randnotiz zu PNG an dieser Stelle. PNG ist ein *lossless*-Format. Das bedeutet, dass die bloße Kompression die Qualität des Bildes nicht mindert. Stattdessen müssen Programme einfach das Bild erst dekomprimieren um es öffnen bzw. darstellen zu können. Kompression wird bei PNG ausschließlich durch entfernen von unnützen Metadaten und intelligentem zusammenlegen von Farb- bzw. Pixelinformationen (*Filtering*) erreicht.

Einige Optimierungsschritte funktionieren jedoch unabhängig vom Bildformat. Dies umfasst das bereits angesprochene entfernen von Metadaten, im Besonderen sind damit textuelle Daten gemeint, wie z. B. Exchangeable image file format⁵ (Exif), Adobes Extensible Metadata Platform⁶ (XMP) oder Dateibewertungen (Microsoft Windows).

Ein weiterer Schritt ist die Farbraumreduktion. Bei einem Bild das nur Grauwerte enthält wird weiß mit dem

dezimalen Wert 255 (8 Bit) repräsentiert, bei vollem sRGB-Spektrum⁷ würde der Wert hier dezimal hingegen mit 16.777.215 (24 Bit mit 3×8 Bit pro Kanal) repräsentiert werden. Den richtigen Farbraum zu finden, oder Farben gezielt zu reduzieren (*dithering*) kann von geeigneter Software meist automatisiert erledigt werden, durch genaue Filtereinstellung können Autoren manchmal sogar noch bessere Ergebnisse erreichen.

Ein weiterer (strittiger) Optimierungsschritt der für alle Bildformate funktioniert sind eingebettete Bildprofile. Dies umfasst sowohl Farb- als auch Gammawerte. Das W3C hat bei Einführung des sRGB-Standards alle dazu aufgerufen auf solche Profile zu setzen und unter Fotografen und Künstlern werden sie gerne verwendet damit das Bild auf unterschiedlichen Geräten identisch dargestellt werden. Die Verwendung der Profile kann jedoch zu diversen Problemen führen und das Entfernen die Dateigröße signifikant beeinflussen. An dieser Stelle soll nicht genauer auf die Vor- und Nachteile eingegangen werden (da dies den Rahmen sprengen würde). Es sei nur so viel erwähnt: die meisten Browser beachten keine Profile und wenn sie es machen sorgt es dafür, dass die Bilder andere Farben verwenden als der Browser selbst. Bei Fotos ist dies – da sie gesonderten Inhalt darstellen – in der Regel kein Problem. Bei Icons oder anderen Designelementen führt dies jedoch oft dazu, dass das Bild sich von der Umgebung abhebt, da sich der Farbwert verändert. Bei Gammainformationen kann es noch zu weitaus kurioseeren Ergebnissen kommen, Beispiel:

<http://www.quora.com/Computer-Science/Why-does-this-image-appears-differently-in-Chrome-and-IE>

Um einen Abschluss zu finden soll an dieser Stelle auf zwei einführende und sehr interessante Artikel verwiesen werden, die auf typspezifische Optimierungsmöglichkeiten eingehen:

<http://www.smashingmagazine.com/2009/07/15/clever-png-optimization-techniques/>

<http://www.smashingmagazine.com/2009/07/01/clever-jpeg-optimization-techniques/>

Objektive Qualitätsbestimmung

Das Problem aller Optimierungsversuche ist der mögliche Verlust von Qualität, im Besonderen bei Verwendung von verlustbehafteten Formaten wie JPEG. Aber

⁷ <http://www.w3.org/Graphics/Color/sRGB.html> A „Standard Default Color Space for the Internet – sRGB“

¹ <http://www.cns.nyu.edu/lcv/ssim/> Original Website der Algorithmus-Autoren zum Algorithmus

² <http://www.jpeg.org/> Offizielle Website des JPEG-Komitees

³ <http://www.libpng.org/pub/png/> Offizielle PNG Website

⁴ <http://zlib.net/> Offizielle zlib Website

⁵ <http://www.metadataworkinggroup.org/> Offizielle Website der Metadata Working Group

⁶ <http://www.adobe.com/products/xmp/> Offizielle XMP-Informationssseite bei Adobe

auch bei PNG kann es zu Qualitätsverlust kommen, hierbei speziell bei Verringerung der verwendeten Farben. Die Frage ist, wie kann der Qualitätsverlust objektiv bestimmt werden und diesem Problem haben sich natürlich bereits sehr viele angenommen. Der SSIM stellt eine mögliche Lösung dafür dar (nebst z. B. Algorithmen wie *Peak signal-to-noise ratio* (PSNR) oder Weiterentwicklungen von SSIM selbst). Der Autor hat versucht den originalen Matlab-Algorithmus in PHP umzusetzen (das Programm ist nicht lauffähig), damit er einfach in eine Webanwendung integriert werden kann. Das größte Problem was sich ergeben hat, ist die Tatsache, dass sehr viele Funktionen die Matlab (oder andere Programmiersprachen) bereitstellt in PHP einfach nicht zur Verfügung stehen. Dies liegt darin begründet, dass PHP in erster Linie dafür gedacht ist schnell Webseiten auszuliefern und nicht für wissenschaftliche Arbeiten oder mathematische Kalkulationen konzipiert wurde. So liefern fast alle Bildfunktionen Bilder als Ergebnis zurück und nicht – wie bei anderen Programmiersprachen meist der Fall – eine Matrix. Ein anderes Problem das den effizienten Einsatz jedoch völlig zur Nichte macht sind die aufwändigen Berechnungen. Der Aufwand zur Berechnung des SSIM-Wertes steht somit in keiner Relation zum möglichen Qualitätsgewinn. Selbst das ImageJ-Plugin das in Java umgesetzt wurde und unabhängig vom Bildformat im Speicher mit den Matrizen die Berechnungen durchführt benötigte auf dem High-End-Computer des Autors mehr als eine Sekunde zur Berechnung der SSIM-Werte (das ImageJ-Plugin berechnet SSIM, MS-SSIM und MS-SSIM* zur gleichen Zeit).

Resume

Algorithmen zur Bestimmung der Bildqualität eignen sich nur zum Vergleich von Encodern, nicht jedoch zur Optimierung von Bildkompression im Kontext von Webanwendungen wo Bilder zu tausenden mit unbekanntem Inhalt automatisiert optimiert werden müssen.