**Wissenschaftliche Arbeit zum Thema Bildformate**

**von**

**Sarah Anderhofstadt,**

**Matrikelnummer:**

**Annika Dudenhausen,**

**Matrikelnummer:**

**Geraldine Schmid,**

**Matrikelnummer:182680**

**Gliederung:**

1. Was versteht man unter einem verlustfreien Dateiformat?

**Seite 3**

1. Verlustfrei oder Verlustbehaftet?

**Seite 4**

1. Verschiedene Bildformate

**Seite 4**

* 1. JPEG
  2. TIFF
  3. PNG
  4. GIF

1. Dateiformat BMP im Detail

**Seite 5 – 6**

1. Das RGB Farbmodell

**Seite 6 – 8**

1. Die Farbpalette

**Seite 8**

1. Die Versuchsdurchführung

**Seite 9 – 12**

1. Fazit

**Seite 13**

1. Quellen und Erklärung

**Seite 14**

1. **Was versteht man unter einem verlustfreien Dateiformat?**

Um Dateien vor allem im Internet zu verwenden, ist es von Vorteil wenn sie nur geringe Dateigröße aufweisen. Je größer die Datei, umso länger braucht zum Beispiel die Übertragung der Datei. Wenn man die Dateien komprimiert, um eine kleinere Speicherkapazität zu erreichen, können Verluste entstehen.

Originaldaten Encoder Datei

=

Daten Decoder [[1]](#footnote-1)

An diesem Schaubild wird eine verlustfreie Komprimierung aufgezeigt. Auf die Originaldaten wendet der Encoder einen Algorithmus an. Die Datei hat nun zum Beispiel das Dateiformat BMP. Wenn nun aus den komprimierten Daten wieder alle Originaldaten ausgelesen werden, d.h. Originaldaten = Daten, spricht man von einer verlustfreien Kompression.

Bei der verlustbehafteten Kompression geht ein Teil der Informationen verloren, weil die Originaldaten nicht mehr aus den komprimierten Daten ausgelesen werden können.

Originaldaten Encoder Datei

**≠**

Daten Decoder [[2]](#footnote-2)

1. **Verlustfrei oder Verlustbehaftet?**

Bilddateiformate gibt es viele, im Folgenden werden nun aber nur die Eigenschaften der wichtigsten Dateiformate zum Abspeichern einer Rastergrafik dargestellt:

|  |  |
| --- | --- |
| Verlustfrei | Verlustbehaftet |
| BMP (Bitmap) | JPEG  (Joint Photographic Experts Group) |
| TIFF (Tagged Image File Format) |  |
| PNG (Portable Network Graphics) |  |
| GIF (Graphics Interchange Format) |  |

1. **Verschiedene Bildformate**

JPEG (Joint Photographic Experts Group)

Bilder werden häufig in diesem Format abgespeichert, um sie im Internet anzusehen. Sie zeichnen sich durch ihre geringe Dateigröße aus, sind aber verlustbehaftet.

Das kommt daher, dass die Bilder nach der JPEG-Norm komprimiert werden.

TIFF (Tagged Image File Format)

Bei diesem Format handelt es sich um ein verlustfreies Kompressionsverfahren. Es eignet sich gut zum Austausch von Dateien zwischen unterschiedlichen Programmen und Plattformen da es das CMYK-Farbmodell unterstützt.

Obwohl JPEG deutlich weniger Speichplatz benötigt, wird TIFF trotzdem auch im Internet genutzt, um Bilder in hoher mit hoher Qualität zur Verfügung zu stellen.

GIF (Graphics Interchange Format)

Grafiken im GIF-Format sind geeignet für die Verwendung im Internet. Es ist ein Format mit verlustfreier Kompression, vor allem für Bilder mit geringer Farbtiefe.

PNG (Portable Network Graphics)

Auch PNG wird verlustfrei komprimiert, vor allem im Bereich Web. „PNG ist nicht so verbreitet wie GIF, aber moderner und flexibler.“[[3]](#footnote-3) Das patentfreie Format ist eine Alternative zu dem patentbelasteten GIF-Format und ist weniger komplex als TIFF.

1. **Das BMP-Format im Detail**

Das Bitmap-Format ist ein Windows-Standard-Format. Da das Format nur schwach komprimiert wird, sind die Daten meist größer als z.B. im JPEG-Format. Aus diesem Grund werden BMP-Dateien im Internet kaum eingesetzt. [[4]](#footnote-4)

Das Format ist relativ einfach aufgebaut, was das Einlesen und Speichern vereinfacht. Erlaubt sind Farbtiefen von 1,4,8,16,24 oder 32 bpp (Bit pro Pixel).

Die Besonderheit an BMP ist, sie werden meist unkomprimiert oder verlustfrei mit der Lauflängenkodierung (RLE) gespeichert (oft zum schnellen Zwischenspeichern von Rohdaten).

Der interne Aufbau des Dateiformats BMP:

Der interne Aufbau wird in drei Teile aufgeteilt:

Der erste Teil ist der Dateikopf, der Informationen zum Lesen der Datei enthält.

Der zweite Teil ist der Informationsblock, hier wird festgelegt wie hoch und wie breit die Datei ist und falls nötig die Farbpalette.

Im dritten Teil stehen die Bilddaten.

DATEIKOPF (14 Byte)

INFORMATIONSBLOCK:

(40 Byte)

( Farbpalette:

Farbenzahl \* 4 Byte)

BILDDATEN

Der Dateikopf:

Im Dateikopf wird gekennzeichnet dass es sich um eine Bitmap handelt.

Außerdem enthält er Angaben über die Länge der gesamten BMP-Datei.

Der Informationsblock:

Der Informationsblock ist in zwei Bereiche aufgeteilt. Der erste Bereich zeigt die Höhe und die Breite des Bildes, Informationen über die Anzahl der Bits pro Pixel, die verwendete Kompression und die Bildauflösung in Pixel pro Meter.

Im zweiten Bereich findet man die Farbpalette, falls das Bild aus 2,16 oder 256 Farben besteht. Wenn das Bild aus mehr als 256 Farben besteht, werden die Farben in ihren RGB-Werten abgespeichert.

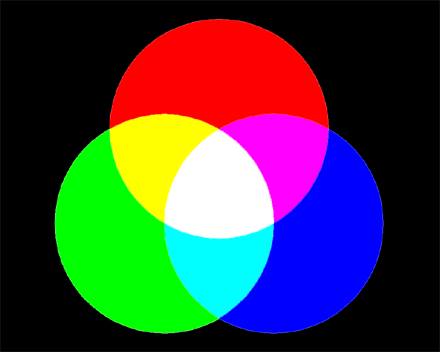
Die Bilddaten:

Hier werden die genauen Bilddaten aufgeführt. Diese werden im Speicher von links unten nach rechts oben abgespeichert. Falls das Bild aus mehr als 256 besteht werden die RGB-Werte jeweils mit einem Speicherplatz von 4 Byte gespeichert.

Die RGB-Werte sind in diesem Fall aber als BGR abgespeichert und dem vierten Byte wird ein Platzhalter „0“ zugewiesen.

**5. Das RGB-Farbmodell**

Bei dem RGB-Farbmodell handelt es sich um ein Standardmodell für additive Farbmischungen. Es basiert auf den drei Grundfarben: Rot, Grün und Blau. Addiert man diese drei Farben, also sobald man diese vermischt, nimmt das menschliche Auge diese Mischung als „Weiß“ wahr. Falls die Addition der drei Grundfarben nicht gleichermaßen erfolgt, werden andere Farbeindrücke gebildet. „Aus Rot und Grün wird Gelb, Rot und Blau ergeben Magenta und Grün und Blau werden zu Cyan.“[[5]](#footnote-5) Wenn kein Licht einstrahlt, sieht der Betrachter lediglich die Farbe „Schwarz“. [[6]](#footnote-6)

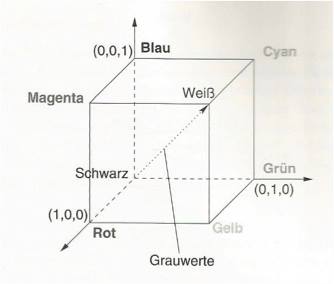


Die additive Farbmischung[[7]](#footnote-7)

Darstellung des RGB-Farbmodells in einem Farbwürfel

Das RGB-Farbmodell besteht, wie bereits erklärt, aus den drei Farben Rot, Grün und Blau. Wie aus diesen Grundfarben 16,7 Millionen Farben entstehen, wird im Folgenden am Beispiel des Farbwürfels genauer veranschaulicht. Innerhalb des Würfels, also im sogenannten **Farbraum**, lassen sich alle aus Rot, Grün und Blau mischbaren Farben darstellen. Ein Würfel besitzt acht Ecken. Jede Ecke des Farbwürfels spiegelt eine unterschiedliche Farbe wider. Hierbei handelt es sich einerseits um die Grundfarben Rot, Grün und Blau, zudem die Farben Weiß und Schwarz. Andererseits werden ebenfalls die aus den Grundfarben resultierenden Farben Cyan, Gelb und Magenta berücksichtigt.

**Der Farbwürfel[[8]](#footnote-8)**

Der Farbwürfel befindet sich in einem dreidimensionalen Koordinatensystem. Da jede Primärfarbe 256 (=28) verschiedene Sättigungswerte annehmen kann, gibt es insgesamt circa 16,7 Millionen (28\*3) verschiedene Farben. (256x256x256)

Dieses Koordinatensystem der additiven Farbmischung endet bereits bei der Zahl 255, da der Ursprung (0/0/0) selbst die Farbe Schwarz definiert und somit mitgezählt wird. Die dezimale Darstellung des RGB-Farbmodells beginnt bei dem Wert 0 und endet mit dem Wert 255.

Die Farbe Rot hat die Koordinaten (255/0/0), Grün die Koordinaten (0/255/0) und Blau die Koordinaten (0/0/255). Durch das Addieren der drei Grundfarben ergibt sich, wie bereits erwähnt, die Farbe Weiß, welche somit die Koordinaten (255/255/255) hat.

RGB-Farbsystem in Hexadezimal-Darstellung

|  |  |
| --- | --- |
| Dezimal | Hexadezimal |
| 0 | 0 |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |
| 6 | 6 |
| 7 | 7 |
| 8 | 8 |
| 9 | 9 |
| 10 | A |
| 11 | B |
| 12 | C |
| 13 | D |
| 14 | E |
| 15 | F |

Um die Darstellung des RGB-Farbsystems zu erleichtern, wird die Dezimaldarstellung der Koordinaten nun in eine Hexadezimal-Darstellung umgerechnet. Das Hexadezimalsystem beruht auf der Basis b = 16. Die Zahlenmenge beinhaltet die Werte von {0;1;2;3;4;5;6;7;8;9;10;11;12;13;14;15}.

Zur Veranschaulichung der Rechnung wurde die Farbe Rot ausgewählt.

x1  x2 x3

*Dezimale Darstellung der Farbe Rot:* 255, 0, 0

*Umrechnung in Hexadezimalsystem:* x1= 255

Von unten nach oben lesen!

255 div 16 = 15 Rest 15

15 div 16 = 0 Rest 15

* 15 entspricht F im Hexadezimalsystem

*Hexadezimaldarstellung* der Stelle x1 = FF

x2= 0

0 div 16 = 0 Rest 0

***Hexadezimale Darstellung der Farbe Rot:* FF 00 00**

**6. Die Farbpalette**

Wenn weniger als 256 Farben genügen um ein Bild mit ausreichender Qualität darstellen zu können, wird eine Farbpalette verwendet. In dieser können die notwendigen Farben abgespeichert und mit einem Index versehen werden.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index | R | G | B | Farbe |
| 0 | 0 | 0 | 0 | Schwarz |
| 1 | 255 | 0 | 0 | Rot |
| 2 | 255 | 255 | 255 | Weiß |
| 3 | 0 | 0 | 255 | Blau |
| … |  |  |  |  |

Beispiel für den Aufbau einer Farbpalette

Bei einer Bitmap-Datei wird beispielsweise die Farbpalette nur bei 2,16 und 256 Farben verwendet. Bei mehr als 256 Farben existiert dagegen keine Farbpalette - die Farben werden in RGB abgelegt. „Farbpaletten ermöglichen eine Datenreduktion von beispielsweise 24 Bits auf 8 Bits pro Bildpunkt. Hinzu kommt noch die Tabelle mit den Farbtripeln.“[[9]](#footnote-9)

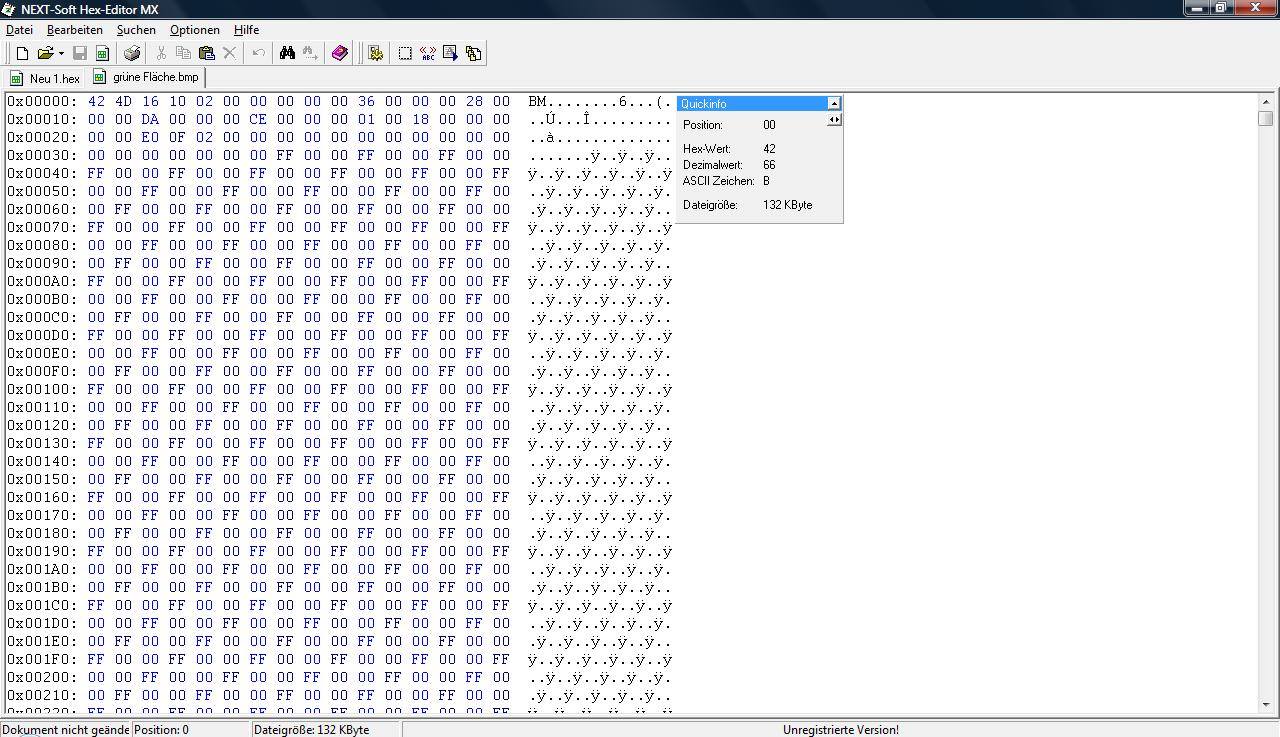
Da verlustbehaftete Komprimierungsmethoden meist nur mit einer eingeschränkten Qualität möglich sind kommen in der Praxis hauptsächlich verlustfreie Verfahren vor.

**7. Die Versuchsdurchführung**



Als erstes wird eine reingrüne (#00FF00) Fläche in Paint erstellt, die anschließend in einem 24-bit-Bitmap-Format abgespeichert wird.

Voraussetzung für den Versuchsverlauf ist das Installieren des Hex-Editors. Darunter versteht man ein Computerprogramm, das es ermöglicht „beliebige Dateien als Folgen von Hexadezimal-Zahlen anzeigen [zu] lassen oder [zu] bearbeiten.“[[10]](#footnote-10)



[Beispiel des grünen Bildes im Hex-Editor dargestellt]

Wie unter dem Punkt „Interner Aufbau des Dateiformats bmp“ bereits beschrieben, ist dieses Format in folgende Bereiche aufgeteilt:

1. **Der Dateikopf:**

2 Byte: „BM“ [42 4D

4 Byte: Länge der Datei 16 10

2 Byte: Reserviert = 0

2 Byte: Reserviert = 0

4 Byte: Beginn der Rasterdaten

1. **Der Informationsblock:**

4 Byte Länge des Informationsblocks (Windows = 40)

4 Byte Breite des Bildes

4 Byte Höhe des Bildes

2 Byte Zahl der Ebenen (Windows = 1)

2 Byte Farbbits je Pixel [Anzahl der Farben] (1 [2], 4 [16], 8 [256],

24 [16 Mio])

4 Byte Kompression (0 = ohne, 1 = 8 Bit RLE, 2 = 4 Bit RLE)

4 Byte Größe des unkomprimierten Bildes

4 Byte horizontale. Anzahl von Rasterpunkte pro Meter

4 Byte vertikale Anzahl von Rasterpunkte pro Meter

4 Byte verwendete Farben (0 = alle)

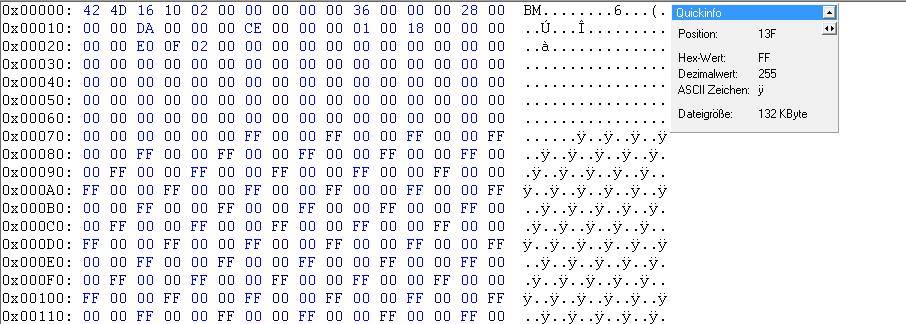
4 Byte wichtige Farben (0 = alle)

anschließend folgen die Farbwerte, 4 Bytes je Farbe.

1. **Die Bilddaten** enden vor 00FF00, also vor der Darstellung der Farbe Grün.

Nun werden mit Hilfe des Hex-Editors mehrere Farbwerte geändert. Um die Veränderung und die Stelle der Veränderung zu veranschaulichen, wurden ab dem Anfang der Bilddaten mehrere Farbpixel verändert. Im Folgenden wird die Veränderung sichtbar:

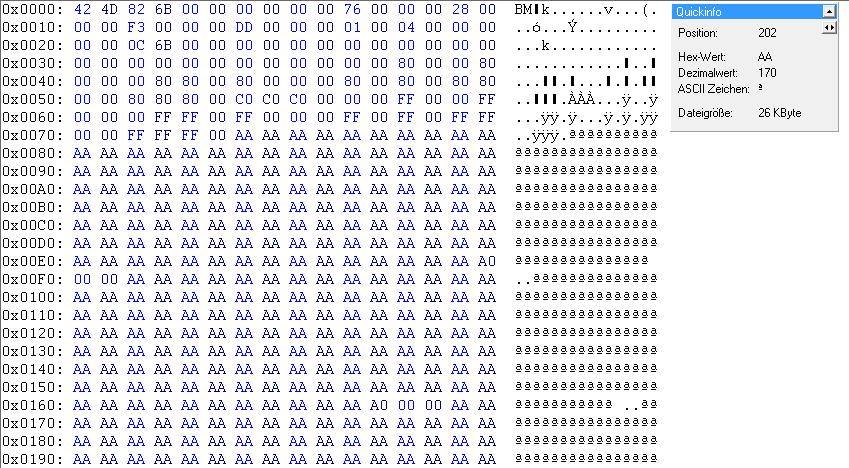
 [Veränderte Bilddatei]



[Veränderung des Codes im Hex-Editor]

In diesem Beispiel wird deutlich, dass die Daten von Unten nach Oben und von links nach rechts gelesen werden. Die Änderung der Farbwerte im Hex-Editor am Anfang der Bilddaten verursacht eine Änderung der Farbwerte am Ende der Bitmap-Datei.

Die rein-grüne Fläche wird nun im 16-Bit-Bitmap-Format abgespeichert um die Veränderung zum 24-Bit-Format zu überprüfen. Es ist davon aus zu gehen, dass wie unter Punkt 4 beschrieben, die Farbpalette im Informationsblock aufzufinden ist.



[16-Bit-Bitmap Datei im Hex-Editor dargestellt / Farbpalette in rot umrandet]

9

10

A

B

C

D

E

F

0

1

2

3

4

5

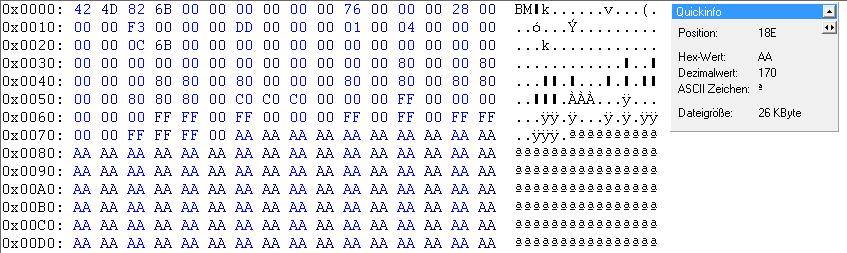
6

7

8

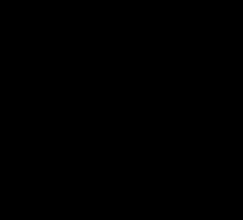
**Diese Werte sind in dieser Reihenfolge in der Farbpalette (s.o.) aufgeführt.**

In der Farbpalette wird deutlich, dass das reinegrün (#00FF00) den Hexadezimalwert „A“ hat. Aus diesem Grund ist die grüne Fläche in der Hex-Editor Ansicht als A dargestellt.

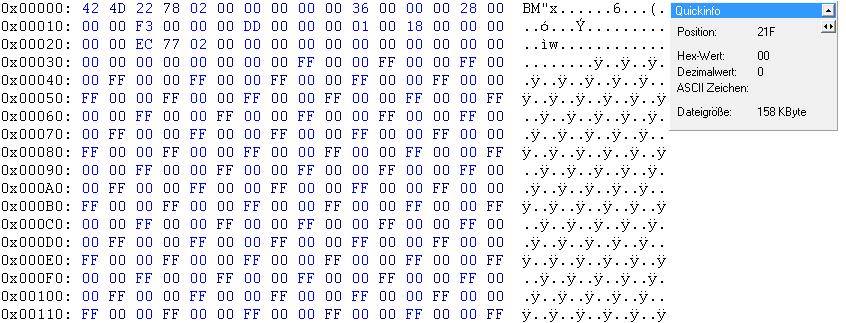


An dieser Stelle wird das Bild komplett verändert. Indem man nur die Farbe an der Stelle 10 der Farbpalette (Da das Reinegrün an der Stelle A=10 ist) ändert. Die Farbe wurde von #00 FF 00 00 zu # 00 00 00 00 geändert.

Das Ergebnis ist eine Reinschwarze Fläche:



Um zusätzlich noch eine andere Farbe zu veranschaulichen, wird nun eine Reinrote Fläche im Hex-Editor betrachtet:

[Reinrot im 24-Bit-Bitmap-Format im Hex-Editor dargestellt]

Auffällig ist, dass die Farbe Rot in den Bilddaten nicht wie gewöhnlich in #FF 00 00 dargestellt wird, sondern in #00 00 FF. Das heißt, das RGB-Farbsystem wurde „umgedreht“ in ein „BGR-Farbsystem“.

**8. Fazit**

Im 16-Bit-System hat die Farbe Reingrün A nur eine Stelle. A im Dualsystem entspricht #1010, jede Stelle stellt ein Bit dar. D.h. A hat eine Größe von 4 Bit.

Im 24-Bit-System hingegen hat die Farbe Reingrün sechs Stellen (Hexadezimal: # 00 FF 00) á 4 Bit. Somit ergeben sich 24-Bit (6x4 Bit).

Vergleicht man die beiden Dateiformate wird deutlich, dass die Datei im 24-Bit-System sechsmal größer ist als die Datei im 16-Bit-System.

**Vor- und Nachteile des BMP-Formats:**

|  |  |
| --- | --- |
| Vorteile | Nachteile |
| Einfache Dateistruktur | große Dateigröße im Vergleich zu komprimierten Formaten[[11]](#footnote-11) |
| Verlustfreie Kompression |  |
| Änderung der Farbinformation durch die Änderung der Farbpalette ohne Bilddaten zu ändern |  |

Aus dieser Tabelle wird ersichtlich, dass das BMP-Format mehr Vorteile als Nachteile aufweist.

**9. Quellen**

*Printmedien:*

* Mitschrieb aus der Vorlesung „Medientechniktheorie“ von Herr Prof. Dr. Gerlach
* Rainer Malaka, Andreas Butz, Heinrich Hußmann: Medieninformatik – Eine Einführung, München 2009, Seite 92
* Tilo Strutz: Bilddatenkompression, Wiesbaden 3. Auflage 2005, Seite 155

*Elektronische Medien:*

* http://www.chip.de/downloads/BMP-Datei-oeffnen\_50849943.html

(entnommen am 09.01.2014)

* http://lehrerfortbildung-bw.de/kompetenzen/gestaltung/farbe/physik/additiv/ (entnommen am 09.01.2014)
* http://www.chip.de/downloads/Hex-Editor-MX\_30351843.html (entnommen am 10.01.2014) (entnommen am 10.01.2014)
* http://www.it-service24.com/lexikon/b/bitmap/ Bitmap (entnommen am 10.01.2014)

**Erklärung:**

Wir versichern hiermit, dass wir diese Arbeit selbstständig angefertigt und keine anderen als von uns angegebenen Quellen und Hilfsmittel verwendet haben. Die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen sind als solche gekennzeichnet.

Neu-Ulm, den 20. Januar 2014

1. Grafik aus Vorlesung Medientechnik Professor Gerlach [↑](#footnote-ref-1)
2. Grafik aus Vorlesung Medientechnik Professor Gerlach [↑](#footnote-ref-2)
3. Rainer Malaka, Andreas Butz, Heinrich Hußmann: Medieninformatik – Eine Einführung, München 2009, Seite 112 [↑](#footnote-ref-3)
4. http://www.chip.de/downloads/BMP-Datei-oeffnen\_50849943.html

   (entnommen am 09.01.2014) [↑](#footnote-ref-4)
5. Rainer Malaka, Andreas Butz, Heinrich Hußmann: Medieninformatik – Eine Einführung, München 2009, Seite 92 [↑](#footnote-ref-5)
6. Vgl. Rainer Malaka, Andreas Butz, Heinrich Hußmann: Medieninformatik – Eine Einführung, München 2009, Seite 92 [↑](#footnote-ref-6)
7. http://lehrerfortbildung-bw.de/kompetenzen/gestaltung/farbe/physik/additiv/ (entnommen am 09.01.2014) [↑](#footnote-ref-7)
8. Tilo Strutz: Bilddatenkompression, Wiesbaden 3. Auflage 2005, Seite 155 [↑](#footnote-ref-8)
9. Tilo Strutz: Bilddatenkompression, Wiesbaden 3. Auflage 2005, Seite 155 [↑](#footnote-ref-9)
10. http://www.chip.de/downloads/Hex-Editor-MX\_30351843.html (entnommen am 10.01.2014) [↑](#footnote-ref-10)
11. http://www.it-service24.com/lexikon/b/bitmap/ (entnommen am 10.01.2014) [↑](#footnote-ref-11)