

ÜBUNG ZUR VORLESUNG
MULTIMEDIA IM NETZ

Ludwig-Maximilians-Universität
Wintersemester 2010/2011

STEGANOGRAPHIE

- Steganographie ist die verborgene Speicherung oder Übermittlung von Informationen.
- Wörtlich aus dem Altgriechischen „bedeckt schreiben“.
- Steganographie baut darauf, dass Dritte nicht bemerken, dass eine Nachricht übermittelt wird.
- Zum Vergleich: In der Kryptographie kann ein Dritter wissen, dass eine Nachricht übermittelt wird, aber er kann den Inhalt nicht entziffern.

DATEN IN BILDERN VERSTECKEN

- Menschliches Auge ist gegen Bildrauschen relativ unempfindlich
→ Foto kann verändert werden, ohne dass Veränderung auffällig oder störend ist
- Eine Möglichkeit: Informationen in Farbkanälen verstecken
- Ändert man in jedem Farbkanal den Wert um eins, ändert sich die Farbe im gesamten Bildpunkt um $1/255$
→ 0,39%

FARBKANÄLE MANIPULIEREN

- Ein Pixel besteht aus einem Alphakanal und drei Farbkanälen (Rot, Grün, Blau).
- Ein Kanal besteht aus 1 Byte.
- Einem Buchstaben ist laut ASCII Tabelle genau eine Zahl zugeordnet, die sich binär mit einem Byte darstellen lässt.

BEISPIEL

- Farbe rechts hat folgende Werte:

- Rot: 191 → 10111111
- Grün: 29 → 00011101
- Blau: 189 → 10111101

→ Man kann einen oder mehrere Kanäle verwenden, um Teile einer Nachricht zu verstecken

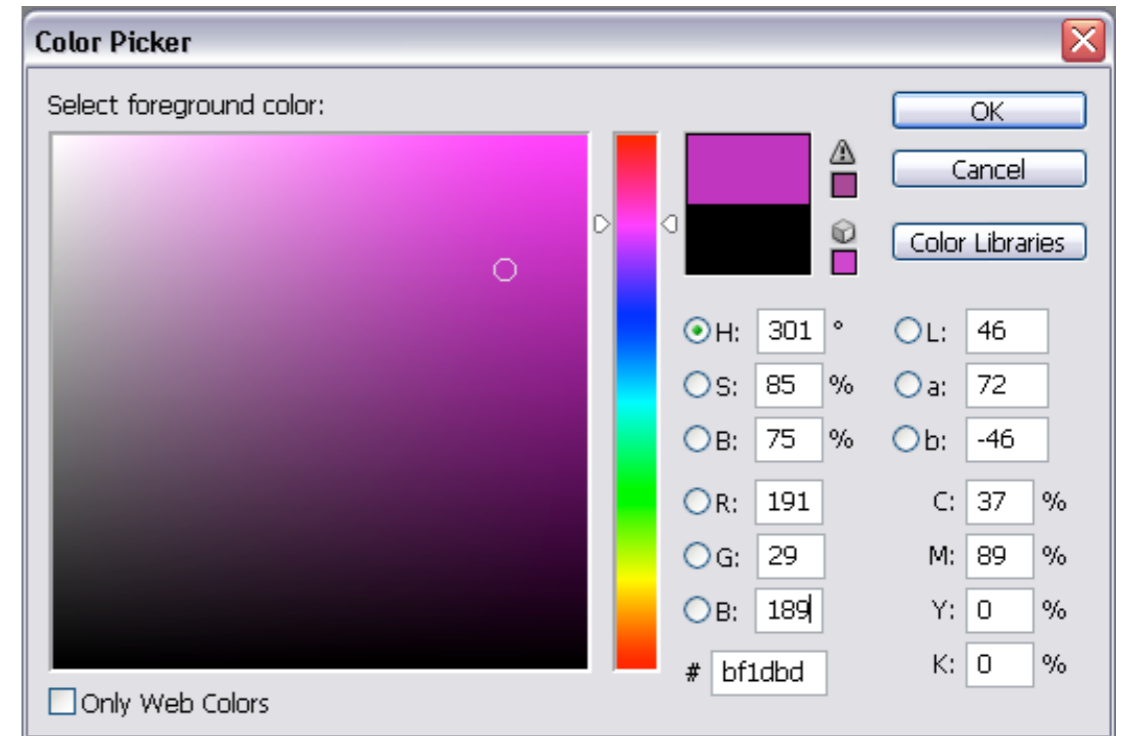
- Der Buchstabe „A“ hat den ASCII Wert 65 → 01000001
→ als erstes kodiert man die erste Stelle, also „0“

- Das erste Bit des „A“ wird in die letzte Stelle des Blaukanals kodiert
→ der neue Wert des Blaukanals ist: 10111100

- Als nächstes wird das zweite Bit des „A“, also „1“, in den Grünkanal kodiert

- Als nächstes wird das dritte Bit des „A“, also „0“, in den Rotkanal kodiert

- Danach macht man mit dem nächsten Pixel weiter



VERSCHIEBUNGSOPERATOREN

- SHIFT RIGHT >>

Beispiel: **01000001** >> 1 → **00100000**
01000001 >> 7 → **00000000**

- SHIFT LEFT <<

Beispiel: **01000001** << 1 → **10000010**
01000001 << 5 → **00100000**

LOGISCHE BIT-OPERATIONEN

- Und „&“

Beispiel:

$$\begin{array}{r} 10011010 \\ \& 10101011 \\ \hline = 10001010 \end{array}$$

- Oder „|“

Beispiel:

$$\begin{array}{r} 10011010 \\ | 10101011 \\ \hline = 10111011 \end{array}$$

- XOR (Exclusives Oder) „^“

Beispiel:

$$\begin{array}{r} 10011010 \\ ^ 10101011 \\ \hline = 00110001 \end{array}$$

VERSTECKEN

Pixel: ARG10111101 A: 01000001 → Ziel ARG10111100

1. $A \gg 7 \rightarrow 00000000$

2. $A \& 1$

00000000

&00000001

=00000000 (→ modifiziertes „A“)

(Setzt die ersten sieben Stellen auf „0“. Die Letzte, also die von Interesse, bleibt erhalten.)

3. Wert des Pixel & 1

ARG10111101

&000000000001

=000000000001 (→ temp. Pixel)

4. Überprüfen, ob der Wert des temp. Pixel gleich dem Wert des modifizierten „A“ ist

- Wenn „Ja“ → Weiter zum nächsten Bit des „A“ und zum nächsten Farbkanal (falls alles drei Kanäle bereits belegt worden sind, zum nächsten Pixel)
- Wenn „Nein“
 - Ist modifiziertes „A“ gleich 1 → Pixel = Pixel + 1;
 - Ist modifiziertes „A“ gleich 0 → Pixel = Pixel - 1 ;
 - Hier: mod. „A“ ist 0 → 10111101-1 = 10111100 (→ neuer Blaukanal)

FARBKANAL AUSWÄHLEN

- ein Pixel besteht aus einem Alphakanal und 3 Farbkänen (Rot, Grün, Blau) (siehe Folie 4)

A	R	G	B
1 Byte	1 Byte	1 Byte	1 Byte

→ 32 Bit um Alphakanal und RGB darzustellen

- eine Möglichkeit, um einen Farbkänel auszuwählen und den Wert zu ändern, ist die Verwendung von SHIFT RIGHT
 - Blaukanal: $(ARG11000110 \gg 0) = ARG11000110$
 - Grünkanal: $(AR11000110B \gg 8) = 00000000 AR 11000110$
 - Rotkanal: $(A11000110GB \gg 16) = 00000000 00000000 A 11000110$
- Dadurch steht das Bit, das man verändern möchte, immer am Ende
- Möchte man den neuen Farbkänel setzen, muss der entsprechende SHIFT LEFT angewandt werden (0 bei Blau, 8 bei Grün und 16 bei Rot).

AUSLESEN

Gesuchter Buchstabe: wird Anfangs mit 0 initialisiert
Eingelesener Pixel, z.B.: ARG10111100
→ Ziel: erstes Bit vom ursprünglichen „A“ finden

1. Eingelesener Pixel & 1

$$\begin{array}{r} \text{ARG10111100} \\ \& \text{00000000001} \\ \hline = \text{00000000000} \quad (\rightarrow \text{temp. gesuchter Buchstabe}) \end{array}$$

2. Temp. gesuchter Buchstabe << 7

$$\rightarrow \text{00000000000} \ll 7 = \text{00000000000} \quad (\rightarrow \text{temp. gesuchter Buchstabe geshiftet})$$

3. Gesuchter Buchstabe | temp. gesuchter Buchstabe geshiftet

$$\begin{array}{r} \text{00000000000} \\ | \text{00000000000} \\ \hline = \text{00000000000} \quad (\rightarrow \text{erstes Bit vom gesuchten Buchstaben; wird beim nächsten Durchlauf als Wert für „gesuchter Buchstabe verwendet}) \end{array}$$