

Computergrafik.Online

Drehbuch Bilddatenreduktion

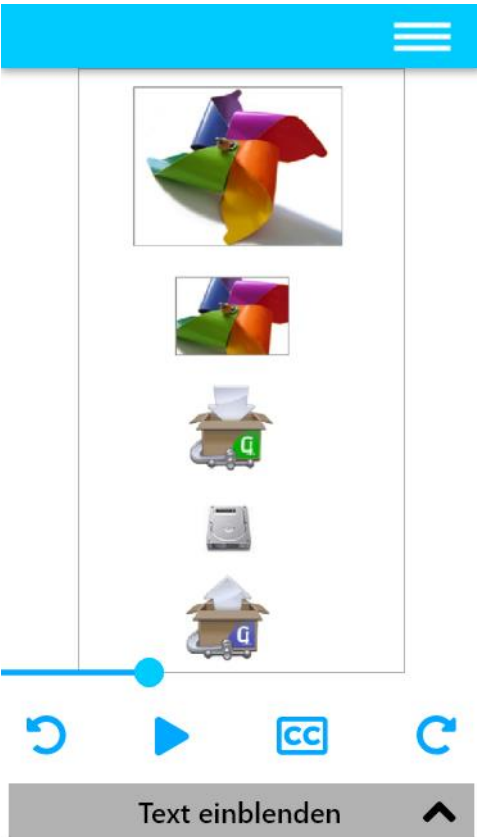
Hochschule Furtwangen University
Fakultät Digitale Medien
Betreut von:
Prof. Jirka Dell'Oro-Friedl

Version: 2.0
Letzte Änderung: 21.10.2018
Autor: Steven Romanek


Inhalt

6.1 (A) Einleitung.....	2
6.2 (A) Skalieren und Beschneiden	3
6.3 (A) Farbtiefenreduktion	4
6.4 (I) Farbtiefenreduktion.....	5
6.5 (A) RLE	6
6.6 (A) LZW	7
6.7 (A) Huffman-Kodierung.....	8
6.8 (A) JPG	9
6.9 (I) JPG	11
6.10 (A) GIF.....	12
6.11 (A) PNG.....	14
6.12 (A) Tipps & Tricks.....	16

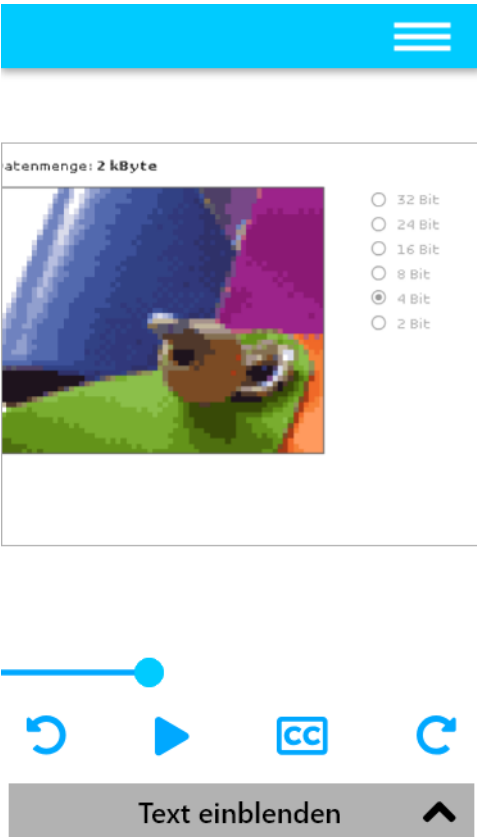
6.1 (A) Einleitung

Screen	Sprechertexte	Stichwörter / Notizen	Regieanweisungen
	<p>130101 Beim Arbeiten mit digitalen Bildern können sehr schnell große Datenmengen entstehen. Um dennoch eine kompakte Speicherung und schnelle Übertragung zu gewährleisten, ist es häufig sinnvoll, die Bilddaten zu reduzieren.</p> <p>Zunächst kann das Bild auf einen relevanten Bereich zugeschnitten, skaliert und in der Farbtiefe reduziert werden. Anschließend wird es komprimiert, also platzsparend kodiert, wobei zwischen verlustfreier und verlustbehafteter Kompression gewählt werden kann.</p> <p>Nun ist das Bild bereit um abgespeichert oder verschickt zu werden. Will man es nun wiederverwenden, so wird es vorher dekomprimiert. Bei der verlustfreien Kompression kann das Bild dabei vollständig wiederhergestellt werden, während dies bei der verlustbehafteten nicht möglich ist.</p>	<p>130101 -Zuschneiden, skalieren oder Farbtiefe reduzieren</p> <p>- Verlustfreie oder verlustbehaftete Kompression</p>	<p>130101 Die einzelnen Kompressionsschritte werden nacheinander eingeblendet.</p>

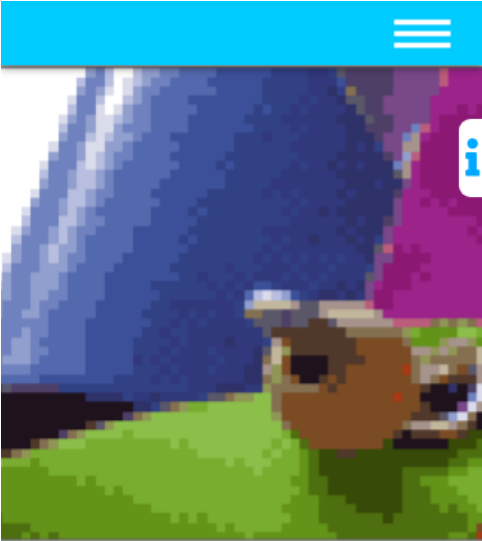
6.2 (A) Skalieren und Beschneiden

Screen	Sprechertexte	Stichwörter / Notizen	Regieanweisungen
	<p>130201 Bilddatenreduktion muss nicht automatisch Kompression bedeuten. Bilddaten können durch andere Methoden ebenfalls reduziert werden. Zum Beispiel indem die Pixelanzahl verringert wird. Beschneidet man das Bild, und skaliert es anschließend auf eine geringere Kantenlänge, so wird die Bilddatenmenge erheblich reduziert. Dabei muss aber drauf geachtet werden, welche Skalierungsmethode gewählt ist.</p> <p>Pixelwiederholung ist eine Methode, die Pixel beim Verkleinern weglässt, was unschöne Effekte auf das Bild haben kann.</p> <p>Interpolation hat in der Regel ein schöneres optisches Ergebnis zur Folge, erzeugt aber neue Farbzwischenstufen. Im Gegensatz zur Pixelwiederholung, hat das Bild nun mehr Farben als vor der Skalierung.</p>	<p>130201 -Skalieren und bescheiden verringert die Datenmenge stark</p> <p>-2 Arten der Skalierung: Pixelwiederholung und Interpolation</p>	<p>130201 Das Skalieren und Beschneiden wird visuell dargestellt.</p>

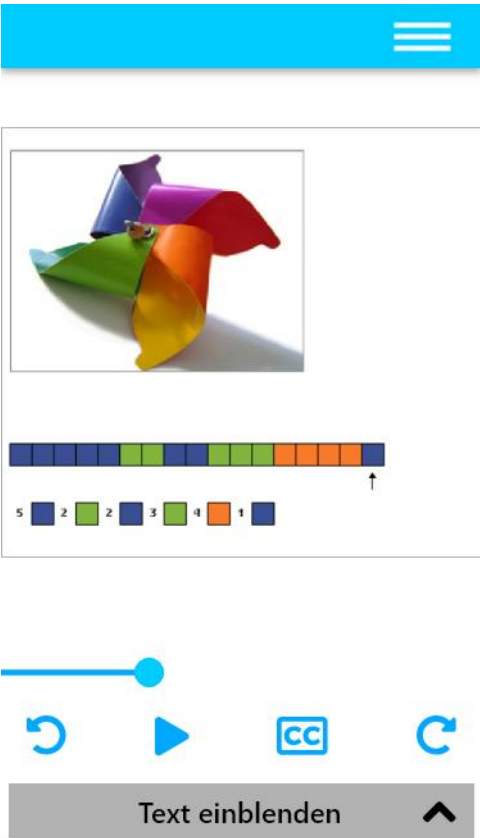
6.3 (A) Farbtiefenreduktion

Screen	Sprechertexte	Stichwörter / Notizen	Regieanweisungen
 <p>atenmenge: 2 kByte</p> <p> <input type="radio"/> 32 Bit <input type="radio"/> 24 Bit <input type="radio"/> 16 Bit <input checked="" type="radio"/> 8 Bit <input type="radio"/> 4 Bit <input type="radio"/> 2 Bit </p> <p>Text einblenden</p>	<p>130301</p> <p>Die Farbtiefe ist die Anzahl der Bits die pro Pixel zur Speicherung der Farbinformation zu Verfügung stehen. Üblich sind Farbtiefen wie 8,16, 24 oder 32 Bit.</p> <p>Häufig überschreitet die Farbtiefe allerdings die vom Bild benötigten Farben. Bei 32 Bit Farbtiefe ist das vierte Byte in der Regel für einen unsichtbaren Transparenzkanal, denn sogenannten Alpha Kanal, vorgesehen. Diese Farbtiefe ist nur für bestimmte Aufgaben notwendig. Eine Reduktion auf 24 Bit spart in einem solchen Fall Speicherplatz.</p> <p>Reduziert man die Farbtiefe weiter, so spart man noch mehr Speicherplatz. Es können aber unter Umständen unschöne Farbverschiebungen auftreten.</p>	<p>130301</p> <p>- Gibt Bits pro Pixel an</p> <p>- Alpha Kanal für Transparenz möglich</p> <p>- Zu geringe Farbtiefe -> Farbverschiebungen</p>	<p>130301</p> <p>Das Beispielbild wird in verschiedenen Farbtiefen angezeigt.</p>

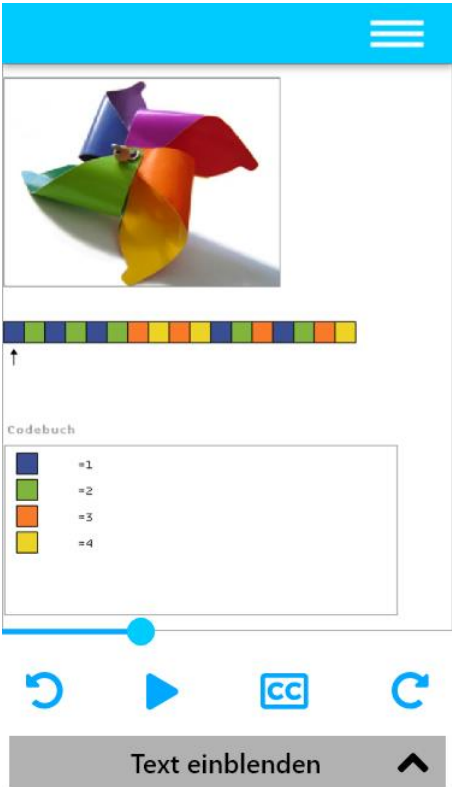
6.4 (I) Farbtiefenreduktion

Screen	Sprechertexte	Stichwörter / Notizen	Regieanweisungen
 <p>Datenmenge: 2 kByte</p> <p>2 Bit <input type="radio"/> 16 Bit <input type="radio"/></p> <p>4 Bit <input checked="" type="radio"/> 32 Bit <input type="radio"/></p> <p>8 Bit <input type="radio"/> 64 Bit <input type="radio"/></p>	<p>130401 Probieren Sie nun selbst die Farbtiefe einzustellen, beobachten sie dabei die jeweilige Datenmenge.</p>		<p>130401 Man kann die verschiedenen Farbtiefen einstellen und somit sehen, welchen Einfluss die Bit-Angabe auf das Endergebnis hat. Außerdem wird auch der verbrauchte Speicherplatz angezeigt.</p>

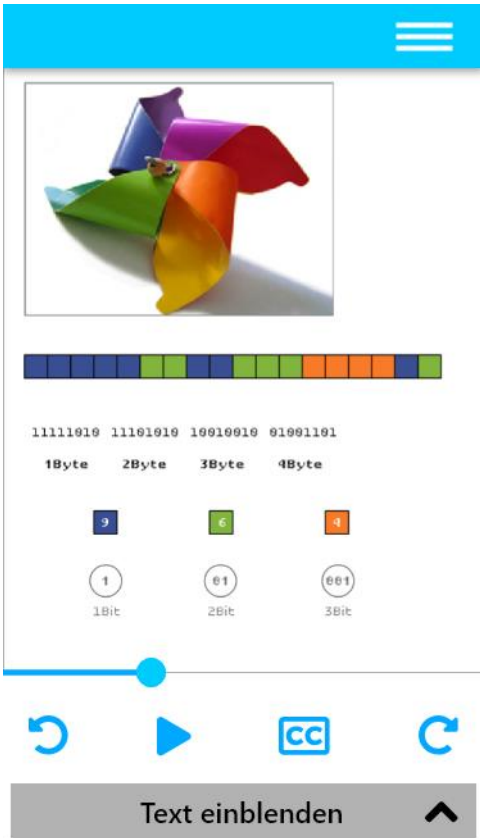
6.5 (A) RLE

Screen	Sprechertexte	Stichwörter / Notizen	Regieanweisungen
 <p>The screenshot shows a presentation interface. At the top is a blue header bar with a white menu icon. Below it is a 3D image of several colorful paper strips (blue, green, yellow, orange, red) arranged in a fan-like shape. Underneath the image is a horizontal bar composed of small colored squares (blue, green, yellow, orange, red) with a small upward-pointing arrow at the right end. Below this bar is a sequence of colored squares with numbers written below them: 5 (blue), 2 (green), 2 (green), 3 (blue), 4 (green), 1 (orange), 1 (blue). At the bottom of the slide area, there are four circular icons: a back arrow, a play button, a Creative Commons (CC) logo, and a refresh/circular arrow icon. Below these icons is a grey button with the text 'Text einblenden' and an upward-pointing arrow.</p>	<p>130501 RLE steht für run-length encoding, also Lauflängenkodierung und ist der einfachste der verlustfreien Kompressionsalgorithmen. Er untersucht die Daten des Bildes und fasst aufeinander folgende gleichfarbige Pixel zusammen. Dabei wird die Farbe einmal abgespeichert sowie ein Zahlenwert angegeben, der beschreibt in wie vielen Pixeln die Farbe hintereinander auftaucht. RLE ist eine sehr einfache Form der Kompression, kann aber bei bestimmten Bildern äußerst effektiv sein.</p>	<p>130501 - run-length encoding - Fasst Pixel zusammen</p>	<p>130501 Das Zählen der Pixel wird visuell dargestellt.</p>

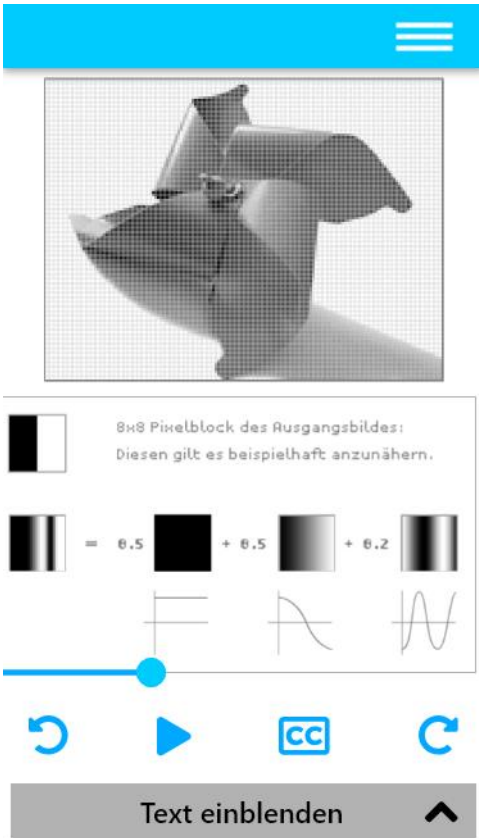
6.6 (A) LZW

Screen	Sprechertexte	Stichwörter / Notizen	Regieanweisungen
	<p>130601</p> <p>LZW ist ein weiterer verlustfreier Kompressionsalgorithmus, der nach seinen Erfindern Lempel, Ziv und Welch benannt ist. Dieser Algorithmus untersucht die Pixel eines Bildes auf sich wiederholende Muster. Ausgehend von den im Bild vorhandenen Farben, wird ein sogenanntes Codebuch angelegt. Dieses wird erweitert, wenn der Algorithmus das Bild auf Pixelkombinationen untersucht. Findet er beim Codieren eine unbekannte Pixelkette, so speichert er den zuletzt gefunden bekannten Wert. Die unbekannte Pixelkombination wird anschließend im Codebuch abgelegt. Anschließend, wenn die gleiche Pixelkette noch einmal gefunden wird, verweist LZW nur noch auf den Eintrag im Codebuch, was eine Einsparung an Speicherplatz bedeutet.</p> <p>Diese Form der Kompression gibt nur bei bestimmten Bildinhalten ein optimales Ergebnis, schafft aber in der Regel eine bessere Datenreduktion als RLE.</p>	<p>130501</p> <ul style="list-style-type: none"> - Verlustfrei - Pixelkombinationen werden im Codebuch vermerkt - Besser als RLE 	<p>130501</p> <p>Die Einträge ins Codebuch werden visuell dargestellt.</p>

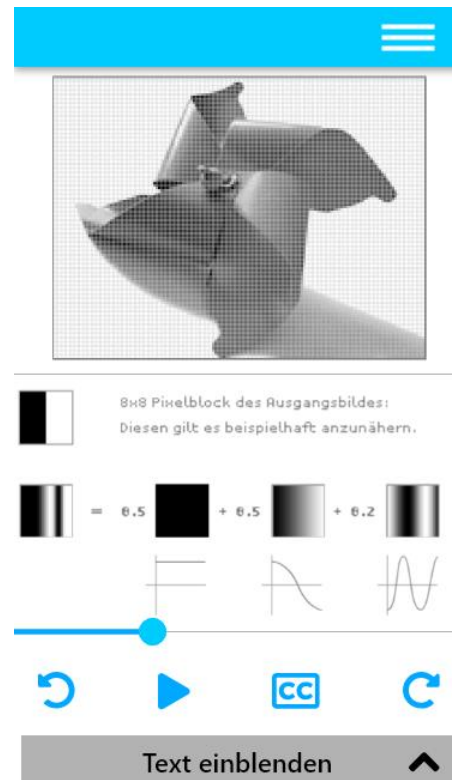
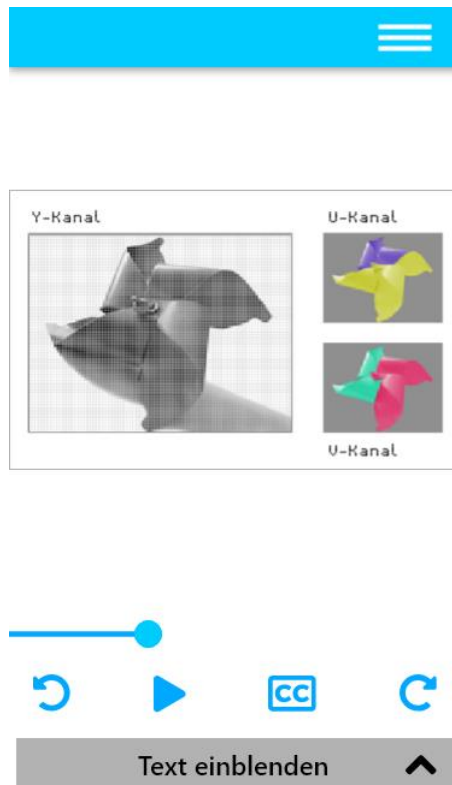
6.7 (A) Huffman-Kodierung

Screen	Sprechertexte	Stichwörter / Notizen	Regieanweisungen
 <p>The screenshot shows a software interface for Huffman coding. At the top, there's a blue header with a menu icon. Below it is a 3D rendered image of a colorful, abstract object. Underneath the image is a horizontal bar representing color frequencies, with segments in blue, green, orange, and red. Below the bar is a table of bit codes for different colors. The table has four columns: 1Byte, 2Byte, 3Byte, and 4Byte. The rows show the bit codes for each color. Below the table are three circular buttons labeled 1 (1Bit), 01 (2Bit), and 001 (3Bit). At the bottom, there are navigation icons (back, play, CC, forward) and a 'Text einblenden' button with an upward arrow.</p>	<p>130701 Die Huffman-Kodierung ist ebenfalls ein verlustfreier Kompressionsalgorithmus. Bei dieser Form der Kodierung nutzt man die Häufigkeit einzelner im Bild auftretender Farben aus.</p> <p>Der Algorithmus untersucht zuerst das ganze Bild und ermittelt die Häufigkeit jeder einzelnen Farbe. Dann wird jeder Farbe ein Bit-Wert zugewiesen. Häufig auftretende Farben werden mit möglichst wenigen Bits dargestellt, während seltenere Farben mit mehr Bits repräsentiert werden. Die häufigste Farbe kann somit nur mit einem einzigen Bit kodiert werden.</p> <p>Die hier dargestellten farbigen Pixel erzeugen dabei einen Code von nur 4 Byte Länge.</p>	<p>130701 - Verlustfrei</p> <p>- Farben bekommen Bit-Wert - Häufigste Farbe hat 1 Bit</p>	<p>130701 Das Zählen der Farben wird visuell dargestellt. Zum Schluss werden die Ergebnisse in Byte eingeblendet.</p>

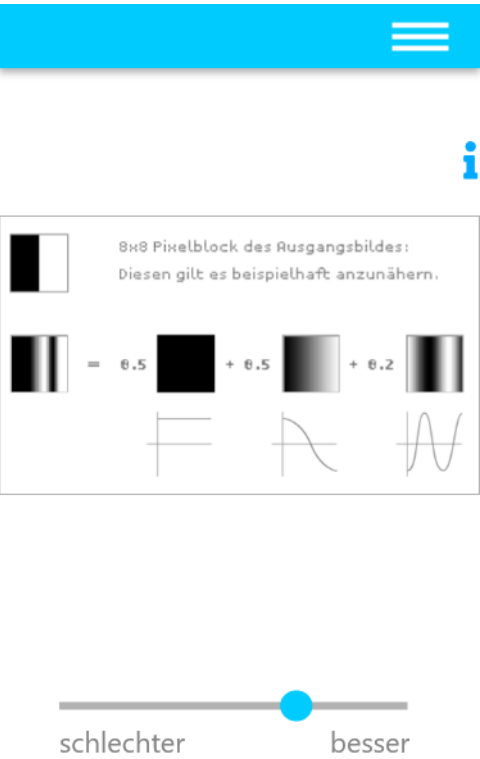
6.8 (A) JPG

Screen	Sprechertexte	Stichwörter / Notizen	Regieanweisungen
	<p>130801 JPG ist ein verlustbehaftetes Grafikformat, bei dem mehrere Kompressionsvorgänge durchgeführt werden. Verlustbehaftete Kompression bedeutet, dass bei der Kompression Bildteile zusammengefasst oder weggelassen werden. Dies geschieht hinsichtlich der optischen Ansprüche eines Menschen. Dadurch kann eine sehr kleine Datenmenge entstehen, die nach der Dekompression dem Original nur noch ähnlich ist.</p> <p>Bei JPG im Speziellen läuft es folgendermaßen ab. Zuerst wird das Bild vom RGB in den YUV Farbraum umgerechnet, hierbei findet die erste Reduktion statt, da die Farbinformation U und V mit geringer Auflösung gespeichert werden.</p> <p>130802 Als Nächstes kommt es zur diskreten Kosinustransformation, oder kurz DCT genannt, die anhand des Y-Kanals beispielhaft gezeigt wird. Zunächst wird das Bild in 8x8 Pixel große Blöcke eingeteilt. Nun wird versucht, jeden Block mit Hilfe von Verläufen mathematisch anzunähern, diese Verläufe basieren auf einfachen vordefinierten Kosinuskurven. Verschiedene</p>	<p>130801 - Verlustbehaftet</p> <p>- Transformation von RGB zu YUV</p> <p>130802 - Diskrete Kosinustransformation</p> <p>- Annäherung durch mathematische Verläufe</p>	<p>130801 Die Konversion des Beispielfotos von RGB in den YUV Farbraum wird gezeigt.</p> <p>130802 Dann wird gezeigt wie das Bild in 8x8 Pixel große Blöcke eingeteilt wird. Anhand eines Blocks werden nach und nach die verschiedenen Verläufe gezeigt.</p>

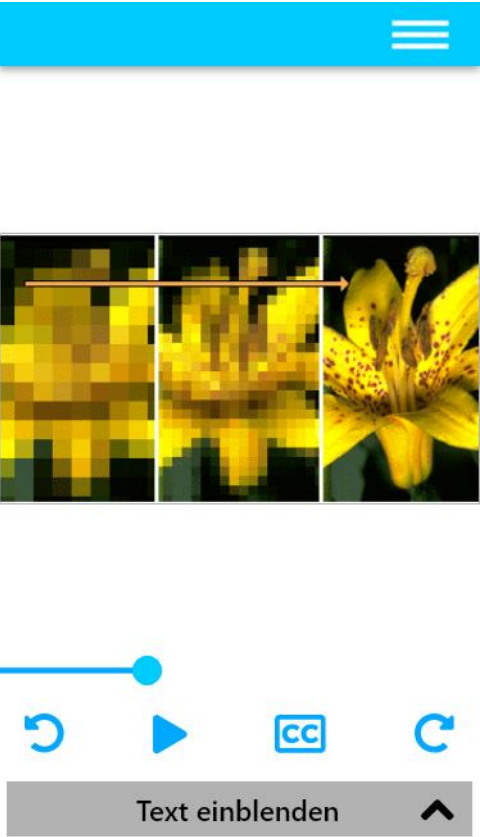
solcher Verläufe werden in unterschiedlicher Gewichtung überlagert. Dabei kann die Annäherung an das Originalbild durch die Anzahl der Überlagerungen beeinflusst werden. Schließlich erhält man zur Beschreibung des Bilds nur noch eine Reihe von mathematischen Faktoren. Zuletzt werden die nun zurecht sortierten Faktoren mit einer Huffman-Kodierung verlustfrei komprimiert.



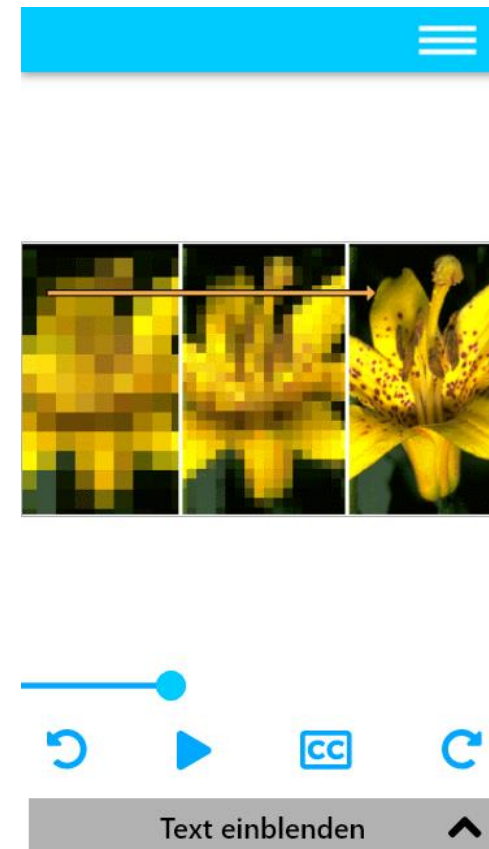
6.9 (I) JPG

Screen	Sprechertexte	Stichwörter / Notizen	Regieanweisungen
 <p>8x8 Pixelblock des Ausgangsbildes: Diesen gilt es beispielhaft anzunähern.</p> <p>$\text{Block} = 0.5 \cdot \text{F} + 0.5 \cdot \text{A} + 0.2 \cdot \text{W}$</p> <p>schlechter besser</p>	<p>130901 Probieren Sie nun die JPG Kompression mithilfe des Reglers durchzuführen, beobachten Sie dabei die Bildung von Blockartefakten im Bild.</p>		<p>130901 Durch das Verschieben des Reglers wird eine andere Kurve angewendet und man sieht welche Auswirkung das auf den Block hat.</p>

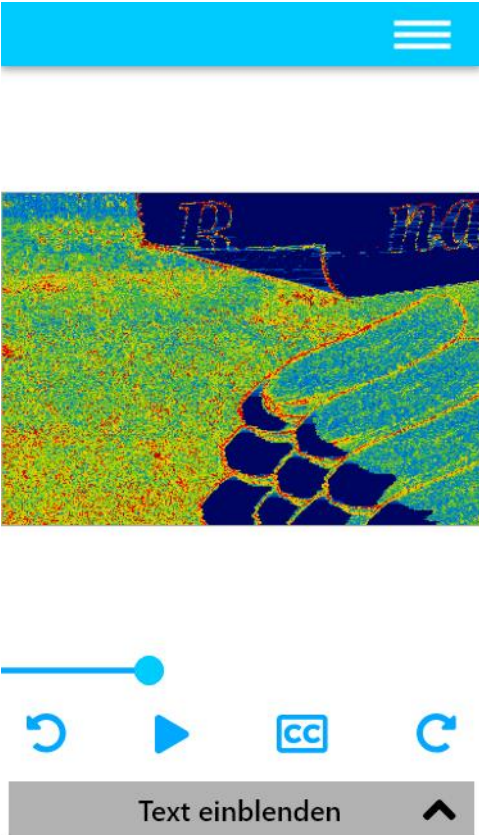
6.10 (A) GIF

Screen	Sprechertexte	Stichwörter / Notizen	Regieanweisungen
	<p>131001 Das Graphics Interchange Format, kurz GIF, ist ein Grafikformat mit einer verlustfreien Kompression. Das besondere an GIF ist die Möglichkeit der Speicherung von mehreren, auch übereinanderliegenden, Einzelbildern. Dadurch wird die Darstellung als Animation ermöglicht, was auch der Grund ist, warum GIF eine hohe Popularität besitzt.</p> <p>131002 GIF unterstützt nur 256 indizierte Farben oder Graustufen, weshalb komplexe Bilder in ihrer Farbdarstellung reduziert werden müssen. Später kam auch die Möglichkeit der Transparenz dazu.</p> <p>131003 Außerdem unterstützt das Format auch so genanntes Interlacing, wodurch beim Laden eines GIFs die Auflösung Schritt für Schritt erhöht werden kann. Das war vor allem früher von Vorteil, da trotz langsamer Internetverbindung schon etwas angezeigt wurde.</p>	<p>131001 - Verlustfrei - Animationen möglich</p> <p>131002 - Unterstützt nur 256 Farben - Transparenz möglich</p> <p>131003 - Interlacing</p>	<p>131001 Als erstes wird eine GIF-Animation gezeigt.</p> <p>131002 Dann wird gezeigt, wie die Farben des Beispielbildes auf 256 reduziert werden.</p> <p>131003 Zum Schluss sieht man wie Interlacing aussieht.</p>

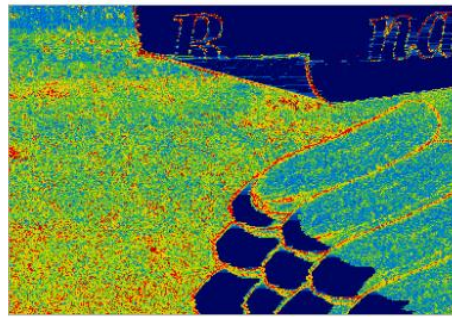
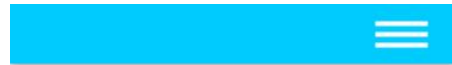
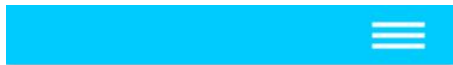
	Zur Kompression wird LZW verwendet, welchen wir schon in einem anderen Kapitel kennengelernt haben.		
--	---	--	--



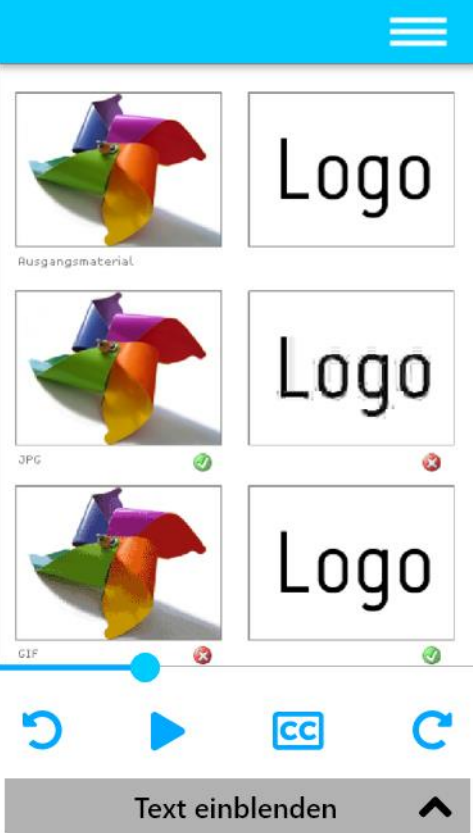
6.11 (A) PNG

Screen	Sprechertexte	Stichwörter / Notizen	Regieanweisungen
	<p>131101 PNG steht für Portable Network Graphics und ist heute das meistverwendete verlustfreie Grafikformat im Internet. Die Entwicklung dieses Grafikformates begann Ende 1994 mit dem Ziel das Grafikformat GIF zu ersetzen.</p> <p>Das PNG Format unterstützt verschiedene Farbtiefen, üblicherweise 8, 24 und 32 Bit, wobei die 32Bit Variante einen zusätzlichen 8-Bit-Kanal für Transparenz-Informationen enthält.</p> <p>Die Kompression eines PNG verläuft in drei Schritten. Zuerst kommt es zum Vorfiltern, wo sehr ähnliche Farben auf einen Farbwert gesetzt werden.</p> <p>131102 Danach kommt es zur Wörterbuch-basierten Kodierung per LZ77 Algorithmus, welcher ein Vorgänger des im LZW verwendeten LZ78 ist.</p> <p>Das Verlustfreie LZ77 sucht zu Beginn nach sich wiederholende Sequenzen von Daten. Wenn der Algorithmus auf eine Sequenz trifft, welche es schon einmal gab, gibt es nur einen Verweis auf</p>	<p>131101 - Unterstützt 8, 24 und 32 Bit - Transparenz möglich</p> <p>- Ähnlich Farbwerte werden vereinheitlicht</p> <p>131102 - Wiederholende Sequenzen erzeugen nur Verweis</p>	<p>131101 Zu Beginn wird das Beispielbild eingeblendet.</p> <p>131102 Dann wird gezeigt wie PNG sich wiederholende Sequenzen auslöst. Zur Veranschaulichung wird gezeigt welche Sequenzen im Beispielbild sich wiederholen.</p>

	<p>die entsprechende Sequenz, was bei manchen Bildern viel Speicherplatz spart. Enthält ein Bild zum Beispiel zwei identische schwarze Kreise, so verbraucht nur der erste Kreis Speicherplatz.</p> <p>Im Letzten Schritt werden die bis dahin erzeugten Daten noch mithilfe der Huffman-Kodierung komprimiert.</p>		
--	---	--	--



6.12 (A) Tipps & Tricks

Screen	Sprechertexte	Stichwörter / Notizen	Regieanweisungen
	<p>131201</p> <p>Um richtig zu komprimieren, sollten im Grafikbereich ein paar Tipps und Tricks beachtet werden. Nicht alle Kompressionsverfahren lassen sich gut auf jede Art von Bildern anwenden. Kompression wird häufig bei Bildern verwendet um diese im Internet zu publizieren, per Mail zu verschicken oder zum Download anzubieten.</p> <p>Ein Bild, das viele Farbverläufe besitzt, also z.B. ein klassisches Foto, sollte mit JPG komprimiert werden. Dieses Verfahren wurde entwickelt, um Bilder mit Farbverläufen besonders gut zu komprimieren, da diese optisch leicht mit mathematischen Verläufen angenähert werden können.</p> <p>Ein Bild das harte Farbkanten besitzt, wie zum Beispiel Logos oder Schriftzüge auf einfarbigem Grund, sollte mit GIF oder PNG komprimiert werden, da bei diesen Verfahren die Reduktion nicht durch Zusammenfassung von Bildinhalten erfolgt. Unscharfe Kanten, wie bei JPG üblich, gibt es bei PNG und GIF nicht. Dafür ist speziell GIF auf 256 Farben beschränkt, weshalb man heute in der Regel PNG verwendet.</p>	<p>131201</p> <ul style="list-style-type: none"> - JPG für Bilder mit Farbverläufen - PNG & GIF für harte Farbkanten - Geringster Verbrauch von Speicherplatz: <ol style="list-style-type: none"> 1. GIF 2. JPG 3. PNG 	<p>131201</p> <p>Die Tipps werden nach und nach anhand des Beispielfotos visuell dargestellt.</p>

	<p>Allgemein kann gesagt werden, dass GIF aufgrund der beschränkten Anzahl von Farben am wenigsten Speicherplatz benötigt. JPG braucht mehr Speicherplatz als GIF, ist aber immer noch kleiner als das verlustfreie PNG.</p>		
--	--	--	--