

jpegAGA

COLLABORATORS

	<i>TITLE :</i> jpegAGA		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		July 19, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1 jpegAGA	1
1.1 jpegAGA 2.2	1
1.2 inhaltsverzeichnis	1
1.3 vorteile	2
1.4 dankagung	2
1.5 installation	3
1.6 programmaufruf	3
1.7 optionen	5
1.8 programme	7
1.9 map	7
1.10 bildschirm	9
1.11 geschichte	10
1.12 aufruf	11
1.13 autor	12
1.14 bezugsquellen	12

Chapter 1

jpegAGA

1.1 jpegAGA 2.2

jpegAGA 2.2 von Günther Röhrich

jpegAGA ist ein Anzeigeprogramm für JPEG-Bilder mit der derzeit besten Bildqualität. (sofern eine map-Datei für das Bild vorhanden ist) Zur Anzeige ist der AGA-Chipsatz, ein 68020 Prozessor (oder höher) sowie OS 3.0 (oder höher) erforderlich. Es wird der HAM8-Modus verwendet, lediglich die Graustufendarstellung erfolgt im 256-Farben-Modus. Selbstverständlich ist jpegAGA 2.x noch voll kompatibel zu vorhergehenden Versionen, trotz der vielen Neuerungen. Derzeit wird die Version 6 des Quelltextes der "Independent JPEG Group" für die Dekodierung verwendet.

Wenn zu dem JPEG-Bild eine vorberechnete Farbpalette in Form einer Datei mit der Endung .map vorhanden ist dann erfolgt die Anzeige in einer Bildqualität, die von keinem anderen Anzeigeprogramm bisher erreicht wird. Mit Hilfe der beigefügten Programme und des Shell-Skripts kann zu jedem Bild eine solche map-Datei, die lediglich 198 Bytes lang ist, erzeugt werden.

Falls keine map-Datei vorhanden ist, dann erfolgt die Anzeige in schlechterer Qualität. Für Graustufendarstellung wird diese Datei nicht benötigt.

Inhaltsverzeichnis

1.2 inhaltsverzeichnis

Inhaltsverzeichnis

Vorteile von jpegAGA
Copyright-Hinweise und Danksagungen
Installation
Programmaufruf und Eingabedateien
Optionen
Weitere Programme

Erzeugung der map-Dateien
Anzeigebildschirm
Entstehungsgeschichte
Aufruf zur Mitarbeit
Adresse des Autors
Bezugsquellen

1.3 vorteile

Die Vorteile von jpegAGA:

- sehr gute Bildqualität (besser als bei allen anderen Anzeigeprogrammen)
- vollständiger Quelltext ist vorhanden
- größtmögliche Kompatibilität zu allen JPEG-Bildern da zur Dekodierung der JPEG-Quelltext der Independent JPEG Group im Originalzustand verwendet wird (derzeit Version 6)
- Unterstützung des Graustufen-JPEG Formats sowie Anzeige von Farbbildern in Graustufen
- Auswahl des Bildschirmmodus nach der Größe des Bildes
- Verkleinerungen sind möglich, kann auch für eine sehr schnelle Vorschau verwendet werden
- Bilder werden, falls gewünscht, so skaliert, daß sie komplett auf den Bildschirm passen
- bei Auswahl mehrerer Bilder wird schon das nachfolgende Bild im Hintergrund geladen während das Vorherige noch angezeigt wird
- unterstützt progressive JPEG Dateien

1.4 dankagung

Dieses Programm ist Public Domain (d.h. man braucht nichts zu bezahlen). Der beigefügte Quelltext kann nach Belieben verwendet werden. Für den JPEG-Quelltext sind die Hinweise in der Datei README zu beachten.

Bedanken möchte ich mich noch bei folgenden Personen:

Meine Betatester:

Denis Ahrens (d.ahrens@bamp.berlinet.in-berlin.de)
Lars Tewes (LSR@quick.sh.sub.de)
Tim Reddy (tcreddy@prairienet.org)
sowie alle anderen die hier nicht aufgeführt sind.

Gunther Nikl (gnikl@informatik.uni-rostock.de) für seine Hilfe und Vorschläge.

Pawel Hernik (spi-ph@srv1.tu.kielce.pl) und andere Autoren für den "Chunky to Planar" Quelltext

The Independent JPEG Group (jpeg-info@uunet.uu.net):
für den Quelltext zum Dekodieren des JPEG-Bildformats

Free Software Foundation, Inc. sowie allen Mitwirkenden für die Entwicklung des GNU C Compilers und der weiteren Software

Martin Huttenloher (xen@magic.in-ulm.de)
für die wunderschönen MagicWB Piktogramme

1.5 installation

Installation:

Die Programme aus dem bin-Verzeichnis können in ein beliebiges Verzeichnis kopiert werden, sie sollten sich aber im Suchpfad der Shell befinden. Die Endung der Dateien gibt an für welchen Prozessor sie gedacht sind, nach der Installation sollte man aber die Endungen entfernen. Wer jpegAGA von der Workbench aus verwenden möchte der kann das Hauptprogramm (jpegAGA.020) an einen beliebigen Ort kopieren.

Endung 000: 68000 oder 68010 Prozessor (läuft auch auf allen anderen)
Endung 020: nur für 68020 oder höhere Prozessoren
Endung 020881: nur für 68020 oder höhere Prozessoren mit Koprozessor

Die Programme wurden mit dem gcc 2.7.0 übersetzt. Die ixemul.library wird aber trotzdem nicht benötigt.

ACHTUNG: jpegAGA 2.2 funktioniert nicht mehr mit Grafikkarten, da es jetzt eine eigene, inkompatible "Chunky to Planar" Routine verwendet.

Für die Verwendung der Shell-Skripte CreateMap(Big) muß JPEGTMP: mit einem assign-Befehl an ein Verzeichnis auf einer Festplatte mit ausreichend Speicherplatz zugewiesen werden. (z.B. assign JPEGTMP: HD:T) Die Programme djpeg und cjpeg greifen unter Umständen ebenfalls darauf zu. Dieses Verzeichnis wird jetzt auch von jpegAGA selbst verwendet, und zwar bei der Bearbeitung großer progressiver JPEG-Dateien. Am Besten schreibt man diesen Befehl in die User-Startup Datei damit er nach jedem Neustart ausgeführt wird.

Die Shell-Skripte CreateMap und CreateMapBig muß man in das S: Verzeichnis kopieren. Sie setzen voraus, daß sich die Programme mit den Namen djpeg sowie ppm2AGA im Suchpfad der Shell befinden und daß ein JPEGTMP: Verzeichnis existiert.

1.6 programmaufruf

Der Aufruf von jpegAGA geschieht wie folgt:

Von der Shell:

```
jpegAGA Optionen Eingabedateien
```

Von der Workbench:

Doppelklick auf das Piktogramm oder auf ein entsprechend vorbereitetes Piktogramm einer Bilddatei.

Shell-Benutzer können das Skript PV für eine schnelle Vorschau verwenden. Es wird genauso aufgerufen wie jpegAGA selbst, allerdings sind schon etliche Optionen voreingestellt. Natürlich kann man sich auch eigene Skripte mit bevorzugten Optionen erstellen. Damit spart man später eine Menge Tipparbeit.

Eingabedateien:

Die JPEG-Dateien, die gelesen werden sollen. Es können beliebig viele Dateien angegeben werden, an beliebiger Stelle innerhalb der Eingabezeile. Namensmuster (z.B. #?.jpg) sind ebenfalls zulässig. Um eine Datei anzuzeigen, die mit dem "-" Zeichen beginnt muß man statt dessen "?" angeben. (Ansonsten nimmt das Programm an, daß es sich um eine falsch geschriebene Option handelt.) Wenn keine Dateinamen angegeben wurden, dann erscheint das bekannte Auswahlfenster der asl.library. Es können mit Hilfe der Shift-Taste auch mehrere Bilder ausgewählt werden. Um auch Bilder ohne die Endungen ".jpg" oder ".jpeg" auswählen zu können muß das vorgegebene Namensmuster gelöscht werden.

Bei einem Start durch einen Doppelklick von der Workbench aus erscheint ebenfalls das Auswahlfenster, sofern man keine Dateien mit Hilfe der Shift-Taste ausgewählt hat. Wesentlich komfortabler ist es, wenn man für die JPEG Datei ein Projekt-Piktogramm erstellt und in dem Feld "Standardprogramm" das Programm jpegAGA (mit genauer Pfadangabe!!) einträgt. Nach einem Doppelklick auf dieses Piktogramm wird das Bild dann sofort angezeigt. Wie üblich können mit Hilfe der Shift-Taste auch mehrere Bilder gleichzeitig zum Anzeigen ausgewählt werden. Das ist empfehlenswert, damit durch das Laden im Hintergrund Zeit gespart wird.

Für alle die es noch nicht wissen:

Man wählt mehrere Dateien aus, indem man die Shift-Taste gedrückt hält und alle Bilder einmal anklickt. Anschließend führt man noch einen Doppelklick auf eines der Bilder aus. Alternativ dazu kann man (ebenfalls bei gedrückter Shift-Taste) einen Rahmen um die gewünschten Bilder ziehen und anschließend auf eines der Bilder einen Doppelklick ausführen. Eine Kombination der genannten Verfahren ist auch möglich.

Neu in Version 2.2: Es werden auch progressive JPEG-Dateien unterstützt. Vom Beginn des Einlesens solcher Dateien bis zum Öffnen des Bildschirms kann es etwas dauern. Das läßt sich nicht vermeiden. Im Shell-Fenster wird mit dem Kürzel "progr." angezeigt, wenn eine progressive JPEG-Datei vorliegt.

Graustufenbilder werden direkt ohne aufwendige Umrechnung auf einem 256-Farben Bildschirm angezeigt. Leider gibt es auch Graustufenbilder die fälschlicherweise als Farbbilder codiert wurden. In diesem Fall sollte man die -GRAY Option verwenden. (Und schick eine Beschwerde an denjenigen, der das Bild erzeugt hat.)

Bei der Anzeige in Farbe wird, falls vorhanden, die map-Datei des jeweiligen Bildes mit verwendet. Die map-Datei muß den Namen des Bildes mit der zugefügten Endung .map haben. Zur Unterstützung des PC-Dateisystems wird aber auch ein Namen unterstützt, der sich durch Ersetzen der Endung (der Text nach dem letzten Punkt) durch "map" ergibt. Wenn eine map-Datei gefunden wurde, wird dies im Konsolenfenster mitgeteilt.

Beispiele für Dateinamen:

```
Bilddatei: picture.jpeg      map-Datei: picture.jpeg.map oder picture.map
           pic1.jpg          map-Datei: pic1.jpg.map      oder pic1.map
```

Map-Dateien werden außerdem noch in dem Verzeichnis gesucht, auf das die Umgebungsvariable MAPDIR weist.

1.7 optionen

Optionen:

Die Optionen dürfen in beliebiger Reihenfolge stehen, die Groß- und Kleinschreibung ist beliebig. Werden Optionen weggelassen, dann werden interne Vorgabewerte angenommen.

Beim Start von der Workbench müssen die Optionen mit Hilfe des Merkmals "ARGS" (auch Tool Type genannt) angegeben werden. (z.B. ARGS= -vga -scale 1/2 -gray)

Wenn das Bild ein Piktogramm hat, werden die Merkmale des Bildes ebenfalls ausgewertet, sie haben Vorrang vor den Merkmalen von jpegAGA. Allerdings gelten sie ausschließlich für das betreffende Bild. Wenn das Bild über das Datei-Auswahlfenster ausgewählt wurde, werden die Merkmale der einzelnen Bilder nicht berücksichtigt, das gilt auch beim Aufruf von der Shell aus.

-GRAY

Die Anzeige erfolgt stets im Graustufen Modus. Das ist sehr viel schneller als in Farbe und daher nützlich für eine Vorschau.

-VGA

Beim Anzeigen des Bildes wird der VGA-Monitortreiber (auch Multiscan genannt) verwendet.

Die Option -VGA ist dann zu setzen wenn bekannt ist, daß das Bild für die Anzeige auf einem PC oder auf einem anderen Rechner mit quadratischer Auflösung des Bildschirms vorgesehen war. (Dies trifft für fast alle JPEG-Bilder zu.) Apropos quadratische Auflösung: Hier gibt es leider viele Mißverständnisse, so daß ich näher darauf eingehen werde:

Unter "quadratisch" ist nicht gemeint, daß auf dem Bildschirm in horizontaler und vertikaler Richtung genau gleich viele Bildpunkte vorhanden sind. (z.B. 1024x1024) Es ist gemeint, daß ein Quadrat bestehend aus z.B. 100x100 Bildpunkten auf dem Monitor ebenfalls als Quadrat zu sehen ist. Da das Seitenverhältnis einer üblichen Bildröhre 4:3 beträgt, (angelehnt an die internationale Fernsehnorm) muß auch die horizontale und vertikale Auflösung Vielfache davon betragen. (z.B. 640x480 da

```
640  4
--- = - )
```

480 3

Nur so ist gewährleistet, daß ein solches Bild verzerrungsfrei dargestellt wird.

-SUPER72

Diese Option wird nur dann akzeptiert, wenn auch -VGA angegeben wurde. Bei sehr großen Bildern wird dann der SUPER72 Bildschirmmodus verwendet. Bei vielen Monitoren muß aber der SUPER72-Monitortreiber angepaßt werden, (z.B. mit dem Programm "moned") damit sich eine quadratische Auflösung ergibt.

-SCALE 1/x (neu in Version 2.0)

Das Bild wird um den Faktor 1/x verkleinert. Zulässige Faktoren sind 1/1, 1/2, 1/4 sowie 1/8. Als angenehmer Nebeneffekt wird das Bild sehr viel schneller dekodiert und natürlich auch angezeigt. Damit kann man eine sehr schnelle Vorschau des Bildes erzeugen.

-FIT (neu in Version 2.0)

Wenn diese Option angegeben wird, dann wird das Bild stets so skaliert, daß es vollständig auf den Bildschirm paßt. Ein Überschreiten bis ca. 35 Bildpunkte wird aber noch toleriert. Leider ist nur eine Verkleinerung um die Faktoren 1/2, 1/4 und 1/8 möglich. Die Optionen -VGA und -SUPER72 werden nach wie vor berücksichtigt. Der SUPER72-Modus wird aber nur verwendet, wenn das Bild für den VGA Modus zu groß ist.

-NOSMOOTH (neu in Version 2.0)

-DCT FAST (neu in Version 2.0)

Mit diesen Optionen erfolgt die Dekodierung etwas schneller, allerdings wird die Bildqualität ein wenig schlechter. Für eine schnelle Vorschau sind die Optionen aber durchaus sinnvoll.

-SMR (neu in Version 2.0)

Vor dem Anzeigen eines Bildes kann man den Bildschirmmodus mit einem Auswahlfenster wählen. Das Laden im Hintergrund ist mit dieser Option nicht mehr möglich. Außerdem wird diese Option ignoriert, wenn man sie als Merkmal in das Piktogramm eines Bildes schreibt.

-MAXMEM N (neu in Version 2.2)

Mit dieser Option kann der maximale Speicherverbrauch für das Dekodieren von progressiven JPEG-Bildern eingestellt werden. Es müssen die DCT-Koeffizienten für das gesamte Bild zwischengespeichert werden, das kann mehrere Megabytes erfordern. (Standardeinstellung: 1MB) N ist in Kilobyte anzugeben. (z.B. -maxmem 200 für 200K)

Mit der Endung M sind es Megabytes. (z.B. -maxmem 4M für 4 Megabyte) Wenn mehr Speicher benötigt wird, dann werden Auslagerungsdateien im Verzeichnis JPEGTMP: verwendet. Diese werden natürlich automatisch auch wieder gelöscht.

Alternativ zu dieser Option kann auch die Umgebungsvariable JPEGMEM verwendet werden, die genau die gleiche Funktion übernimmt.

In Piktogrammen von Bildern werden noch weitere Optionen unterstützt, man kann mit ihnen Optionen im Piktogramm von jpegAGA aufheben. Wenn man z.B. die Option -NOGRAY in einem Bild-Piktogramm setzt, dann wird das Bild immer in Farbe angezeigt, auch wenn im jpegAGA-Piktogramm die Option -GRAY gesetzt ist.

Option	Gegenstück (nur Bild-Piktogramm)
-DCT FAST	-DCT INT
-NOSMOOTH	-SMOOTH
-GRAY	-NOGRAY
-VGA	-NOVGA
-SUPER72	-NOSUPER72
-FIT	-NOFIT

Wenn man jpegAGA mit einer falschen Option startet, (z.B -V) dann erhält man eine Liste der verfügbaren Optionen, jeweils mit einer kurzen Erklärung. Das funktioniert aber nur von der Shell aus.

1.8 programme

Ich habe das Programm "djpeg" beigelegt, man kann damit JPEG-Bilder in das ppm-Format umwandeln. (Es ist auch zur Erzeugung der map-Dateien erforderlich)

Aufruf: djpeg Eingabedatei >Ausgabedatei

Bei der direkten Umwandlung in das GIF-Format muß man JPEGTMP: mit einem assign Befehl zuweisen (z.B. assign JPEGTMP: HD:T), da djpeg eventuelle temporäre Dateien dorthin ablegt. Alles Weitere ist der zugehörigen Anleitung zu entnehmen. Wenn ich mal Zeit habe werde ich die JPEG-Routinen in ppm2AGA einbauen, so daß dieses Programm dann nicht mehr benötigt wird.

Der Vollständigkeit halber ist noch das Programm "cjpeg" vorhanden, mit dem man selber JPEG-Dateien erzeugen kann. Eine Beschreibung von djpeg und cjpeg findet man in den zugehörigen Anleitungen.

1.9 map

Erzeugung der map-Dateien

Diese Dateien sind unbedingt erforderlich, damit die Anzeige in der best-möglichen Qualität erfolgt. Sie werden im Normalfall durch das Shell-Skript CreateMap erzeugt. Der Aufruf lautet: (von der Shell aus)

CreateMap Bilddatei

Bei erfolgreicher Ausführung wird eine map-Datei mit korrektem Namen erzeugt. (auch auf PC-Dateisystemen) Man kann auch mehrere Dateien mit einem Aufruf erzeugen. So werden z.B. durch den Befehl

```
SPat CreateMap pictures/#?.jpg
```

alle Bilder mit der Endung .jpg im Verzeichnis pictures bearbeitet.

Falls es nicht möglich ist, die map-Datei an der gleichen Stelle wie das Bild abzuspeichern, dann wird die map-Datei in dem Verzeichnis abgelegt, auf das die Umgebungsvariable MAPDIR weist.

Das Setzen der Umgebungsvariablen kann z.B. erfolgen mit dem Befehl:

```
setenv MAPDIR HD:tempdir
```

Die Variable geht dann allerdings nach einem Zurücksetzen des Rechners verloren. Man kann sie dauerhaft sichern mit dem Befehl:

```
copy ENV:MAPDIR ENVARC:
```

Somit ist es möglich, auch Bilder von einer CD-ROM in optimaler Qualität anzuzeigen.

ACHTUNG: Es muß unbedingt vermieden werden, daß dieselbe map-Datei für verschiedene Bilder verwendet wird. Das führt zu einer schlechteren Bildqualität.

Mit dem beigefügten Shell-Skript "CreateMAPBig" geht die Erzeugung der map-Dateien erheblich schneller. Beim Dekodieren des Bildes wird eine Verkleinerung um den Faktor 1/2 vorgenommen. Das ist hinnehmbar, denn zum Erzeugen der map-Dateien wird nur die Farbverteilung ausgewertet und nicht der ganze Inhalt des Bildes. Bei ganz großen Bildern könnte man sogar eine Verkleinerung um den Faktor 1/4 vornehmen. Das Shell-Skript müßte man dann ein bißchen ändern. Bei kleinen Bildern (z.B. 320x200 Bildpunkte) sollte man allerdings überhaupt nicht verkleinern und das normale Skript "CreateMap" verwenden. Wenn man ein komplettes Verzeichnis auf einmal, mit großen und kleinen Bildern, bearbeiten möchte, dann empfiehlt es sich ebenfalls, das normale Skript CreateMap zu verwenden.

Bei Auftauchen von Problemen mit der Erzeugung der map-Dateien ist folgendes zu prüfen:

- Wurde, wie im Abschnitt "Installation" beschrieben, alles Erforderliche installiert ?

Benötigt werden die Programme djpeg sowie ppm2AGA, die sich im Suchpfad der Shell unter diesem Namen befinden müssen. Das Shell-Skript selbst muß im S: Verzeichnis sein.

Achtung: - ppm2AGA muß die Version 1.4 oder höher aufweisen
- nur djpeg aus dieser Distribution verwenden

- Ist das Verzeichnis JPEGTMP: korrekt zugewiesen ?
- Reicht der Speicherplatz auf der Festplattenpartition, auf die JPEGTMP: weist, aus ?

Bei sehr großen Bildern können sogar mehr als 10MB benötigt werden. Es empfiehlt sich dann aber "CreateMapBig" zu verwenden, das reduziert den Speicherverbrauch auf der Festplatte dramatisch.

- Ist genug Arbeitsspeicher vorhanden ?

Falls nicht kann man den benötigten Arbeitsspeicher reduzieren, indem man beim Aufruf des Shell-Skripts die Option -M0 angibt. (Näheres siehe Beschreibung von ppm2AGA)

Beispiel: CreateMap pic.jpg -M0 oder SPat CreateMap #?.jpg -M0

Der erforderliche Arbeitsspeicher ist bei Verwendung der -M0 Option nahezu unabhängig von der Größe des Bildes.

- Ist die Diskette, auf der sich das JPEG-Bild befindet, schreibgeschützt oder ist zu wenig Speicherplatz vorhanden ?
- Wurde die Umgebungsvariable MAPDIR korrekt gesetzt ? Man kann dies überprüfen mit dem Befehl: getenv MAPDIR

Das Shell-Skript kann stets durch Eingabe von CTRL-C abgebrochen werden. Temporäre Dateien im Verzeichnis JPEGTMP: werden bei Abbruch oder Beendigung automatisch gelöscht.

1.10 bildschirm

Der Anzeigebildschirm

Durch Bewegen der Maus kann der Bildausschnitt verschoben werden, falls das Bild größer als der Overscan-Bereich ist. Bei Farbanzeige treten leider Störungen am linken Bildrand auf, die durch das HAM-Verfahren bedingt sind. Die Störungen sind aber in dieser Version in der Regel kleiner als bei allen anderen Programmen.

Der Bildschirm hat außerdem eine voll funktionsfähige Ziehleiste sowie einen Vordergrund/Hintergrund-Schalter, die allerdings unsichtbar sind.

Durch Druck auf die rechte Maustaste wird die Anzeige beendet bzw. abgebrochen. Wenn es noch weitere Bilder gibt wird auf das nachfolgende umgeschaltet. Sollte das Chip-RAM für das Laden des nächsten Bildes im Hintergrund nicht ausreichen, dann wird dies noch einmal versucht, sobald der vorherige Bildschirm geschlossen wird. An der Form des Mauszeigers ist zu erkennen, ob noch eine Aktivität im Hintergrund stattfindet. Da auf die linke Maustaste nicht reagiert wird, kann man während der Berechnung des Bildes einen anderen Bildschirm aktivieren und erst später den Anzeigebildschirm durch Drücken der

linken Maustaste erneut aktivieren. Ein vollständiger Abbruch einer Bildfolge ist durch Drücken der Esc-Taste möglich.

Hinweis: Aufgrund von Fehlern im Betriebssystem kann es vorkommen, daß seitlich Teile des Bildes fehlen oder daß nur Müll angezeigt wird. Es empfiehlt sich in diesem Fall, das Bild so lange seitlich zu verschieben bis es korrekt angezeigt wird.

1.11 geschichte

Entstehungsgeschichte

1.0 - 3. Juni 1994

- erste veröffentlichte Version

1.1 - 5. Oktober 1994

- die Option -SUPER72 wurde hinzugefügt
- beim Schließen des Grafikbildschirms ist jetzt kein Flackern mehr zu sehen und die Verzögerung ist geringer
- map-Dateien werden auch in dem Verzeichnis gesucht, auf das die Umgebungsvariable MAPDIR weist
- falls keine map-Datei gefunden wurde wird der Benutzer darauf hingewiesen, daß er eine erzeugen sollte
- es werden jetzt beliebig lange Dateinamen (inklusive Pfad) akzeptiert

2.0 - 14. Januar 1995

- etwas höhere Geschwindigkeit
 - die ixemul.library wird nicht mehr benötigt
 - es wird die Version 5A des JPEG Quelltextes der Independent JPEG Group verwendet (statt V4 in jpegAGA 1.x)
 - der Eingangspuffer ist jetzt 32768 Bytes groß, zum Lesen werden direkt die Routinen der dos.library verwendet
 - die Optionen -SCALE, -FIT, -NOSMOOTH, -DCT FAST und -SMR wurden hinzugefügt, weitere Optionen für Bild-Piktogramme
 - das nachfolgende Bild wird bereits eingeladen und dekodiert während das vorangegangene noch angezeigt wird
 - für einen vollständigen Abbruch muß man jetzt die Esc-Taste drücken, das Drücken des rechten Mausknopfes bewirkt das Weiterschalten zum nächsten Bild
 - der Grafikbildschirm erhält schon zum Zeitpunkt des Öffnens die
-

korrekten Farben, kein Flackern mehr nach dem Öffnen

- es werden beliebig viele Eingabedateien an beliebiger Stelle der Eingabezeile unterstützt, Namensmuster werden ebenfalls akzeptiert
- Auswahl der Dateien mit einem Auswahlfenster ist möglich
- volle Workbench-Unterstützung, allerdings gibt es immer noch keine grafische Benutzeroberfläche
- die Ausgabe in das Shell-Fenster ist jetzt besser und übersichtlicher geworden
- der Quelltext enthält auch eine make-Datei für den Aztec C Compiler
- eine kleine Änderung des HAM8 Kodierers bewirkt, daß beim Verschieben eines Bildes die Störungen am linken Rand sehr viel kleiner werden

2.1 - 4. Februar 1995

- Fehler behoben: wenn auf den Bildschirmen das Fenster bzw. die temporäre Bitmap nicht geöffnet werden kann kommt es nicht mehr zu Fehlfunktionen

2.2 - 17. September 1995

- es wird die Version 6 des IJG JPEG Quelltextes verwendet
- alle Programme sind mit dem gcc 2.7.0 und libnix 1.0 kompiliert
- progressive JPEG-Dateien werden unterstützt
- das Dateiauswahl-Fenster erscheint wieder, wenn die gewählten Bilder angezeigt wurden
- die Auswahlfenster merken sich jetzt die zuletzt getroffene Einstellung
- es wird eine eigene "Chunky to Planar" Routine verwendet, dadurch wird der Bildaufbau wesentlich schneller (20% schneller im Farbmodus, 50% im Graustufenmodus)
- das Programm reagiert auf das Drücken von CTRL-C im Shell Fenster (nicht immer)
- neue Option und Umgebungsvariable: -maxmem bzw. JPEGMEM (wurde nötig zur besseren Unterstützung progressiver JPEG-Dateien)
- Anleitungen im AmigaGuide Format

1.12 aufruf

Aufruf

Wie Du sicher gemerkt hast gibt es bei diesem Programm noch einiges zu tun, z.B. eine grafische Benutzeroberfläche, eventuell sogar mit MUI.

Meine Zeit ist leider begrenzt. (Ich studiere derzeit Elektrotechnik an der Uni Stuttgart.) Aus diesem Grund suche ich weitere Programmierer die bereit sind, mit mir zusammenzuarbeiten sowie Beta-Tester für die fertigen Programme. Wenn Du mitmachen willst, dann melde dich bitte. Ich werde dann auch eine ausführliche Beschreibung des Quelltextes erstellen. Einige Projekte können aber auch völlig unabhängig voneinander realisiert werden. Zum kompilieren kann jedes C-Entwicklungssystem auf dem Amiga verwendet werden. Die beste Optimierung erreicht man jedoch nur mit dem gcc.

Für die Neukompilierung von ppm2AGA ist das newiff-Paket von Commodore zwingend erforderlich. Die Version 37 ist auf der Fish-Disk 705 zu finden, die Version 39 ist auf den "3.1 Amiga Developer Update" - Disketten vorhanden. (Bezugsquelle siehe unten) oder auch im Aminet.

1.13 autor

Solltest Du weitere Fragen haben, so kannst Du mir jederzeit schreiben. Ich werde mich bemühen, alle Zuschriften zu beantworten. Falls Du einen Fehler im Programm entdecken solltest, (oder Mungwall/Enforcer Meldungen) so bitte ich ebenfalls um eine Benachrichtigung.

Alle Zuschriften sind an die folgenden Adressen zu richten:

Elektronische Post:

Guenther@studbox.uni-stuttgart.de
(Internet EMAIL)

Normale Post:

Günther Röhrich
Lerchenbergstr. 4
D-73733 Esslingen

Ich habe jetzt auch eigene WWW-Seiten eingerichtet. Sie sind zu finden unter:

<http://rpool1.rus.uni-stuttgart.de/~etk10325/>

1.14 bezugsquellen

Bezugsquellen

Gute Bilder im JPEG-Format findet man in rauhen Mengen in:

<ftp://uni-stuttgart.de> in `pub/graphics/pictures`

Das newiff-Paket, Version 39, findet man in:

<ftp://uni-stuttgart.de> in `pub/systems/amiga/aminet/gfx/misc`

sowie auf allen anderen Aminet-Servern. Wenn Du dazu keinen Zugang hast, dann

kannst Du es auch direkt von mir bekommen.

Das "3.1 Amiga Developer Update" ist erhältlich bei:

Hirsch & Wolf oHG
Mittelstr. 33
D-56564 Neuwied
Tel.: 02631-8399-0
Fax: 02631-839931

und kostet 50 DM. Ich weiß aber nicht, ob und wie lange es noch verfügbar ist.
